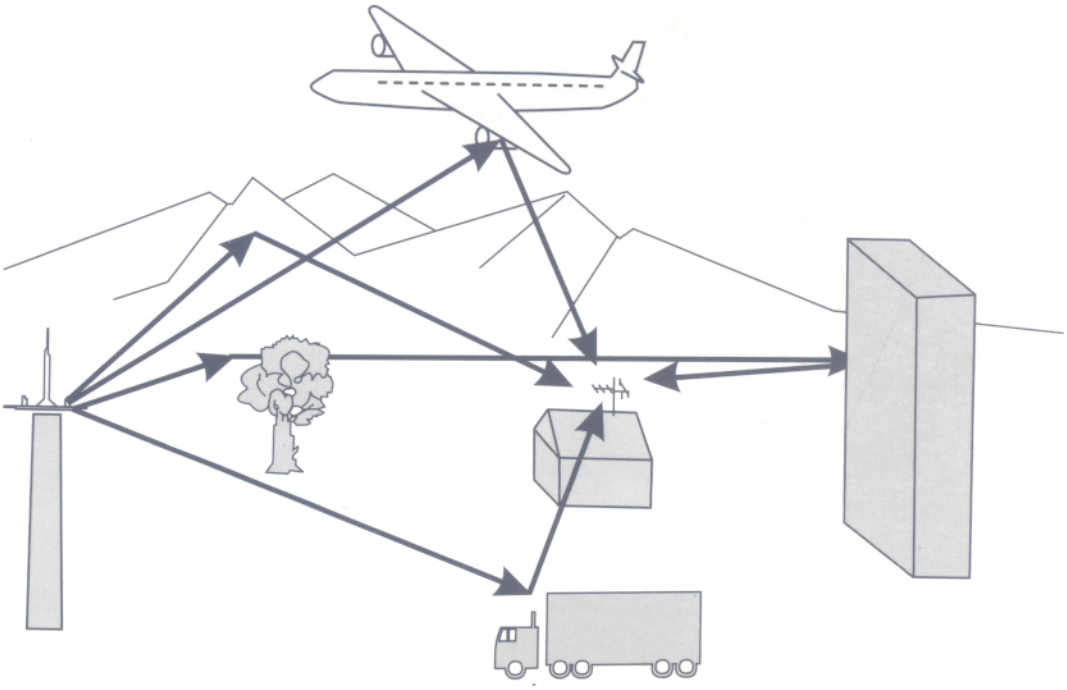
NGUYỄN VĂN HẬU – NGUYỄN HIẾU MINH

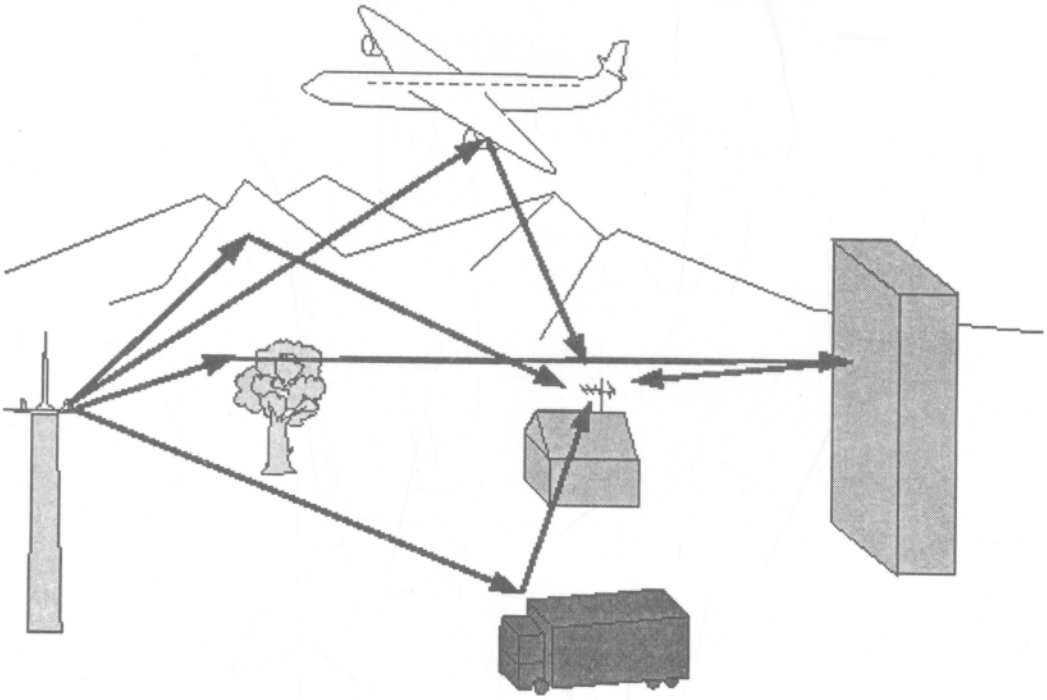
CƠ SỞ LÝ THUYẾT TRUYỀN TIN



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

NGUYỄN VĂN HẬU – NGUYỄN HIẾU MINH

CƠ SỞ LÝ THUYẾT TRUYỀN TIN



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
HÀ NỘI

MỤC LỤC

LỜI NÓI ĐẦU

Quyển sách này được viết trên cơ sở *hoàn chỉnh, mở rộng* các bài giảng của môn học "**Cơ sở lý thuyết truyền tin**" trong nhiều năm của nhóm tác giả. Quyển sách gồm hai phần: *lý thuyết mã* và *lý thuyết tín hiệu* được trình bày trong bảy chương.

Trong quyển sách này, lần đầu tiên, một số vấn đề được đề cập trong chương trình môn học, như vấn đề kỹ thuật nén số liệu, nén tín hiệu hình ảnh và tín hiệu tiếng nói; vấn đề dải thông của đường dây đôi; cụ thể hóa thuật toán mã Reed-Solomon; vấn đề điều chế OFDM, v.v... Đối tượng chính của quyển sách này là sinh viên đại học, cao đẳng của các trường kỹ thuật, thuộc chuyên ngành điện tử - viễn thông và công nghệ thông tin; cũng có thể làm tài liệu tham khảo cho các giáo viên thuộc các chuyên ngành trên.

Vì đối tượng chính là các sinh viên, cho nên quyển sách này rất chú trọng đến phần diễn giải các thí dụ. Mọi thuật toán đều có các thí dụ minh họa. Phần cuối của quyển sách có soạn thảo sẵn các câu hỏi và bài tập, sẽ rất lợi cho sinh viên và giáo viên. Quyển sách này không quá khó để đọc, nhưng ngay cả các nghiên cứu sinh cũng có thể tìm thấy các gợi ý bổ ích.

Đối với các giáo viên trẻ, chúng tôi có một lời khuyên nên bắt đầu giảng dạy từ phần lý thuyết tín hiệu, rồi sau đó đến phần lý thuyết mã, như vậy sẽ thuận lợi hơn, hợp logic hơn. Bởi vì, sinh viên cần làm quen với các khái niệm lấy mẫu, phổ tần số của tín hiệu trước khi nghiên cứu phần lý thuyết mã.

Chúng tôi mong rằng quyển sách sẽ đáp ứng được các nhu cầu của sinh viên và giáo viên và nhất định không làm độc giả thất vọng.

Tuy nhiên, do tài liệu tham khảo hạn hẹp và trình độ có hạn, nên nội dung quyển sách còn có chỗ thiếu sót, không vừa ý, chính bản thân nhóm tác giả cũng đã nhận thấy các vấn đề đó. Mong rằng độc giả, đồng nghiệp đóng góp ý kiến, để nội dung quyển sách được tốt hơn. Tác giả xin chân thành cảm ơn.

Các tác giả

CHƯƠNG MỞ ĐẦU

NHỮNG VẤN ĐỀ CHUNG CỦA HỆ THỐNG TRUYỀN TIN VÀ NỘI DUNG MÔN HỌC

1. MỞ ĐẦU

Tiếp nhận, lưu giữ, xử lý, truyền thông tin- đó là một trong các lĩnh vực phát triển sôi động nhất trong những năm gần đây và là lĩnh vực đầy triển vọng trong hoạt động của con người. Hàng chục nghìn công ty lớn, nhỏ khác nhau trên toàn thế giới với doanh thu hàng năm hàng trăm tỷ đôla từ dịch vụ tham gia vào nghiên cứu- phát triển, chế tạo- sản xuất, chuyển giao, khai thác-sử dụng và buôn bán các thiết bị truyền tin.

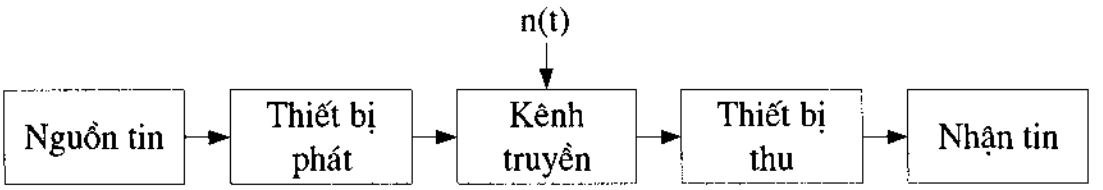
Môn học "**Cơ sở lý thuyết truyền tin**" nhằm cung cấp các hiểu biết về *các phương pháp (thuật toán), các mô hình (sơ đồ) khác nhau, cũng như nguyên lý chung xử lý thông tin; đồng thời từ đó biết được nguyên tắc chung để xây dựng và hoạt động của các hệ thống truyền tin, cũng như nghiên cứu các xu hướng hiện đại phát triển các hệ thống này.*

Đối tượng nghiên cứu của môn học là *hệ thống truyền tin điện tử*, là hệ thống được sử dụng để truyền **thông tin**, chứ không phải để truyền bất kỳ một đối tượng, hoặc năng lượng, hoặc tín hiệu nào khác. Tín hiệu trong các hệ thống truyền tin được thể hiện dưới dạng sóng điện từ (hoặc dòng điện) là đại lượng vật lý mang thông tin từ nguồn tin đến người sử dụng. Trong thực tiễn các hệ thống truyền tin rất đa dạng, chẳng hạn, hệ thống điện thoại cố định, di động, hệ thống internet, hệ thống phát thanh, truyền hình, v.v...Trong thực tiễn cũng tồn tại các hệ thống kỹ thuật điện tử khác, chúng được sử dụng làm *hệ thống đo lường*, chẳng hạn, hệ thống vô tuyến định vị (rađa), hệ thống dẫn đường, hệ thống đo tham số của môi trường, v.v...Trong các hệ thống này, thông tin có ích được tích luỹ vào tín hiệu (hay phát sinh trong tín hiệu) khi tương tác với môi trường xung quanh, hoặc đối tượng và phản ánh các tham số và đặc tính của các đối tượng này, hoặc của môi trường tín hiệu truyền lan. Tại điểm thu thông tin có ích được tách ra từ tín hiệu bằng cách đo các tham số phù hợp của sóng điện từ.

Vấn đề thứ hai mà chúng ta quan tâm trong mục này là khái niệm thông tin (information): *thông tin là gì và đâu là bản chất của quá trình truyền thông tin?* Khái niệm thông tin là một khái niệm trừu tượng và tồn tại nhiều cách diễn giải khác nhau, tuy nhiên, định nghĩa thông tin của C.Shannon là hàm ý nhất: *thông tin như là số đo của độ bất định* (mức độ chưa biết chứa trong nội dung của tin tức). Mục đích của truyền thông tin- đó là quá trình *làm mất đi* (hoặc nói chung *làm giảm bớt*) độ bất định đối với người nhận tin. Nghiên cứu định lượng thông tin được trình bày trong chương 1.

2. MÔ HÌNH CỦA HỆ THỐNG TRUYỀN TIN ĐIỆN TỬ

Để nghiên cứu hệ thống truyền tin điện tử, chúng ta hãy bắt đầu bằng việc nghiên cứu một mô hình chung nhất của mọi hệ thống truyền tin, như trình bày trên hình1. Hơn nữa, từ việc nghiêm cứu mô hình hệ thống truyền tin, chúng ta sẽ nhận thức rõ hơn về *nội dung môn học*.



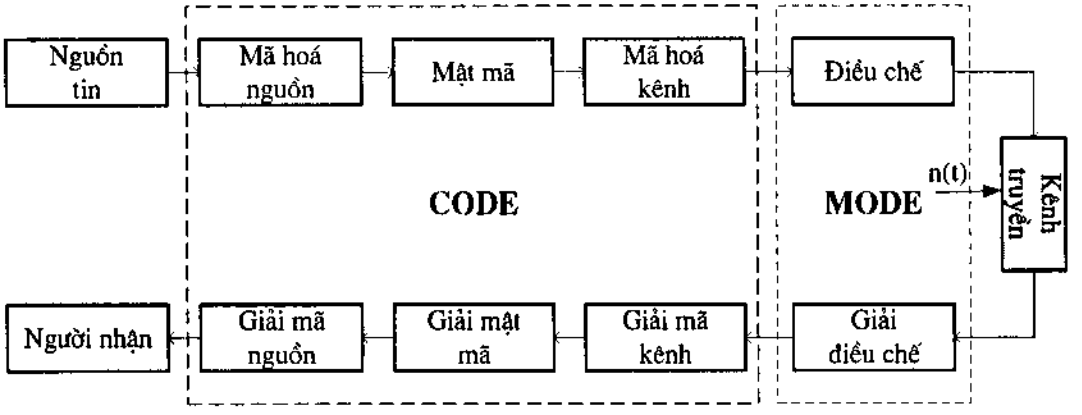
Hình 1. Mô hình chung của hệ thống truyền tin.

Mô hình này chứa đựng các phần tử cơ bản của mọi hệ thống truyền tin. Tuy nhiên, mô hình này không phản ánh được mọi hoạt động cần thực hiện (hay có thể thực hiện) vào thông tin trong quá trình truyền từ nguồn đến người sử dụng. Vì thế, chúng ta cần xây dựng mô hình chi tiết hơn, phản ánh *những mối quan tâm của môn học*, theo nghĩa là *mô hình của hệ thống truyền (cũng như lưu giữ) thông tin*, như được trình bày trên hình 2.

Chúng ta hãy xem xét các đặc trưng, các chức năng chính của các khối trong mô hình 2.

1) Nguồn thông tin (hay tin tức), nói ngắn gọn là nguồn tin

Nguồn tin là nơi sản sinh ra hay chứa đựng thông tin cần truyền đi.



Hình 2. Mô hình chi tiết hơn của hệ thống truyền tin.

Khi nói nguồn tin là nơi sản sinh ra thông tin, chúng ta ngụ ý chỉ đó là đối tượng, hệ thống, hoặc hiện tượng vật lý hình thành nên tin tức để truyền. Còn chính tin tức là giá trị hoặc số đo theo một đại lượng vật lý nào đó phản ánh trạng thái của đối tượng (của hệ thống, hoặc của hiện tượng). Như chúng đã biết tin tức nguyên thuỷ như tiếng nói, âm nhạc, hình ảnh, số đo các tham số môi trường, v.v... là hàm của thời gian, không phải là đại lượng điện. Với mục đích để truyền theo kênh liên lạc của hệ thống truyền tin điện tử, thì các tin tức này cần được biến đổi thành *tín hiệu điện* thay đổi theo thời gian, chúng ta ký hiệu là s(t) (hoặc x(t), w(t), ... tùy hoàn cảnh), tất

nhiên phải phản ánh tin tức ban đầu. Quá trình biến đổi này nhờ các thiết bị đầu cuối (còn gọi là *thiết bị định dạng*) như micro, camera, v.v... Còn khi nói nguồn tin là nơi chứa đựng (hay lưu giữ) thông tin để truyền, chúng ta muốn nói tới các bản ghi, các trang viết, hay bức ảnh, v.v..., nhờ phép định dạng, chúng được biểu diễn là chuỗi của các phần tử nhị phân, chúng ta gọi là véctơ **S**, có dạng $\mathbf{S}(s_1, s_2,...,s_N)$.

2). Mã hoá nguồn

Các tin tức ban đầu như tiếng nói, âm nhạc, hình ảnh, v.v...sau khi được chuyển đổi thành tín hiệu điện, ký hiệu là s(t), chúng không chỉ là hàm liên tục của thời gian, mà còn là hàm của một vài biến không gian. Chẳng hạn, tín hiệu video màu là hàm của 2 biến không gian chỉ độ sáng, màu sắc và biến thời gian t. Khi mã hoá nguồn, các tín hiệu liên tục theo thời gian (tín hiệu tương tự) cần liên quan đến thủ tục lấy mẫu (rời rạc theo thời gian), có nghĩa là biến đổi thành các phần tử rời rạc $\{s_i\}$. Vì vậy, chúng cũng có thể trình bày dưới dạng véctơ **S**.

Mã hoá trong trường hợp chung là phép biến đổi tin tức $\mathbf{S}(s_1, s_2,...,s_N)$ thành dạng mã được chọn trước nào đấy $\mathbf{C}(\alpha_1, \alpha_2,..., \alpha_K)$. Thông thường (chứ không nhất thiết) phép mã hoá nguồn là *phép mã hoá kinh tế, hay tiết kiệm, hay là phép nén số liệu, nhằm nén chặt số liệu để nâng cao tốc độ truyền tin, hoặc để giảm dải thông yêu cầu đối với kênh truyền.* Vì vậy, kích cỡ của tập mã - dim $\mathbf{C}(\alpha_1, \alpha_2,..., \alpha_K)$ nhỏ hơn, hoặc nhỏ hơn nhiều kích cỡ của tập tin - dim $\mathbf{S}(s_1, s_2,...,s_N)$.

Nếu phép nén số liệu với mục đích để khi giải nén *khôi phục chính xác* tin tức ban đầu, thì được gọi là *phép nén không tiêu hao thông tin.* Phép nén số liệu không gây tiêu hao thông tin được sử dụng để nén văn bản, các file chương trình, bản vẽ, v.v...Trong trường hợp này cần khôi phục lại chính xác tin tức ban đầu.

Trong trường hợp khác không cần (hoặc chính xác là không thể) khôi phục chính xác tin tức ban đầu, vì nhiều lý do khác nhau. Trong trường hợp như thế sẽ sử dụng *phép nén có tiêu hao thông tin*, khi giải nén *không có khả năng khôi phục chính xác tin tức ban đầu.* Tuy nhiên, phép nén có tiêu hao cho mức độ nén chặt (giảm dung lượng số liệu nén) cao hơn nhiều so với phép nén không gây tiêu hao. Các tin tức có nguồn gốc từ tiếng nói, âm nhạc, hình ảnh, v.v..., khi thực hiện phép mã hoá nguồn, thông thường là phép mã nén số liệu có tiêu hao.

Các phép mã hoá nguồn, hay các mã kinh tế (mã nén) chúng ta sẽ nghiên cứu kỹ hơn ở chương 2.

3). Mật mã

Các thông tin *quan trọng* và *nhạy cảm* khi truyền theo kênh liên lạc, hay lưu giữ cần phải được *an toàn.* Để bảo đảm chức năng *an toàn* và các chức năng khác như *xác thực*, *toàn vẹn nội dung tin tức*, người ta sử dụng mật mã.

Mật mã thực hiện biến đổi tin tức từ dạng *đọc* (hoặc *nghe*, hoặc *nhìn*) hiểu được thành ra dạng, mà khi đọc (hoặc nghe, hoặc nhìn) *không hiểu được.*

Mật mã và các vấn đề liên quan là một chủ đề rộng lớn, chúng ta sẽ tìm hiểu sơ lược ở chương 4.

4). Mã kênh (hay mã chống nhiễu)

Khi truyền tin tức theo kênh liên lạc có nhiễu có thể sẽ có một phần thông tin bị mất mát trên đó, tại điểm thu tin tức có thể bị mắc lỗi. Để bảo đảm khả năng *phát hiện ra lỗi* và mạnh hơn là *sửa để giảm lỗi*, người ta sử dụng *mã chống nhiễu*.

Ngày nay, tồn tại nhiều phương pháp (thuật toán) xây dựng mã chống nhiễu, nhưng chúng đều dựa trên nguyên tắc chung là *bổ sung* vào tin tức cần truyền *một lượng dư thừa* nào đó, chính nhờ tin tức dư thừa này mà phía thu có khả năng phát hiện lỗi và sửa lỗi. Như vậy, phép mã hoá nguồn xây dựng theo nguyên tắc *giảm độ dư thừa*, còn phép mã hoá chống nhiễu phải xây dựng theo nguyên tắc *tăng độ dư thừa*.

Cả phép mã hoá nguồn (phép mã nén), cả phép mã hoá bảo mật (mật mã), cũng như mã kênh (mã chống nhiễu) không nhất thiết phải tồn tại trong hệ thống truyền tin. Trong quá khứ các *hệ thống truyền tin tương tự* đã không có các phép biến đổi này. *Các hệ thống truyền tin số hoá*, có nhiều ưu điểm hơn, xuất hiện trong mấy chục năm gần đây, là xu thế tất yếu của thời đại, có thể bắt buộc phải có các phép biến đổi này.

5). Điều chế

Đầu ra của bộ mã hoá kênh là chuỗi các ký hiệu nhị phân, chúng ta có thể ký hiệu bởi véctơ **V**, có m phần tử - $\mathbf{V}(v_1, v_2,..., v_m)$ và biểu diễn tín hiệu tương ứng với **V** được gọi là *tín hiệu băng cơ sở*. Tín hiệu băng cơ sở là tín hiệu mà phổ tần số của nó với phần lớn năng lượng tập trung ở vùng tần số thấp, quanh gốc toạ độ ($f = 0$), có thể coi gần đúng là tín hiệu có phổ tần số từ 0 đến F_{max} nào đó (Vấn đề phổ tần số của các tín hiệu băng cơ sở được nghiên cứu trong chương 5). Các tín hiệu băng cơ sở không có khả năng thích ứng với kênh truyền và không thể tự chúng truyền lan đi xa được.

Điều chế có chức năng làm cho tin tức nguồn, hoặc chuỗi các ký hiệu nhị phân phối hợp với đặc tính kênh truyền, có thể truyền đồng thời tín hiệu từ nhiều nguồn tin tức và tín hiệu sau điều chế có thể truyền đi khoảng cách xa.

Điều chế thực hiện biến đổi tín hiệu nguồn s(t), hoặc véctơ **V** (đều là tín hiệu băng cơ sở) thành ra *tín hiệu dải thông*. Tín hiệu dải thông là tín hiệu mà phổ tần số của nó với phần lớn năng lượng tập trung quanh *tần số mang fc*.

Ngày nay, tồn tại nhiều phương pháp và các mô hình khác nhau để điều chế. Chúng có *các đặc tính* khác nhau, cũng như *các hiệu quả* khác nhau, chúng ta sẽ nghiên cứu trong chương 6.

6). Kênh truyền

Kênh truyền là môi trường để truyền lan tín hiệu. Dạng vật chất cụ thể của kênh truyền rất đa dạng, chẳng hạn:

- Đường dây đôi,
- Cáp đồng trục,
- Cáp quang, v.v...,

Trong nhiều trường hợp khác, kênh truyền là *khoảng không gian* giữa thiết bị phát và thiết bị thu.

Trên kênh truyền ngoài *tín hiệu có ích*, ký hiệu S(t, **V**), còn tồn tại *nhiễu*, ký hiệu n(t). Nhiễu có thể phân biệt thành hai loại chính: *tạp âm* và *can nhiễu*.

Có thể phân loại *nguồn gốc của tạp âm* như sau:

- *Tạp âm khí quyển* được sinh ra do sự phóng điện, do thay đổi các tham số của môi trường, như nhiệt độ, độ ẩm, mật độ không khí, độ cao và mật độ các lớp điện tích (lớp E, D, F), v.v...Khi truyền sóng trong *khoảng không gian* có thể phát sinh *hiện tượng pha đinh* làm cho biên độ của tín hiệu *thăng giáng ngẫu nhiên* hoặc chậm, hoặc nhanh theo thời gian,
- *Tạp âm vũ trụ* do hoạt động của mặt trời, các vì sao, v.v...Đôi khi có thể xuất hiện bão từ, gây gián đoạn liên lạc,
- *Tạp âm nội bộ* của hệ thống truyền tin, trong đó có tạp âm nhiệt, tạp âm do các khâu, khối làm việc không như thiết kế,v.v...
- *Tạp âm nhân tạo* do các hoạt động công nghiệp của con người, như truyền tải điện năng, do hàn điện, đánh lửa của buzi ôtô, xe máy,v.v...

Can nhiễu: can nhiễu vô tuyến là một trong những vấn đề quan trọng bậc nhất trong thiết kế, khai thác, bảo trì các hệ thống truyền tin. Do sự tăng trưởng nhanh chóng số lượng các hệ thống truyền tin vô tuyến, cho nên không thể bảo đảm là một hệ thống nào đó hoạt động không gây nhiễu, hoặc không bị nhiễu từ các hệ thống khác. Nhiều giải pháp kỹ thuật khác nhau đã được xây dựng để hạn chế can nhiễu.

Do có nhiễu, nên tín hiệu tại điểm thu có dạng:

$$U(t) = \varepsilon.S[(t-\tau), \mathbf{V}^*] + n(t)$$

trong đó: ε - đặc trưng cho suy giảm năng lượng tín hiệu khi đi qua kênh truyền. ε là đại lượng ngẫu nhiên, phụ thuộc vào thời gian t, vào tần số f, vào địa hình, v.v...;

τ - thời gian giữ chậm tín hiệu, cũng là đại lượng ngẫu nhiên;

Dấu (*) chỉ sự sai lệch của véctơ **V** giữa đầu vào với đầu ra kênh truyền, do nhiễu;

n(t) - gọi là *nhiễu cộng* (còn có cả khái niệm *nhiễu nhân*).

Chi tiết hơn về vấn đề truyền sóng qua khoảng không gian được nghiên cứu trong môn học "Trường - truyền sóng".

Một vấn đề thực tiễn khác, không quan hệ tới can nhiễu và tạp âm, đó là vấn đề *dải thông của kênh truyền không đủ khi truyền các ký hiệu tốc độ cao*. Hoặc một vấn khác cũng tương tự như trên, mặc dù bản chất khác xa nhau, là vấn đề truyền tin trong *môi trường đa đường*, chúng ta gọi là *nhiễu đa đường*. Lúc này khi truyền qua kênh, các ký hiệu tại điểm thu có xu hướng kéo dài ra và chồng lấn lên nhau, được gọi là *nhiễu giao thoa giữa các ký hiệu- ISI* (Inter Symbol Interference). Hiện tượng này gây cho mạch quyết định nhầm và phát sinh lỗi.

Thiết bị thu. Kết thúc của mọi kênh truyền là thiết bị thu (xem hình.1), với chức năng thu nhận tín hiệu và khôi phục lại các tin tức đã gửi đi từ nguồn tin. Thiết bị thu với khả năng thu nhận chính xác cực đại theo dao động U(t), khôi phục lại tin tức nguồn s(t), hoặc véctơ nguồn **S**. Tin tức thu nhận (khôi phục) do có nhiễu, nên sai lệch so với tin tức đã gửi đi. Tin tức thu nhận sẽ được gọi là *đánh giá* (vì thế có tên gọi tin tức đánh giá) và được ký hiệu là $\tilde{s}(t)$, hoặc $\tilde{S}$. Quá trình đánh giá tin tức theo dao động thu nhận trong trường hợp chung liên quan đến một vài giai đoạn, được trình bày dưới đây.

7). Giải điều chế

Để khôi phục lại tin tức nguồn, hay đánh giá $\tilde{s}(t)$, hoặc $\tilde{S}$, thiết bị thu cần phải phụ thuộc theo dao động thu nhận U(t) và với hiểu biết của mình về *dạng tín hiệu* và *khả năng điều chế* khi phát, để nhận ra *đánh giá* của chuỗi ký hiệu mã $\tilde{V}$, được gọi là *chuỗi ký hiệu mã thu nhận*, ký hiệu **P**. Thủ tục này được gọi là *giải điều chế*. Đồng thời thủ tục giải điều chế cần phải thực hiện sao cho chuỗi ký hiệu mã thu nhận **P** khác biệt cực tiểu so với chuỗi ký hiệu mã phát **V**. Ở đây, tuỳ theo cách đặt vấn đề và tuỳ theo khả năng giải quyết bài toán giải điều chế dao động thu nhận U(t), xuất hiện các bài toán *thu tối ưu tín hiệu* trong nền nhiễu (phát hiện tối ưu, phân biệt tối ưu từ một vài tín hiệu, v.v...). Vấn đề này chúng ta sẽ nghiên cứu ở các môn học khác.

8). Giải mã kênh

Chuỗi ký hiệu mã thu nhận **P**, trong trường hợp chung, khác biệt với chuỗi ký hiệu mã phát **V** (ở đầu ra của mã hoá kênh), do có nhiễu trên kênh truyền. Số lượng lỗi trong chuỗi **P** phụ thuộc vào mức độ nhiễu trên kênh truyền, tốc độ truyền (trong sự phù hợp với dải thông của kênh truyền), dạng kênh truyền, chọn dạng điều chế và tất nhiên phụ thuộc vào khả năng thu (giải điều chế) dao động U(t). *Bài toán giải mã kênh* là *phát hiện* và tuỳ thuộc khả năng *sửa* các lỗi. Thủ tục phát hiện và sửa lỗi trong chuỗi ký hiệu mã thu nhận **P** được gọi là *giải mã kênh*. Ngày nay tồn tại hàng loạt các phương pháp (các thuật toán) và kèm theo đó là hàng loạt các mô hình giải mã kênh, chúng ta sẽ nghiên cứu chủ đề này trong chương 3.

9). Giải mã mật

Nếu trong mô hình kênh truyền có sử dụng mã hoá bảo mật, thì sau khi giải mã kênh sẽ tiếp đến thủ tục giải mã mật. Từ văn bản mã thu nhận được (cũng gọi là *đánh giá của văn bản mã*) và

tuỳ thuộc theo *khoá đồng thoả thuận* của hai phía, thủ tục giải mã mật có chức năng khôi phục lại *văn bản rõ* (đánh giá văn bản rõ). Tuỳ thuộc vào phương pháp (thuật toán) được chọn lựa khi mã hoá, mà có thủ tục giải mật mã tương ứng. Ở đây còn thêm thủ tục trao đổi khoá cho từng phiên liên lạc.

10). Giải mã nguồn

Từ chuỗi ký hiệu văn bản rõ, hoặc từ chuỗi ký hiệu mã (sau giải mã kênh), giải mã nguồn có chức năng khôi phục (cũng là quá trình đánh giá) *tin tức* $\tilde{s}(t)$, hoặc véctơ $\tilde{S}$. Bản chất của bài toán giải mã nguồn, nói chung, là bài toán *giải nén*. Tuỳ thuộc vào số liệu được nén và tuỳ thuộc thuật toán nén được sử dụng, mà quá trình giải nén là quá trình biến đổi ngược. Tuy nhiên, đánh giá tin tức $\tilde{s}(t)$ chỉ là xấp xỉ, còn toàn bộ quá trình đánh giá $\tilde{S}$ phải có độ chính xác cao (xem mục dưới về BER).

Để kết thúc mục này, độc giả cần chú rằng, trong nhiều sách " Cơ sở lý thuyết truyền tin" khác, có đề cập tới sơ đồ hệ thống truyền tin với nhiều chức năng hơn nữa. Tuy nhiên, vì điều ấy có thể gây quá tải cho người đọc, cho nên chúng tôi chỉ trình bày những nội dung gì, mà môn học có thể với tới.

3. CÁC YÊU CẦU CƠ BẢN CỦA HỆ THỐNG TRUYỀN TIN

Các hệ thống truyền tin khác nhau, có các yêu cầu cụ thể khác nhau. Tuy nhiên, chúng đều có các yêu cầu chung sau đây:

1). Tốc độ truyền tin. Truyền tin nhanh là một yêu cầu của mọi hệ thống truyền tin, nó được đặc trưng bằng tốc độ truyền tin, ký hiệu là R[bit/sec]. Tốc độ truyền tin phụ thuộc vào tốc độ sản sinh tin của nguồn, cũng ký hiệu là R[bit/sec]; phụ thuộc vào mức độ nhiễu trên kênh truyền và bị hạn chế bởi khả năng thông qua (hay dải thông) của kênh truyền.

2). Dung lượng của kênh truyền, ký hiệu là C [bit/sec]. Dung lượng kênh truyền được định nghĩa là tốc độ truyền cực đại, $C = R_{max}$. Dung lượng của kênh truyền phụ thuộc vào bản chất vật lý của kênh truyền và vào các kỹ thuật điều chế/giải điều chế tiên tiến được sử dụng trong hệ thống truyền tin. Nó được hiểu là khả năng thông qua cực đại của kênh truyền, mà khi truyền với tốc độ đó, hệ thống truyền tin vẫn làm việc theo các tiêu chuẩn đã định trước (chẳng hạn, tỉ số bit lỗi).

3). Độ chính xác. Khi truyền trên kênh có nhiễu, hệ thống truyền tin phải có khả năng chống nhiễu. Độ chính xác được đặc trưng bằng tỉ số lỗi bit:

Tỉ số lỗi bit- BER (Bit Erro Ratio) = (tổng số bit lỗi/tổng số bit đã truyền), xác định trong khoảng thời gian đo.

Với các hệ thống truyền tin khác nhau, có yêu cầu về BER khác nhau. Chẳng hạn, hệ thống điện thoại cố định yêu cầu BER $\leq 10^{-3}$; hệ thống truyền số liệu yêu cầu BER $\leq 10^{-6}$; trong khi đó hệ thông tin quang bảo đảm được BER $\leq 10^{-10}$.

*4). **An toàn**.* Mọi hệ thống truyền tin đều phải bảo đảm chức năng an toàn theo nhiều mức độ khác nhau. An toàn thông tin là yêu cầu của mọi người sử dụng. Nghĩa của thuật ngữ an toàn rất rộng, không chỉ là mật mã, hay kiểm soát quyền truy nhập, hoặc có khả năng phòng, chống virut, v.v...

*5). **Độ tin cậy**.* Độ tin cậy của hệ thống kỹ thuật, nói chung, là tính chất của hệ thống có khả năng hoàn thành những công việc được giao và duy trì các tham số kỹ thuật trong một giới hạn định trước. Độ tin cậy của hệ thống kỹ thuật được đặc trưng bởi các chỉ tiêu của nó, chẳng hạn, tính ít hỏng, tính thuận lợi trong khai thác sử dụng, có tuổi thọ cao, v.v... Để hệ thống kỹ thuật có độ tin cậy cao, đòi hỏi hàng loạt các giải pháp kỹ thuật, trong đó rất thường xuyên sử dụng *dự trữ*.

*6). **Tính kinh tế**.* Vấn đề kinh tế vừa là mục đích vừa là điều kiện để hoạt động của con người. Xu hướng chung của mọi hệ thống kỹ thuật là cung cấp cho con người chất lượng các dịch vụ ngày càng cao, với giá thành ngày càng rẻ. Tuy nhiên, cũng phải thấy rằng giữa giá cả và chất lượng thiết bị, dịch vụ luôn có mối quan hệ tương ứng với nhau.

4. NỘI DUNG MÔN HỌC

Như đã nhận thấy trong các mục trên đây, vừa trình bày mục đích môn học, vừa trình bày các nội dung của môn học. Từ đó chúng ta thấy môn học này có hai nội dung chính:

1. Lý thuyết mã: gồm các chương 1, 2, 3 và 4;

2. Lý thuyết tín hiệu: gồm các chương 5 và 6.

5. BỔ TÚC TOÁN: LÝ THUYẾT XÁC SUẤT, ĐẠI LƯỢNG NGẪU NHIÊN, QUÁ TRÌNH NGẪU NHIÊN

Lý thuyết xác suất, đại lượng ngẫu nhiên, quá trình ngẫu nhiên là cơ sở toán học của môn học này. Vì thế, để có khả năng tiếp thu môn học, bắt buộc mọi sinh viên phải nắm vững đối tượng toán học này. Tuy nhiên, vì độ dài cuốn sách có hạn, nên chúng tôi không trình bày các vấn đề này; độc giả có thể tham khảo ở nhiều tài liệu khác.

CHƯƠNG 1

LÝ THUYẾT THÔNG TIN

Trong chương này, chúng ta sẽ nghiên cứu những vấn đề cơ bản của hệ thống truyền tin bằng lý thuyết toán học thuần tuý (trừ mục cuối có sử dụng kiến thức của kỹ thuật viễn thông). Cơ sở toán học của chương là lý thuyết xác suất và quá trình ngẫu nhiên. Các vấn đề cơ bản của mọi hệ thống truyền tin là vấn đề truyền thông tin qua kênh trong điều kiện không có nhiễu và có nhiễu; vấn đề tốc độ truyền tin, và khả năng thông qua của các loại kênh truyền

1. ĐỘ ĐO LƯỢNG THÔNG TIN CỦA TIN TỨC RỜI RẠC

Các hệ thống truyền tin được sử dụng để truyền tin tức từ người gửi tới người sử dụng. Tuy nhiên, không phải mọi nội dung bất kỳ của tin tức đều chứa đựng thông tin. *Thông tin, đó là tổng hợp những tri thức về đối tượng, hoặc hiện tượng, mà chúng làm tăng hiểu biết của người nhận về đối tượng, hoặc hiện tượng đó.*

Trong lý thuyết toán học của truyền tin (lý thuyết thông tin) chỉ ra rằng trong một tin x_i nào đó có lượng thông tin $I(x_i)$ không phụ thuộc vào nội dung cụ thể, mức độ quan trọng v.v.., mà phụ thuộc vào tin này được chọn như thế nào từ tổ hợp chung các khả năng của các tin.

Trong các điều kiện thực tiễn, việc chọn một tin cụ thể được diễn ra với một xác suất tiền nghiệm $p(x_i)$ nào đó. Xác suất này càng nhỏ, thì lượng thông tin chứa đựng trong tin này càng lớn.

Khi xác định lượng thông tin, người ta xuất phát từ các yêu cầu sau:

1. Lượng thông tin cần phải có đặc tính cộng: lượng thông tin của các tin không phụ thuộc lẫn nhau, cần phải bằng lượng thông tin tổng của các tin.
2. Lượng thông tin về sự kiện chắc chắn ($p(x_i) = 1$) cần phải bằng không, bởi vì các tin như thế không làm tăng hiểu biết của chúng ta về đối tượng hoặc hiện tượng đó.

Các yêu cầu trên phù hợp với việc xác định lượng thông tin riêng của tin nào đó qua dạng logarit như sau:

$$I(x_i) = \log\frac{1}{p(x_i)} = -\log p(x_i) \tag{1.1}$$

Thông thường cơ số của logarit được lấy là 2, hoặc e :

$I(x_i) = -\log_2 p(x_i)$ đơn vị nhị phân của lượng thông tin là [bit].

$I(x_i) = -\ln p(x_i)$ đơn vị logarit tự nhiên của lượng thông tin là [nit].

1 bit là lượng thông tin riêng của tin có xác suất bằng 1/2, và được xác định:

$$I(x_i) = -\log_2\frac{1}{2} = \log_2 2 = 1\,bit\,.$$

Khi áp dụng logarit tự nhiên, thì 1nit được xác định là lượng thông tin riêng của tin có xác suất bằng 1/e:

$$I(x_i) = -\ln\frac{1}{e} = \ln e = 1\,nit$$

Thông thường cho rằng trong thực tiễn việc truyền, xử lí, lưu giữ thông tin ở dạng nhị phân: ký hiệu nhị phân, nguồn nhị phân, kênh nhị phân..., vì vậy đơn vị đo lượng thông tin được sử dụng là [bit].

Khi xác định lượng thông tin của một tin, chúng ta cho rằng tin là phần tử nhỏ bé nhất, không thể chia nhỏ hơn được nữa. Như vậy, tin có thể là các ký tự, hoặc trong dạng nhị phân là các ký hiệu "1" hoặc "0"- có nơi người ta cũng gọi là ký tự. Tổ hợp của các ký tự gọi là từ, cũng tương tự như vậy, tổ hợp của các ký hiệu nhị phân cũng được gọi là từ.

2. ENTROPY CỦA NGUỒN RỜI RẠC VỚI CÁC TIN ĐƯỢC CHỌN KHÔNG PHỤ THUỘC VÀO NHAU

Trong lý thuyết thông tin, cần phải biết không chỉ lượng thông tin của từng tin riêng biệt $I(x_i)$, mà còn cần phải biết lượng thông tin trung bình của các tin trong tin tức, được tạo ra từ một nguồn tin nào đó.

Nếu có một tập tin X (tập toàn phần), bao gồm N tin $x_1, x_2, ..., x_N$ với xác suất $p(x_1), p(x_2), ..., p(x_N)$, thì lượng thông tin trung bình của mỗi tin H(X), còn được gọi là entropy của nguồn tin, được xác định theo công thức:

$$H(X) = E\{I(x_i)\} = \sum_{i=1}^{N} p(x_i)I(x_i) = -\sum_{i=1}^{N} p(x_i)\log_2 p(x_i) \qquad (1.2)$$

Đơn vị của lượng thông tin trung bình là [bit/ký tự], hoặc [bit/ký hiệu].

Entropy đặc trưng cho nguồn tin theo quan điểm độ bất định, khi so sánh các nguồn tin khác nhau.

Các đặc tính của entropy H(X):

1. Entropy là một đại lượng không âm $H(X) \geq 0$. Điều này thấy rõ khi xem xét công thức (1.2).
2. Độ bất định của tập tin càng lớn, thì entropy càng lớn. Độ bất định cực đại khi các tin xuất hiện đồng xác suất, với xác suất mỗi tin $p(x_1) = p(x_2) = ... = p(x_N) = 1/N$.

   Trong trường hợp này:

   $$H(X) = H(X)_{max} = -\sum_{i=1}^{N}\frac{1}{N}\log_2\frac{1}{N} = \log_2 N \qquad (1.3)$$

   Nghĩa là cực đại của entropy bằng $\log_2$ của kích cỡ N của tập tin. Với tập tin nhị phân $N = 2$, $H(X)_{max} = \log_2 2 = 1$ bit.

3. Độ bất định cực tiểu, nếu một tin nào đó của nguồn tin có xác suất bằng 1, các tin còn lại có xác suất bằng 0: $p(x_1) = 1$; $p(x_2) = p(x_3) = ... = p(x_N) = 0$. Trong trường hợp này $H(X) = H(X)_{min} = 0$.

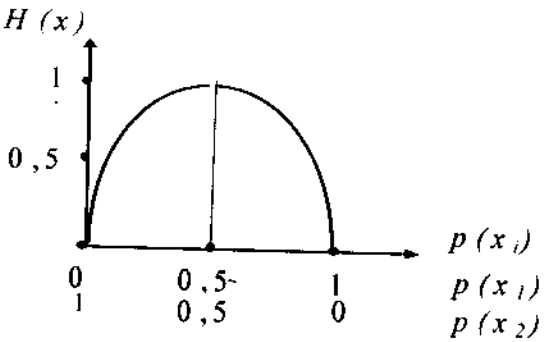
Với một nguồn tin bất kỳ, chúng ta luôn luôn có:

$$0 \le H(X) \le H(X)_{max} = \log_2 N$$

Ta xem xét một nguồn nhị phân, nghĩa là nguồn chỉ bao gồm 2 loại tin x_1, x_2 với $p(x_1) + p(x_2) = 1$. Khi đó:

$$H(X) = -p(x_1).\log p(x_1) - p(x_2)\log p(x_2)$$
$$= -p(x_1)\log p(x_1) - [1 - p(x_1)].\log[1 - p(x_1)] \qquad (1.4)$$

Sự phụ thuộc của H(X) vào xác suất chọn, đối với nguồn nhị phân được trình bày trên hình 1.1.



Hình 1.1. Sự phụ thuộc của H(X) vào xác suất $p(x_i)$ của nguồn nhị phân.

4. Giả thử ở đầu ra của một nguồn nhị phân, ta có thiết bị dùng để nhóm n ký hiệu tạo thành một từ. Khi đó lực lượng các từ $N = 2^n$, hơn nữa các từ xuất hiện đồng xác suất. Trong trường hợp này:

$$H(X)_{max} = \log N = \log_2 2^n = n\,bit \qquad (1.5)$$

Như vậy, khi nhóm các ký hiệu để tạo thành từ dẫn đến làm tăng entropy lên n lần, bởi vì khi đó mỗi từ có n ký hiệu nhị phân. Chính điều này đã chứng minh cho đặc tính cộng của entropy.

Thuật ngữ "entropy" lần đầu tiên được sử dụng bởi nhà bác học C. Shannon trong công trình "Lý thuyết toán học của truyền tin " xuất bản năm 1948. Ngày nay đã trở thành cơ sở của lý thuyết truyền tin.

3. ENTROPY CỦA NGUỒN RỜI RẠC VỚI CÁC TIN PHỤ THUỘC LẪN NHAU

Trên đây chúng ta đã xét entropy của tin tức với các tin có xác suất xuất hiện không phụ thuộc vào nhau. Mục này, chúng ta sẽ khảo sát trường hợp phức tạp hơn, khi có nguồn tin tức có liên hệ tương quan. Chẳng hạn, ta có một nguồn sản ra các ký tự (ký hiệu), mà ký tự được sản sinh ra tiếp theo phụ thuộc vào n ký tự đã sản sinh ra trước đó. Mô hình toán học của một nguồn như thế được gọi chuỗi Markov bậc n, trong đó xác suất xuất hiện của ký tự tiếp theo phụ thuộc vào n giá trị xác suất đã có trước đó, mà không phụ thuộc vào nhiều hơn. Ta có thể biểu diễn:

$$p(x_i / x_{i-1}, x_{i-2}, ..., x_{i-n}) = p(x_i / x_{i-1}, x_{i-2}, ..., x_{i-n}, ..., x_{i-n-c}) \quad (1.6)$$

với c là hằng số dương bất kỳ.

Nếu số lượng các ký tự của nguồn bằng N, thì số lượng các ký tự phụ thuộc vào nhau cần được tính khi xác định xác suất của ký tự tiếp theo, bằng bậc n của nguồn, thì mỗi một ký tự có thể xuất hiện trước $M = N^n$ ký tự kết hợp khác nhau (trạng thái của nguồn), bị ảnh hưởng đến xác suất xuất hiện của ký tự x_i tiếp theo ở đầu ra của nguồn. Còn xác suất xuất hiện trong tin tức bất kỳ gồm N khả năng của các ký tự được xác định bằng xác suất có điều kiện (1.6) với sự tính toán của các ký tự xuất hiện trước đó, nghĩa là với sự tính toán M trạng thái có thể. Những trạng thái này được ký hiệu là $q_1, q_2, ..., q_M$.

Chúng ta xét 2 thí dụ đơn giản

Thí dụ 1: Giả thử có một nguồn nhị phân (số lượng ký hiệu N = 2), chẳng hạn nguồn chỉ có 2 ký tự a, b; bậc của nguồn n = 1. Khi đó số lượng các trạng thái của nguồn $M = N^n = 2^1 = 2$ (gọi các trạng thái của chúng là q_1, q_2). Trong trường hợp này xác suất xuất hiện của các ký tự a, b sẽ được xác định theo các xác suất có điều kiện:

$$p(a / q_1 = a), \quad p(a / q_2 = b), \; p(b / q_1 = a), \; p(b / q_2 = b),$$

q_1 = a, trạng thái thứ nhất q_1;

q_2 = b, trạng thái thứ hai q_2.

với các xác suất $p(q_1) = p(a)$, $p(q_2) = p(b)$,

Thí dụ 2: Giả thử, ta có nguồn tin như thí dụ trên N = 2 (các ký tự a, b), tuy nhiên, bậc của nguồn n = 2. Khi đó $M = 2^2 = 4$. Có 4 khả năng trạng thái: (a, a) = q_1, (a, b) = q_2, (b, a) = q_3, (b, b) = q_4.

Trong trường hợp này có các xác suất có điều kiện sau:

$$p(a/q_1), p(a/q_2), p(a/q_3), p(a/q_4), p(b/q_1),...$$

Các xác suất trạng thái được xác định bằng: $p(q_1) = p(a, a)$; $p(q_2) = p(a, b)$; $p(q_3) = p(b, a)$; $p(q_4) = p(b, b)$.

Entropy của nguồn rời rạc ergodic được xác định trong hai quá trình:

1. Tính toán entropy của nguồn trong mỗi trạng thái từ M trạng thái, coi các trạng thái này đã biết:

Đối với trạng thái q_1 $H(X/q_1) = -\sum_{i=1}^{N} p(x_i / q_1) \log p(x_i / q_1)$

Đối với trạng thái q_2 $H(X/q_2) = -\sum_{i=1}^{N} p(x_i / q_2) \log p(x_i / q_2)$

.....

Đối với trạng thái q_M $H(X/q_M) = -\sum_{i=1}^{N} p(x_i / q_M) \log p(x_i / q_M)$

2. Sau đó tìm H(X) bằng trung bình theo tất cả các trạng thái q:

$$H(X) = \sum_{i=1}^{M} p(q_i) H(X / q_i)$$

Cuối cùng chúng ta nhận được:

$$H(X) = -\sum_{j=1}^{M} p(q_j) \sum_{i=1}^{N} p(x_i / q_j) \log p(x_i / q_j) \qquad (1.7)$$

Khi có sự liên hệ tương quan giữa các ký tự (ký hiệu) trong nguồn ergodic thì entropy sẽ giảm, bởi vì khi đó giảm độ bất định của việc chọn các ký tự và trong hàng loạt các trường hợp một số ký tự có thể đoán ra được theo các ký tự biết trước hoặc đứng gần.

4. ĐỘ DƯ THỪA CỦA NGUỒN TIN

Như đã chỉ ra trong mục 2 và 3, entropy đạt giá trị cực đại khi các tin x_i của nguồn có phân bố đồng xác suất và không có liên hệ tương quan với nhau. Khi phân bố xác suất không đều hoặc các tin có liên hệ tương quan với nhau thì entropy giảm.

Entropy của nguồn càng tiến gần giá trị cực đại, thì nguồn làm việc càng hợp lý.

Để đánh giá về việc nguồn sẽ sử dụng các tin của mình tốt như thế nào, người ta đưa vào khái niện độ dư thừa của nguồn tin tức:

$$g_M(\%) = \frac{H(X)_{\max} - H(X)}{H(X)_{\max}}.100\%$$

hoặc là:

$$g_M(\%) = \left[1 - \frac{H(X)}{\log N}\right].100\% \qquad (1.8)$$

Xét ý nghĩa của độ dư thừa $g_M(\%)$:

* Sự tồn tại độ dư thừa trong nguồn tin tức dẫn đến làm tăng tải cho kênh truyền tin, khi đó phải truyền đi các tin dư thừa, không mang thông tin (chúng có thể dự đoán ra được, mà không cần truyền). Trong trường hợp này, trước khi truyền tin hoặc lưu giữ chúng, người ta sử dụng mã kinh tế hay mã nén tin tức, để giảm độ dư thừa của văn bản mã (xem chương 2)

* Tuy nhiên, độ dự thừa của tin tức, trong trường hợp khác, lại có ích, nó được sử dụng để làm tăng độ tin cậy của quá trình truyền tin. Chẳng hạn, trong kênh truyền có nhiễu, để xây dựng mã chống nhiễu, người ta bổ sung thêm độ dư thừa trong tin tức gửi đi, nhờ nó mà khi giải mã có thể phát hiện ra lỗi và mạnh hơn là sửa lỗi (xem chương 3).

* Độ dư thừa lớn có trong mọi ngôn ngữ bất kỳ. Chẳng hạn, độ dư thừa trong nhiều ngôn ngữ cỡ 50%. Nhờ có độ dư thừa sẽ bảo đảm khả năng hiểu được tiếng nói, khi có khuyết tật trong phát

âm, hoặc xuất hiện lỗi khi truyền tín hiệu tiếng nói qua kênh liên lạc. Độ dư thừa làm cho các ngôn ngữ trở nên đa dạng, phong phú, nhiều ngữ nghĩa.

5. NĂNG SUẤT CỦA NGUỒN TIN

Năng suất (tốc độ sản sinh tin) của nguồn tin được xác định bằng số lượng thông tin truyền qua kênh trong một đơn vị thời gian. Năng suất của nguồn nhị phân được đo bằng [bit/sec]. Nếu tất cả phần tử của nguồn tin tức có độ dài τ như nhau, thì năng suất của nguồn được xác định:

$$R = \frac{H(X)}{\tau} \tag{1.9}$$

Nếu các phần tử của tin tức có độ dài τ_i khác nhau, thì cần phải xác định độ dài trung bình $\bar{\tau}$, là kỳ vọng toán học của các độ dài τ_i :

$$\bar{\tau} = \sum_{i=1}^{N} \tau_i p(\tau_i)$$

Trong công thức trên, chúng ta coi $p(\tau_i) = p(x_i)$ (là xác suất của tin thứ i), bởi vì các xác suất này bằng nhau. Kết quả ta nhận được:

$$\bar{\tau} = \sum_{i=1}^{N} \tau_i p(x_i) \tag{1.10}$$

và năng suất của nguồn được xác định:

$$R = \frac{H(X)}{\bar{\tau}} \tag{1.11}$$

Cực đại năng suất của nguồn rời rạc bằng:

$$R_{\max} = \frac{H(X)_{\max}}{\bar{\tau}} = \frac{\log N}{\bar{\tau}} \tag{1.12}$$

Đối với nguồn nhị phân, và các phần tử có cùng độ dài (N = 2; $\bar{\tau} = \tau$), chúng ta có:

$$R_{\max} = \frac{\log_2 2}{\tau} = \frac{1}{\tau} \quad \text{bit/sec} \tag{1.13}$$

Khi nhóm n ký hiệu để tạo thành từ, lúc đó $N = 2^n$, $\bar{\tau} = n\tau$. Chúng ta có:

$$R_{\max} = \frac{\log_2 2^n}{n\tau} = \frac{1}{\tau} \quad \text{bit/sec}$$

trùng với công thức (1.13).

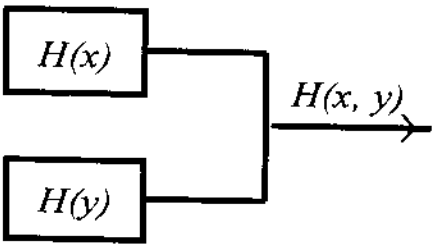
Như vậy, bằng cách nhóm các ký hiệu tạo thành từ, để tăng năng suất của nguồn thì không được, vì khi ấy cả entropy và độ dài của tin tức đều tăng như nhau (n lần).

Để tăng năng suất của nguồn có thể bằng cách giảm độ dài các ký hiệu của nguồn. Tuy nhiên, khả năng này bị hạn chế do dải thông của kênh truyền là hữu hạn; lúc này giảm độ dài của các ký hiệu đồng nghĩa với việc tăng hiệu ứng giao thoa giữa các ký hiệu (ISI). Bởi vậy, năng suất

của nguồn có thể tăng bằng cách sử dụng hiệu quả dải thông kênh truyền, như: sử dụng điều chế tín hiệu nhiều mức, sử dụng công nghệ truyền song song (chẳng hạn, điều chế ghép kênh phân chia theo tần số trực giao - OFDM).

6. ENTROPY ĐỒNG THỜI CỦA HAI NGUỒN TIN TỨC

Giả sử có hai nguồn rời rạc, với các entropy H(X) và H(Y), có kích cỡ các nguồn là N và L, được nối với nhau, tạo thành nguồn phức tạp như hình 1.2 dưới đây:



Hình 1.2. Entropy đồng thời của hai nguồn tin.

Các phần tử đầu ra của nguồn đồng thời bao gồm: tin x_i (với i = 1÷ N), tin y_j (với j = 1÷ L). Entropy của đồng thời được xác định:

$$H(X,Y) = E\{I(x_i, y_j)\} \tag{1.14}$$

trong đó $I(x_i, y_j)$ lượng thông tin giao của 2 tin x_i và y_j và được xác định:

$$I(x_i y_j) = -\log p(x_i y_j) \tag{1.15}$$

Cuối cùng entropy của nguồn đồng thời được xác định:

$$H(X,Y) = -\sum_{i=1}^{N}\sum_{j=1}^{L} p(x_i y_j)\log p(x_i y_j) \tag{1.16}$$

Theo định lý nhân xác suất:

$$p(x_i y_j) = p(x_i).p(y_j / x_i) = p(y_j).p(y_j / x_i)$$

Đặt quan hệ này vào(1.16)chúng ta nhận được:

$$H(X,Y) = H(X) + H(Y/X) = H(Y) + H(X/Y) \tag{1.17}$$

Trong đó H(Y/X) - được gọi là entropy có điều kiện của nguồn Y đối với nguồn X, nó chỉ ra tin tức Y có entropy như thế nào, khi đã biết tin tức X, và được xác định:

$$H(Y/X) = -\sum_{i=1}^{N}\sum_{j=1}^{L} p(x_i y_j)\log p(y_j / x_i) \tag{1.18}$$

còn H(X/Y)- được gọi là entropy có điều kiện của nguồn X đối với nguồn Y, cũng tương tự như trên, nó chỉ ra tin tức X có entropy như thế nào, khi đã biết tin tức Y, và được xác định:

$$H(X/Y) = -\sum_{i=1}^{N}\sum_{j=1}^{L} p(x_i y_j)\log p(x_i / y_j) \tag{1.19}$$

Nếu các nguồn không phụ thuộc lẫn nhau, thì:

$$p(y_j / x_i) = p(y_j); p(x_i / y_j) = p(x_i)$$

và có H(Y/X) = H(Y); H(X/Y) = H(X). Trong trường hợp đó:

$$H(X,Y) = H(X) + H(Y) \quad (1.17a)$$

Nếu nguồn phụ thuộc một phần vào nhau, thì:

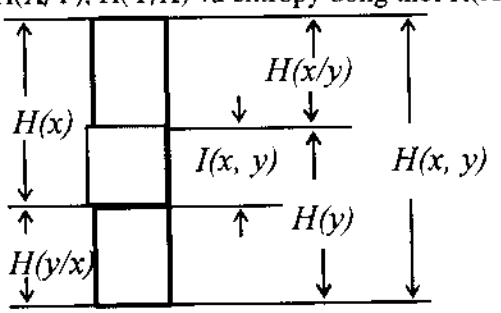
$$H(X,Y) < H(X) + H(Y) \quad (1.17b)$$

Nếu hai nguồn hoàn toàn trùng nhau (X và Y là một, có cùng một lượng thông tin), thì:

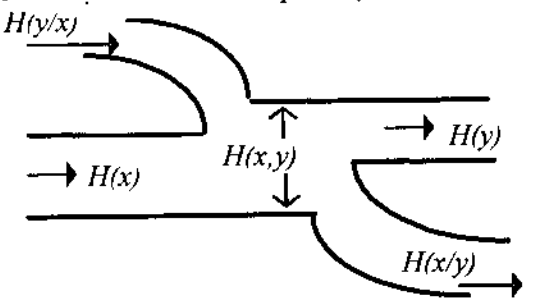
$$H(Y/X) = H(X/Y) = 0, \text{ và có } H(X,Y) = H(X) = H(Y) \quad (1.17c)$$

7. LƯỢNG THÔNG TIN TƯƠNG HỖ CỦA HAI NGUỒN TIN

Trên hình 1.3 chỉ ra các quan hệ giữa các entropy riêng H(X), H(Y), entropy có điều kiện H(X/Y), H(Y/X) và entropy đồng thời H(X,Y). Hình vẽ được dẫn ra từ các quan hệ (1.17).



Hình 1.3. Mối quan hệ giữa các entropy.



Hình 1.4. Mô hình kênh truyền có nhiễu.

Phần của hình được gạch tối, được gọi là lượng thông tin tương hỗ I(X,Y). Nó chỉ ra lượng thông tin trung bình tương hỗ giữa hai nguồn tin. Đại lượng I(X,Y) được xác định như sau:

$$I(X,Y) = E\{I(x_i, y_j)\}$$

trong đó: $I(x_i, y_j) = I(x_i) - I(x_i / y_j) = -\log p(x_i) - [-\log(x_i / y_j)]$;

hoặc $I(x_i, y_j) = I(y_j) - I(y_j / x_i) = -\log p(y_j) - [-\log(y_j / x_i)]$

Sau một vài biến đổi, chúng ta có:

$$I(X,Y) = H(X) - H(X/Y) = H(Y) - H(Y/X) \quad (1.20)$$

* Nếu hai tin tức X, Y hoàn toàn không phụ thuộc vào nhau, thì lượng thông tin tương hỗ:

$$I(X,Y) = 0 \quad (1.20a)$$

* Nếu hai tin tức X, Y hoàn toàn trùng nhau, thì lượng thông tin tương hỗ:

$$I(X,Y) = H(X) = H(Y) \quad (1.20b)$$

Khái niệm lượng thông tin tương hỗ I(X,Y) được sử dụng rộng rãi trong lý thuyết truyền tin. Sau đây ta xét các mô hình phổ thông nhất:

1) Mô hình kênh truyền tin, như hình 1.4: X là tin tức đầu vào kênh truyền, còn Y là tin tức đầu ra kênh truyền. Chúng ta khảo sát hai trường hợp cực đoan:

a) Kênh truyền hoàn toàn không có nhiễu, lúc đó X và Y trùng nhau, ta có quan hệ (1.17c) và (1.20b), tức là: I(X,Y) = H(X) = H(Y)

b) Kênh truyền có nhiễu nặng (người ta thường gọi là hiện tượng *đứt kênh*), lúc đó X và Y hoàn không phụ thuộc vào nhau, ta có quan hệ (1.17a) và (1.20a), tức là I(X,Y) = 0.

c) Khi kênh truyền có một phần nhiễu, thì ta có các quan hệ (1.20)

Với các ý nghĩa như vậy, thì H(X/Y)- chỉ ra lượng thông tin trung bình bị mất mát trên kênh do có nhiễu. Còn H(Y/X)- chỉ ra sai số trung bình về Y, khi đã biết X, sau này chúng ta chứng minh được H(Y/X) là entropy của nhiễu và ký hiệu là H(n).

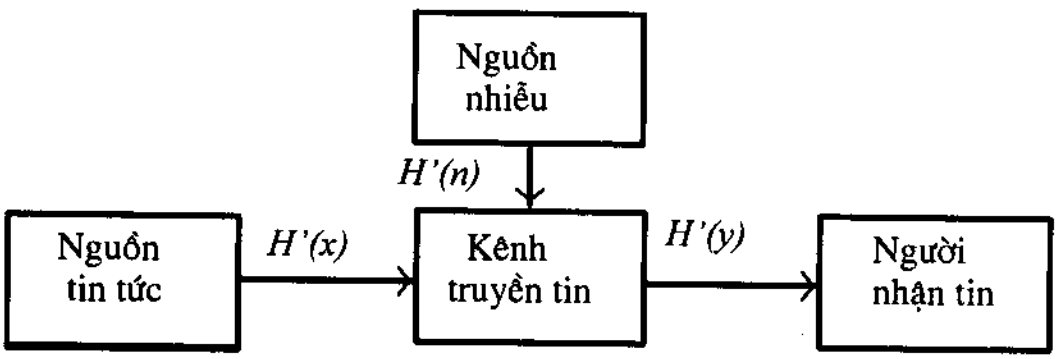
2) Giả thử X, Y là hai tin tức được xuất bản ở hai tờ báo khác nhau, nhưng cùng đề cập về một chủ đề, thì để nhận được lượng thông tin giữa chúng giống nhau nhiều, thì cần các đại lượng H(X/Y) và H(Y/X) là càng nhỏ càng tốt.

8. TỐC ĐỘ TRUYỀN VÀ KHẢ NĂNG THÔNG QUA CỦA KÊNH

Trong hệ thống truyền tin rời rạc, khi không có nhiễu trên kênh, thì lượng thông tin ở đầu ra của kênh hoàn toàn trùng với lượng thông tin ở đầu vào của kênh, tốc độ truyền tin được xác định bằng năng suất của nguồn:

$$R(X,Y) = \frac{H(X)}{\bar{\tau}} = R(X) \text{ [bit/sec]} \tag{1.21}$$

Khi trên kênh truyền có nhiễu (hình 1.5.), thì thông tin bị tổn hao, và tốc độ truyền tin bị giảm đi so với năng suất của nguồn. Đồng thời tin tức ở đầu ra có bổ sung thêm nhiễu.



Hình 1.5. Sơ đồ kênh truyền với nhiễu.

Vì vậy, khi kênh có nhiễu, thì ở đầu ra không phải toàn bộ thông tin như nguồn, mà chỉ là thông tin tương hỗ, do đó tốc độ truyền tin được xác định:

$$R(X,Y) = \frac{I(X,Y)}{\bar{\tau}} \quad \text{[bit/sec]} \tag{1.22}$$

Trên cơ sở của công thức (1.20), chúng ta có:

$$R(X,Y) = \frac{1}{\bar{\tau}}[H(X) - H(X/Y)] = \frac{1}{\bar{\tau}}[H(Y) - H(Y/X)]$$

$$R(X,Y) = R(X) - R(X/Y) = R(Y) - R(Y/X) \tag{1.23}$$

Ở đây: $R(X)$ - năng suất của nguồn;

$R(X/Y)$ - độ tiêu hao của kênh trong một đơn vị thời gian;

$R(Y)$ - entropy của tin tức đầu ra trong một đơn vị thời gian;

$R(Y/X) = R(n)$ - entropy của nhiễu trong một đơn vị thời gian.

Khả năng thông qua của kênh truyền- ký hiệu là C - được gọi là khả năng cực đại của tốc độ truyền tin theo kênh:

$$C = \max R(X,Y) \tag{1.24}$$

Để đạt được mức độ cực đại cần phải tính đến tất cả khả năng của nguồn ở đầu ra và tất cả khả năng mã hoá, cũng như tất cả khả năng xử lý tin ở phía thu.

Như vậy, khả năng thông qua của kênh thông tin liên lạc bằng năng suất cực đại của nguồn ở đầu vào của kênh khi phối hợp hoàn toàn với đặc trưng của kênh, và có tính đến tiêu hao thông tin trên kênh do nhiễu.

Khi kênh không có nhiễu C = max R(X), bởi vì R(X/Y) = 0. Khi sử dụng mã hoá đều với cơ số m, mỗi từ mã có n phần tử, mỗi phần tử có độ dài τ, và kênh không nhiễu, thì:

$$C = \max \frac{H(X)}{\bar{\tau}} = \frac{\log m^n}{n\tau} = \frac{\log m}{\tau}$$

khi m = 2, thì : $C = \dfrac{1}{\tau}$ [bit/sec] (1.25)

Khi xác định khả năng thông qua cực đại của kênh theo công thức (1.25), phải hiểu mỗi ký hiệu có độ dài cực tiểu- $\tau_{\min}$, mà hệ thống truyền tin vẫn làm việc bình thường theo các tiêu chuẩn đã định trước.

Khả năng thông qua C của kênh không chỉ phụ thuộc vào bản chất vật lý của kênh truyền (tức là phụ thuộc cả vào dải thông của kênh được định nghĩa như thế nào), mà còn phụ thuộc vào phương pháp gia công tín hiệu đầu vào, đầu ra của kênh truyền (tức điều chế/giải điều chế), để hệ thống truyền tin vẫn làm việc theo các tiêu chuẩn đã định trước.

Để sử dụng hiệu quả khả năng thông qua của kênh, cần thiết phải phối hợp nó với nguồn thông tin đầu vào. Sự phối hợp như thế cho cả trường hợp kênh không nhiễu và trường hợp kênh có nhiễu, trên cơ sở của hai định lý Shannon:

Định lý thứ nhất (đối với kênh truyền không nhiễu):

Nếu nguồn tin tức có entropy H[bit/ký hiệu], còn kênh truyền có khả năng thông qua C [bit/sec], thì có thể mã hoá tin tức, sao cho việc truyền thông tin trên kênh với tốc độ trung bình gần đến C tuỳ ý, nhưng không vượt được nó.

Chúng ta đã biết năng suất của nguồn và độ dư thừa của nguồn rời xác được xác định phù hợp theo các công thức:

$$R(X) = \frac{H(X)}{\bar{\tau}}, \quad g_M = 1 - \frac{H(X)}{\log N} \tag{1.26}$$

Nếu đem mã hóa tập tin X bằng mã tiết kiệm (xem chương 2), chúng ta sẽ có các định nghĩa:

$$R(X) = \frac{H(X)}{\bar{n}.\tau_0}, \quad g_C = 1 - \frac{H(X)}{\bar{n}} \tag{1.26a}$$

trong đó $\bar{n}$ - chiều dài trung bình của các từ mã, còn τ_0 - chiều dài của một ký hiệu.

Từ các công thức này, rõ ràng rằng để tăng năng suất của nguồn (cũng tức là tăng tốc độ truyền tin) cần phải giảm độ dư thừa (bằng các kỹ thuật mã nén tin tức), cũng tức là bằng cách mã hoá để giảm chiều dài trung bình của các từ mã. Toàn bộ những vấn đề này, chúng ta sẽ đề cập và giải quyết trong chương 2.

Định lý thứ hai (đối với kênh truyền có nhiễu):

Nếu khả năng thông qua của kênh bằng C, còn năng suất của nguồn R(X) < C, thì bằng cách mã hoá phù hợp, có thể truyền thông tin theo kênh liên lạc với tốc độ gần tới C tuỳ ý và với xác suất lỗi nhỏ tuỳ ý. Nếu R(X) > C, thì có thể mã hoá nguồn, nhưng sẽ không tin cậy, để xác suất lỗi nhỏ hơn R(X) - C + ε, ở đây ε- đại lượng nhỏ tuỳ ý.

Không tồn tại khả năng mã hoá, bảo đảm tin cậy, với xác suất lỗi nhỏ hơn R(X) - C.

(*Chú ý*: ở đây cũng như ở nhiều nơi khác của chương này, chúng ta sử dụng ký hiệu R (viết đầy đủ Rate = tốc độ) để chỉ tốc độ. Khi R chỉ tốc độ sản sinh tin của nguồn, thì R có thể tuỳ ý: nhỏ hơn, bằng, hoặc lớn hơn C = khả năng thông qua của kênh. Còn khi R chỉ tốc độ truyền tin trên kênh, thì R bắt phải ≤ C, thực tế R << C !)

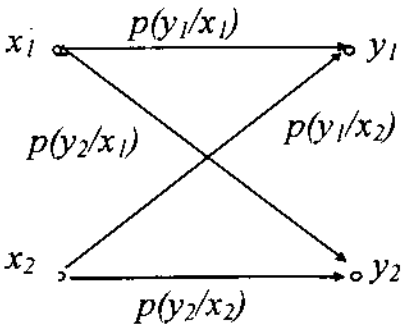
Định lý thứ 2 của Shannon chỉ ra với kênh truyền có nhiễu có thể bằng khả năng mã hoá để giảm lỗi đến *nhỏ tuỳ ý*, nhưng không chỉ ra cách xây dựng các mã như thế. Trong chương 3, chúng ta sẽ nghiên cứu các loại mã như thế, gọi là *mã chống nhiễu*. Tuy nhiên, cũng cần biết rằng khi tiến tới giới hạn của định lý Shannon, thì *thời gian giữ chậm tín hiệu trong các thiết bị mã hoá và giải mã là vô cùng*, vì phải tăng độ dài n của các từ mã. Khi đó xác suất lỗi ở đầu ra kênh truyền tiến tới đại lượng:

$$p = 2^{-n\tau[C - H(X)]} \tag{1.27}$$

Rõ ràng, $p \to 0$, khi $n\tau \to \infty$, như vậy có sự ràng buộc giữa xác suất mắc lỗi khi truyền tin và thời gian truyền bị giữ chậm.

9. KHẢ NĂNG THÔNG QUA CỦA KÊNH TRUYỀN ĐỐI XỨNG ĐỒNG NHẤT

Trong kênh truyền đồng nhất, với các xác suất có điều kiện (xác suất chuyển đổi) $p(y_j/x_i)$. Các trạng thái chuyển đổi của kênh liên lạc nhị phân đồng nhất được biểu diễn trên hình 1.6.



Hình 1.6. Mô hình của kênh nhị phân đồng nhất.

Trên hình vẽ này, x_1 và x_2- các tín hiệu ở đầu vào kênh truyền, còn y_1 và y_2- các tín hiệu ở đầu ra kênh truyền. Nếu gửi đi tín hiệu x_1 và thu được tín hiệu y_1, thì điều đó có nghĩa rằng tín hiệu thứ nhất (có chỉ số 1) đã không bị méo. Nếu gửi đi tín hiệu x_1 và thu được tín hiệu y_2, thì điều đó có nghĩa rằng tín hiệu thứ nhất (có chỉ số 1) đã bị méo. Các xác suất chuyển đổi được chỉ ra trên hình 1.6 . Nếu kênh đối xứng thì các xác suất chuyển đổi từng cặp bằng nhau.

Chúng ta ký hiệu:

$p(y_2/x_1) = p(y_1/x_2) = p$ - các xác suất méo của phần tử tín hiệu,

$p(y_1/x_1) = p(y_2/x_2) = 1 - p$ - các xác suất thu đúng phần tử tín hiệu.

Theo các công thức (1.23), chúng ta có:

$$C = \max \frac{1}{\bar{\tau}}[H(Y) - H(Y/X)] \qquad (1.28)$$

Nếu các tín hiệu x_1 và x_2 có cùng độ dài τ, thì $\bar{\tau} = \tau$. Khi đó khả năng thông qua của kênh sẽ bằng:

$$C = \max \frac{1}{\tau}[H(Y) - H(Y/X)]$$

Trong công thức này maxH(Y) = log m. Đối với kênh nhị phân (m = 2), maxH(Y) = 1. Do đó, chúng ta có:

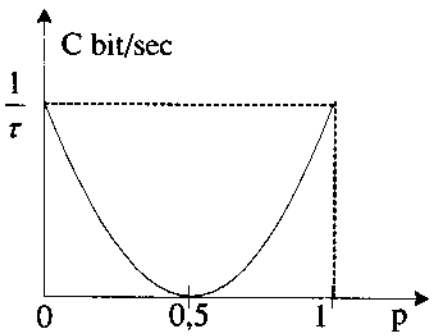
$$C = \max \frac{1}{\bar{\tau}}[1 - H(Y/X)] \qquad (1.29)$$

Điều còn lại là cần phải xác định entropy có điều kiện H(Y/X). Phù hợp với công thức (1.19) đối với nguồn nhị phân, chúng ta có:

$$H(Y/X) = -\sum_{i=1}^{2}\sum_{j=1}^{2} p(x_i y_j)\log p(y_j/x_i) = -\sum_{i=1}^{2}\sum_{j=1}^{2} p(x_i)p(y_j/x_i)\log p(y_j/x_i) =$$
$$= -p(x_1)p(y_1/x_1)\log p(y_1/x_1) - p(x_1)p(y_2/x_1)\log p(y_2/x_1) -$$
$$- p(x_2)p(y_1/x_2)\log p(y_1/x_2) - p(x_2)p(y_2/x_2)\log p(y_2/x_2) =$$
$$= -p(x_1)(1-p)\log(1-p) - p(x_1)\log p - p(x_2)\log p - p(x_2)\log(1-p) =$$
$$= -p\log p - (1-p)\log(1-p).$$

Đặt giá trị này vào công thức (1.29), cuối cùng chúng ta nhận được:

$$C = \frac{1}{\tau}[1 + p\log p + (1-p)\log(1-p)] \tag{1.30}$$



Hình 1.7. Sự phụ thuộc của khả năng thông qua của kênh C vào xác suất lỗi p của kênh nhị phân.

Trên hình 1.7 trình bày sự phụ thuộc của khả năng thông qua của kênh C [bit/sec] vào xác suất lỗi p của kênh nhị phân. Khi xác suất lỗi p = 0, ứng với trường hợp kênh không nhiễu, thì khả năng thông qua của kênh là cực đại và bằng:

$$C = \frac{1}{\tau}\text{ [bit/sec]}$$

Đối với kênh truyền với m > 2 (kênh m phân), khả năng thông qua của kênh được xác định bằng công thức tương tự:

$$C = \frac{1}{\tau}[\log m + p\log\frac{p}{m-1} + (1-p)\log(1-p)] \tag{1.31}$$

Để kết thúc mục này, chúng ta khảo sát một thí dụ. Giả thử, có một nguồn nhị phân với năng suất $R(X) = \frac{1}{\tau}$ = 1000 bit/sec.

Nếu xác suất lỗi của kênh truyền p = 0,01, thì từ 1000 phần tử tín hiệu được truyền trong 1 sec, về trung bình có 990 sẽ truyền qua không lỗi, chỉ có 10 phần tử mắc lỗi. Điều này chỉ ra khả năng thông qua của kênh là 990 bit/sec. Tuy nhiên, nếu tính theo công thức (1.30), cho chúng ta một giá trị nhỏ hơn (C = 919 bit/sec). Tại sao lại như vậy? Vấn đề ở chỗ là khi chúng ta đã nhận được C = 990 bit/sec, nếu như chúng ta đã biết chính xác rằng phần tử nào của tín hiệu bị lỗi.

Thực tiễn thì sự kiện này không thể biết được, điều đó dẫn đến sự kiện 10 phần tử bị lỗi làm hạ thấp mạnh giá trị của tin tức thu và khả năng thông qua của kênh bị giảm mạnh.

Ta xét một thí dụ khác. Nếu p = 0,5, thì từ 1000 phần tử được truyền đi, chỉ có 500 phần tử sẽ không bị lỗi. Tuy nhiên, khả năng thông qua của kênh sẽ không là 500 bit/sec; mà là theo công thức (1.30) cho chúng ta C = 0. Thực vậy, khi p = 0,5 thì tín hiệu truyền theo kênh thực sự đã không diễn ra và kênh truyền có vai trò tương đương như một máy phát tạp.

Khi $p \to 1$, khả năng thông qua của kênh đạt giá trị cực đại. Tuy nhiên, phải hiểu rằng trong trường hợp này tín hiệu đã truyền theo chiều ngược lại.

10. ENTROPY CỦA CÁC ĐẠI LƯỢNG NGẪU NHIÊN LIÊN TỤC VÀ ĐẶC TÍNH CỦA NÓ

Entropy của tín hiệu ngẫu nhiên rời rạc được xác định theo biểu thức (1.2). Đối với các đại lượng ngẫu nhiên liên tục, entropy được xác định bằng công thức tương tự, với sự thay thế xác suất p(x) bằng w(x)dx. Kết quả chúng ta nhận được:

$$H(X) = -\int_{-\infty}^{+\infty} w(x)dx \log[w(x)dx] = -\int_{-\infty}^{+\infty} w(x)[\log w(x) + \log dx]dx$$

Nhưng log của một đại lượng vô cùng nhỏ (dx) bằng âm vô cùng, kết quả chúng ta nhận được:

$$H(X) = \infty - \int_{-\infty}^{+\infty} w(x)\log[w(x)]dx$$

Như vậy, entropy của các đại lượng ngẫu nhiên liên tục là *đại lượng vô cùng*. Nhưng bởi vì trong biểu thức trên thành phần thứ nhất (∞) không phụ thuộc vào x cũng như w(x), nên khi xác định entropy của các đại lượng ngẫu nhiên liên tục, thì thành phần này bị bỏ qua, chỉ kể đến thành phần thứ hai. Entropy được xác định:

$$H(X) = -E\{\log w(x)\} = -\int_{-\infty}^{+\infty} w(x)\log[w(x)]dx \qquad (1.32)$$

được gọi là *entropy vi phân* của đại lượng ngẫu nhiên liên tục. Sau này thuật ngữ "vi phân" thường bỏ qua.

Cũng như đối với tin tức rời rạc, cũng tồn tại các dạng khác nhau của entropy vi phân của các đại lượng ngẫu nhiên liên tục:

1. Entropy có điều kiện của đại lượng ngẫu nhiên Y đối với đại lượng ngẫu nhiên X:

$$H(Y/X) = E\{-\log w(y/x)\} \qquad \text{hoặc là:}$$

$$H(Y/X) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} w_2(x,y)\log w(y/x)dxdy \qquad (1.33)$$

Như vậy, để xác định entropy có điều kiện H(Y/X) cần phải biết không chỉ HMĐXS 2 chiều $w_2(x,y)$, mà còn HMĐXS có điều kiện w(y/x).

2. Entropy đồng thời của hai đại lượng ngẫu nhiên liên tục, bằng:

$$H(X,Y) = E\{-\log w(x,y)\} \qquad \text{hoặc là}$$

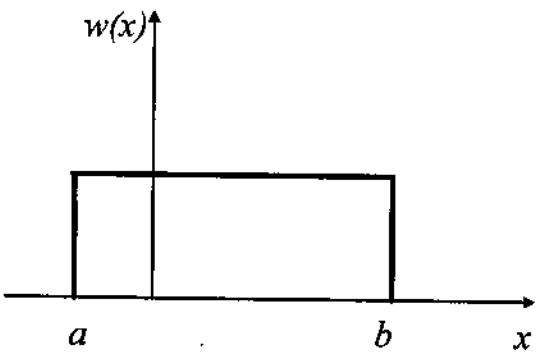
$$H(Y,X) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} w_2(x,y)\log w_2(x,y)dxdy \tag{1.34}$$

Đối với trường hợp X và Y không phụ thuộc lẫn nhau, thì: H(X,Y) = H(X) + H(Y).

Đối với entropy vi phân đồng thời của đại lượng ngẫu nhiên liên tục chính xác phù hợp với (1.17) và (1.18).

3. Lượng thông tin tương hỗ I(X,Y) của hai đại lượng ngẫu nhiên liên tục được xác định theo công thức (1.16). Khi X và Y không phụ thuộc lẫn nhau, thì I(X,Y) = 0.

4. Nếu đại lượng ngẫu nhiên liên tục có HMĐXS 1 chiều phân bố đều trong khoảng (a, b), thì entropy vi phân đạt giá trị cực đại (xem hình 1.8) và bằng:



Hình 1.8. HMĐXS 1 chiều của đại lượng ngẫu nhiên X, có phân bố đều trong khoảng (a, b).

$$H(X)_{\max} = -\int_a^b \left(\frac{1}{b-a}\log\frac{1}{b-a}\right)dx = \log(b-a) \tag{1.35}$$

Như vậy, đại lượng này chỉ phụ thuộc vào hiệu (b - a), không phụ thuộc vào giá trị tuyệt đối của a và b. Do đó $H(X)_{\max}$ không phụ thuộc vào kỳ vọng toán học của đại lượng ngẫu nhiên X.

5. Nếu đại lượng ngẫu nhiên liên tục có HMĐXS 1 chiều phân bố không giới hạn (nghĩa là phân bố trong khoảng từ - ∞ đến +∞), nhưng chỉ giới hạn về công suất, thì entropy vi phân đạt cực đại trong trường hợp phân bố Gauss (phân bố chuẩn).

Chúng ta xác định cực đại này, phù hợp theo công thức (1.32):

$$H(X) = -E\{\log w(x)\}$$

$$w(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-a)^2}{2\sigma^2}}$$

Từ đây có:

$$H(X)_{\max} = -E\{\log[\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-a)^2}{2\sigma^2}}]\} = \log\sqrt{2\pi\sigma^2} + E\{\frac{(x-a)^2}{2\sigma^2}\log e\}$$

Nhưng kỳ vọng $E\{(x - a)^2\} = \sigma^2$, từ đây chúng ta nhận được:

$$H(X)_{\max} = \log\sqrt{2\pi\sigma^2} + \frac{1}{2}\log e$$

Hay, cuối cùng chúng ta có:

$$H(X)_{\max} = \log\sqrt{2\pi\sigma^2.e} \qquad (1.36)$$

Như vậy, entropy vi phân của đại lượng ngẫu nhiên có phân bố chuẩn chỉ phụ thuộc vào công suất σ^2.

Công thức này rất quan trọng, nó sẽ được sử dụng để xác định khả năng thông qua của kênh liên tục.

Cũng như trước đây, chúng ta thấy rằng $H(X)_{\max}$ không phụ thuộc vào kỳ vọng toán học a của đại lượng ngẫu nhiên X. Đặc tính này của entropy rất quan trọng, nó được diễn giải rằng kỳ vọng toán học không phải là đại lượng ngẫu nhiên.

11. ENTROPY VÀ NĂNG SUẤT CỦA NGUỒN TÍN HIỆU LIÊN TỤC ERGODIC

Tín hiệu được sản sinh từ một nguồn tin liên tục, có thể được xem như một quá trình ngẫu nhiên dừng ergodic nào đó, có phổ được giới hạn trong một dải tần. Phù hợp với định lý lấy mẫu, để miêu tả quá trình này có độ dài T_A cần $N = \frac{T_A}{\Delta t} + 1$ mẫu, ở đây $\Delta t = \frac{1}{2F_{\max}}$, gọi là chu kỳ lấy mẫu. Như vậy, tín hiệu với dải phổ tần số hữu hạn, hoàn toàn được đặc trưng bởi giá trị các mẫu của mình. Bởi vậy, các giá trị trung gian giữa các mẫu không làm tăng hiểu biết của chúng ta, hay nói một cách khác là chúng không có thông tin. Khi xác định entropy của tín hiệu liên tục chỉ cần tính với giá trị các mẫu của nó.

Chúng ta biết rằng entropy có đặc tính cộng. Nếu có một tín hiệu rời rạc, mà các phần tử có phân bố đồng xác suất và entropy của mỗi phần tử là H(X), thì entropy của tín hiệu có N phần tử sẽ là N.H(X). Bằng cách tương tự, ta có thể tính entropy của tín hiệu liên tục, có khoảng tồn tại là T_A sẽ là:

$$N.H_1(X) = (2F_{\max}.T_A + 1).\ H_1(X)$$

Ở đây $H_1(X)$- entropy một mặt cắt của tín hiệu ngẫu nhiên, được xác định theo lý luận trình bày ở đầu mục 10, qua hàm mật độ xác suất một chiều. Thứ nguyên của entropy $H_1(X)$ - bit trên một mẫu của tín hiệu ngẫu nhiên (một tiết diện của quá trình ngẫu nhiên).

Năng suất của quá trình ngẫu nhiên liên tục sẽ bằng:

$$R(X) = \frac{(2F_{\max}.T_A + 1).H_1(X)}{T_A}$$

Hoặc gần đúng, bỏ qua entropy của một mẫu, chúng ta có:

$$R(X) = 2F_{\max}H_1(X) \quad \text{[bit/sec]} \tag{1.37}$$

Như vậy, năng suất của nguồn tín hiệu ngẫu nhiên liên tục ergodic hoàn được xác định bằng entropy của một mẫu và hai lần bề rộng phổ tần số của tín hiệu.

12. KHẢ NĂNG THÔNG QUA CỦA KÊNH LIÊN TỤC

Nếu X(t)- tín hiệu ở đầu vào của kênh truyền, còn Y(t) = X(t) + n(t) - là tín hiệu đầu ra của nó, n(t)- nhiễu cộng, (chúng được giả thiết là các quá trình ngẫu nhiên dừng ergodic) thì tốc độ truyền tin qua kênh liên tục được xác định theo công thức (1.23), trong đó đại lượng $\frac{1}{\bar{\tau}}$ được thay thế bằng $2F_{max}$ (hay 2B, với B - dải thông của kênh truyền):

$$R(X,Y) = 2B[H(Y) - H(Y/X)]$$

Ở đây, cũng như trước kia, H(Y)- đó là entropy của tin tức đầu ra, còn H(Y/X)- đó là entropy của nhiễu (tại sao lại gọi như vậy, sẽ thấy rõ từ các trình bày dưới).

Khả năng thông qua bằng cực đại của khả năng tốc độ truyền theo kênh liên lạc, khi đó nguồn tín hiệu hoàn toàn phù hợp với đặc trưng của kênh truyền:

$$C = 2B \times \max[H(Y) - H(Y/X)] \tag{1.38}$$

Cực đại của H(Y) sẽ đạt được trong trường hợp quá trình ngẫu nhiên Y có phân bố Gauss. Khi đó:

$$\max H(Y) = \log\sqrt{2\pi.e.\sigma_y^2} \tag{1.39}$$

Khi tính toán ảnh hưởng của nhiễu, cần thiết phải khảo sát trường hợp xấu nhất, khi đó tất nhiên, nhiễu cũng có phân bố theo luật Gauss.

Hàm mật độ phân bố xác suất có điều kiện w(y/x)- đó là hàm mật độ phân bố xác suất của quá trình Y, khi đã biết quá trình X, mặc dù quá trình X cũng là quá trình ngẫu nhiên. Nhưng bởi vì Y(t) = X(t) + n(t), nên có thể viết:

$$w(y/x) = \frac{1}{\sqrt{2\pi.\sigma_{y/x}}} e^{-\frac{(x-n)^2}{2\sigma_{y/x}^2}}$$

Ở đây: $\sigma_{y/x}^2$ - phương sai của quá trình Y khi đã biết X, nghĩa là phương sai của nhiễu σ_n^2.

Chúng ta xác định entropy có điều kiện H(Y/X):

$$H(Y/X) = -E\{\log w(y/x)\} = -E\{\log[\frac{1}{\sqrt{2\pi\sigma_n}} e^{-\frac{(x-n)^2}{2\sigma_n^2}}]\}$$

Trong biểu thức này chúng ta giả thiết rằng X đã biết từ trước. Như vậy, quá trình X trong biểu thức này đóng vai trò là kỳ vọng của quá trình Y. Tuy nhiên, cũng đã biết rằng entropy của đại lượng ngẫu nhiên liên tục không phụ thuộc vào kỳ vọng toán học.

Khi đó chúng ta nhận được:

$$H(Y/X) = -E\{\log[\frac{1}{\sqrt{2\pi\sigma_n}} e^{-\frac{n^2}{2\sigma_n^2}}]\} = -E\{\log w(n)\} = H(n)$$

Từ đây, chúng ta thấy rằng, tại sao entropy có điều kiện H(Y/X) lại được gọi là entropy của nhiễu.

Đối với luật phân bố Gauss của nhiễu, thì entropy của nhiễu đạt giá trị cực đại, và phù hợp với công thức (1.36), sẽ bằng:

$$\max H(Y/X) = H(n)_{\max} = \log\sqrt{2\pi.e.\sigma_n} \tag{1.40}$$

Đặt biểu thức (1.39), và (1.40) vào (1.38), chúng ta nhận được:

$$C = 2B[\log\sqrt{2\pi.e.\sigma_y^2} - \log\sqrt{2\pi.e.\sigma_n^2}] = 2B.\log\frac{\sigma_y}{\sigma_n}$$

Đem chuyển số 2 vào trong dấu log, chúng ta nhận được: $C = B\log\frac{\sigma_y^2}{\sigma_n^2}$. Trong đó biểu thức $\sigma_n^2 = P_n$ - công suất của nhiễu, còn $\sigma_y^2 = \sigma_x^2 + \sigma_n^2 = P_c + P_n$, ở đây P_c- công suất tín hiệu ở đầu vào kênh truyền. Với sự tính toán này, thì biểu thức nhận được cuối cùng cho công thức tính toán khả năng thông qua của kênh truyền tương tự (theo công thức Shannon):

$$C = B\log\left(1 + \frac{P_c}{P_n}\right) \tag{1.41}$$

Như vậy, khả năng thông qua của kênh truyền C [bit/sec] phụ thuộc vào:

- *Dải thông của kênh truyền B [Hz]*. Đến lượt mình, dải thông B chủ yếu phụ thuộc vào *bản chất vật lý* của kênh truyền (sự phụ thuộc của giá trị các tham số đặc tính kênh truyền vào tần số). Trong một chừng mực nào đó B còn phụ vào *chiều dài kênh truyền* (kênh càng dài B càng giảm).

Chúng ta có bất đẳng thức sau:

B (đường dây đôi) < B (cáp đồng trục) < B (cáp quang).

Chúng ta cũng có thể hỏi *đâu là giới hạn* dải thông của kênh truyền là *khoảng không gian* ?

Trong các hệ thống truyền tin rất thường xuyên phải đối mặt với vấn đề *thiếu dải thông của kênh truyền*. Trong các trường hợp như vậy, tại điểm thu các ký hiệu bị méo dạng mạnh và phát sinh giao thoa giữa các ký hiệu (ISI) và gây ra lỗi. Để giảm hiệu ứng thiếu B, người ta thường sử dụng *điều chế với tín hiệu băng cơ sở nhiều mức biên độ (có bộ DAC trước điều chế) hoặc điều chế OFDM* (xem chương 6).

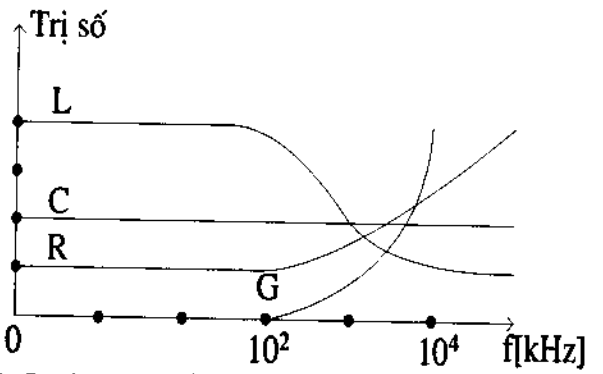
Sự phụ thuộc của C được xem xét dưới đây.

- *Tỉ số (công suất của tín hiệu P_c)/(công suất của tạp âm P_n).* Khi tăng công suất của tín hiệu, hoặc giảm công suất của tạp âm, C đều tăng, nhưng sự tăng này nhanh chóng bão hòa.

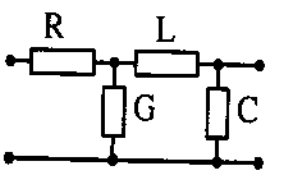
Để kết thúc, chúng ta có các nhận xét sau:

- Để tốc độ truyền thông tin theo kênh tương tự tiến gần đến khả năng thông qua của kênh, thì tín hiệu X(t) cần phải có cấu trúc thống kê tiến gần đến tạp âm trắng với luật phân bố Gauss.

♣ *Để diễn giải khái niệm dải thông* của một dạng kênh truyền cụ thể là đường dây đôi, chúng ta dẫn ra đây các kết quả nghiên cứu về sự phụ thuộc của các tham số đặc trưng của đường dây đôi vào tần số [10], chúng được biểu diễn trên hình 1.9.



a). Sự phụ thuộc của các tham số đơn vị vào tần số.



b). Sơ đồ tương đương của đoạn dây đôi.

Hình 1.9. Diễn giải khái niệm dải thông của kênh truyền là đường dây đôi.

Đường dây đôi là hai lõi đồng xoắn lại cùng vỏ cách điện. Một đoạn dây đôi có sơ đồ tương đương là một mạng bốn cực, với các tham số:

- Điện trở đơn vị R (Ω/km) đặc trưng cho suy giảm tín hiệu dưới dạng nhiệt;
- Độ tự cảm đơn vị L (mH/km) đặc trưng cho sự ngăn chặn truyền lan của các thành phần tần số cao của tín hiệu;
- Điện dung đơn vị C (μF/km) đặc trưng cho suy giảm các thành phần tần số cao của tín hiệu, do bị phân dòng xuống dây đất;
- Điện dẫn đơn vị G (1/μΩ.km) đặc trưng cho suy giảm tín hiệu do *dòng rò* giữa hai dây.

Điện trở đơn vị R do *hiệu ứng bề mặt* nên trị số tăng dần khi tần số đủ cao. Chẳng hạn, trong trường cụ thể này (đường kính lõi đồng d = 4mm, vỏ là chất cách điện polyethylene) bắt đầu vào khoảng tần số $f \geq 100$ kHz, điện trở R bắt đầu tăng tỉ lệ với $\sqrt{f}$, *f*- tần số công tác, còn trước đó hầu như không đổi. Tương tự như vậy cho độ tự cảm L.

Điện dẫn đơn vị G có giá trị nhỏ, chỉ bắt đầu khác không và tăng dần theo tần số ứng với khoảng tần số $f \geq 100$ kHz.

Điện dung đơn vị C không phụ thuộc vào tần số, vì nó không chịu ảnh hưởng của *hiệu ứng bề mặt*. C chỉ phụ thuộc vào khoảng các hai dây và chất cách điện giữa chúng.

Có hai tham số điện từ đặc trưng cho sự truyền lan sóng dọc theo dây dẫn, đó là:

- Trở kháng sóng, ký hiệu Z_0, và được xác định:

$$Z_0 = \frac{u}{i} = \sqrt{\frac{R + j\omega C}{G + j\omega L}}$$

- Hằng số truyền lan, ký hiệu γ, nói chung là đại lượng phức và được xác định:

$$\gamma = \alpha + i\beta = \sqrt{[(R + i\omega L)(G + i\omega C)]}$$

trong đó: α- hằng số suy giản, đặc trưng cho sự giảm sóng điện từ khi truyền dọc theo dây dẫn, đơn vị đo trong dạng $20\log_{10}(\alpha)$ là [dB/km]; γ- hằng số sóng, đặc trưng cho độ di pha của sóng điện từ khi truyền dọc theo dây dẫn, đơn vị đo [rad/km].

Chúng ta nhận thấy rằng các đại lượng này đều phụ thuộc vào tần số. Tuy nhiên, từ hình 1.9, chúng ta thấy rằng sự phụ thuộc vào tần số của các đại lượng này là yếu và gần như là như nhau đối với mọi tần số trong khoảng f = (0 ÷100) kHz. Còn khi tần số công tác f > 100 kHz, thì sự phụ thuộc vào tần số của các đại lượng này là khác nhau với các tần số khác nhau. Hơn nữa, khi tần số công tác càng cao, sự phụ thuộc càng mạnh. Từ đây, chúng ta có thể nhận thấy nếu chọn tốc độ truyền tin R ≤ 100 Kb/sec (đòi hỏi dải thông B ≤ 100 kHz) thì tín hiệu chịu tác động yếu và như nhau với các thành phần tần phổ khác nhau. Ngược lại, khi chọn tốc độ truyền tin R > 100 Kb/sec (đòi hỏi dải thông B > 100 kHz), thì các thành phần tần phổ cao của tín hiệu chịu ảnh hưởng mạnh và khác nhau với các tần số khác nhau. Hơn nữa, các thành phần tần số càng cao càng chịu ảnh hưởng mạnh. Điều này liên quan đến độ méo tín hiệu. Tuy nhiên, nếu hỏi đâu là giới hạn của tốc độ truyền tin R, cũng như đâu là giới hạn của dải thông B của đường dây đôi, thì câu trả lời còn phụ thuộc vào các công nghệ xử lý tín hiệu gì được sử dụng cho hệ thống truyền tin đang xét đó.

Chúng ta đưa ra thảo luận vấn đề này để thấy rằng khái niện dải thông của đường dây đôi nói riêng, của các dạng kênh truyền khác nói chung là đại lượng không dễ gì xác định chính xác. Giá trị dải thông B = 100 kHz ở trong mục này chỉ là con số tương đối.

13. ENTROPY- EPXILON CỦA NGUỒN TIN TỨC LIÊN TỤC

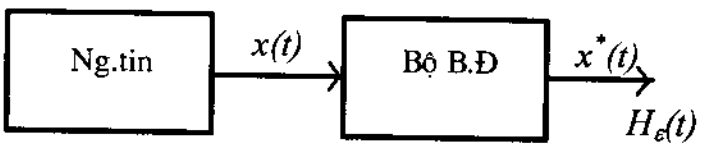
Tín hiệu ở đầu ra của nguồn tin tức liên tục (hình 1.10). Chẳng hạn, ở đầu ra của micrô, máy điện thoại v.v.. là một quá trình ngẫu nhiên liên tục dừng; entropy của nó ở mặt cắt bất kỳ, trong trường hợp chung, bằng vô cùng, như đã chỉ ra trong mục 10. Một lượng thông tin như thế không có thể truyền theo một kênh truyền tin thực, vì khả năng thông qua của nó luôn luôn bị giới hạn. Hơn nữa, đó là điều không cần thiết, bởi vì tốc độ tiếp nhận thông tin của người sử dụng bất kỳ ở đầu ra kênh truyền luôn luôn bị giới hạn do khả năng vật lý của mình. Do tín hiệu liên tục ở đầu

ra của kênh truyền, ngay cả khi không có nhiễu, cũng khác với tín hiệu ở đầu vào, bởi vì nó chứa đựng không phải những thông tin về mình (thông thường trước kênh truyền có thể thực hiện những phép biến đổi từ một tập hợp tín hiệu này thành những tập hợp tín hiệu khác, như: điều chế, khuếch đại, biến tần ...). Ngay cả khi sự biến đổi tin tức liên tục thành tín hiệu bởi các thiết bị phù hợp (micro, máy điện thoại ...) với tiêu hao một phần thông tin, còn tín hiệu biểu diễn tin tức chỉ với một độ chính xác nào đó:

$$\varepsilon(t) = X(t) - X^*(t)$$

Ở đây X(t) - tín hiệu ở đầu vào bộ biến đổi,

$X^*(t)$ – tín hiệu ở đầu ra bộ biến đổi hay là đánh giá của tín hiệu đầu vào bởi bộ biến đổi. Hơn nữa, bộ biến đổi luôn luôn là một thiết bị quyết định nào đó, được làm việc theo qui tắc xác định với các tiêu chuẩn chất lượng đã cho.



Hình1.10. Nguồn tin tức liên tục và thiết bị biến đổi.

Tiêu chuẩn chất lượng, như đã biết, được xác định theo thông tin bởi người sử dụng, chẳng hạn, thông tin độ lệch trung bình bình phương:

$$\varepsilon_{TB} = E\{[X(t) - X^*(t)]^2\} \tag{1.42}$$

hoặc là phương sai của sai lệch:

$$\sigma_\varepsilon^2 = E\{\varepsilon^2(t)\}$$

Entropy-epxilon $H_\varepsilon(X)$ (entropy-ε) là lượng thông tin trung bình cực tiểu, mà cần chứa ở trong tín hiệu đầu ra $X^*(t)$ về tín hiệu đầu vào X(t), để tín hiệu này có thể khôi phục với độ chính xác ε_{TB} đã cho:

$$H_\varepsilon(X) = \min I(X, X^*) = H(X) - \max H(X / X^*) \tag{1.43}$$

Ở đây:

$I(X, X^*)$ – lượng thông tin trung bình tương hỗ giữa X(t) và $X^*(t)$,

H(X) và $H(X, X^*)$ – lượng thông tin vi phân của tín hiệu X(t) và lượng thông tin vi phân có điều kiện của X(t), khi đã biết $X^*(t)$ tương ứng, min và max sẽ chạy theo tất cả các khả năng của hàm phân bố mật độ có điều kiện $w(x / x^*)$.

Trong trường hợp chung, khi tín hiệu (hoặc tin tức) X(t) là quá trình ngẫu nhiên Gauss, với phương sai σ_x^2, thì độ lệch ε(t) cũng là QTNN Gauss với phương sai σ_ε^2, còn với sự tính toán

đặc trưng cộng tính của các sai lệch ε(t) và entropy vi phân có điều kiện $H(X, X^*)$ – sẽ hoàn toàn được xác định bởi entropy vi phân H(ε). Và có:

$$\max H(X, X^*) = \max H(\varepsilon) = \log(\sigma_\varepsilon \sqrt{2\pi.e}).$$

Khi đó entropy- ε của một tiết diện nguồn Gauss (entropy- ε của một mẫu):

$$H_\varepsilon(X) = \log(\sigma_x \sqrt{2\pi e}) - \log(\sigma_\varepsilon \sqrt{2\pi e}) = 0{,}5\log(\frac{\sigma_x^2}{\sigma_\varepsilon^2}) \quad (1.44)$$

Đại lượng $\frac{\sigma_x^2}{\sigma_\varepsilon^2}$ cho tỉ số của công suất (phương sai) của tín hiệu X(t), với công suất (phương sai) của sai lệch, khi đó độ lệch trung bình bình phương của tín hiệu X(t) và $X^*(t)$ không vượt quá σ_ε.

Như vậy, *năng suất của nguồn tin tức liên tục* có thể được xác định như là lượng thông tin cần phải truyền qua kênh trong một đơn vị thời gian, để tin tức khôi phục với độ chính xác đã cho:

$$R_\varepsilon(X) = \nu.H_\varepsilon(X) \quad (1.45)$$

Ở đây ν = 1/Δt - tốc độ truyền một mẫu ở đầu ra của nguồn, Δt - khoảng thời gian giữa các mẫu.

Đối với tín hiệu dừng với bề rộng phổ hạn chế $\Delta t = 1/(2F_{max})$, thì:

$$R_\varepsilon(X) = 2.F_{max}.H_\varepsilon(X).$$

Ngoài ra, nếu nguồn có phân bố Gauss, thì:

$$R_\varepsilon(X) = F_{max}\log(\frac{\sigma_x^2}{\sigma_\varepsilon^2}) \quad (1.46)$$

Lượng thông tin được sản ra của nguồn Gauss sau khoảng thời gian T_c, bằng:

$$T_c.R_\varepsilon(X) = T_c.F_{max}\log(\frac{\sigma_x^2}{\sigma_\varepsilon^2}) \quad (1.47)$$

nó trùng với công thức đối với dung lượng của tín hiệu, khi dải động của tín hiệu là:

$$D_c = \log(\frac{\sigma_x^2}{\sigma_\varepsilon^2})$$

Điều này có nghĩa là dung lượng tín hiệu ở đầu ra của nguồn bằng lượng thông tin (1.46), mà nó được chứa đựng trong tín hiệu để việc khôi phục nó với một độ chính xác đã cho.

Đối với kênh với khả năng thông qua C, ở đầu vào mắc nguồn với năng suất $R_\varepsilon(X)$, định lý Shannon đối với kênh có nhiễu được phát biểu:

Nếu khi cho trước tiêu chuẩn độ chính xác của nguồn (chẳng hạn, σ_ε), còn năng suất-epxilon của nó nhỏ hơn khả năng thông qua của kênh $R_\varepsilon(X) < C$, thì tồn tại khả năng mã hoá (biến

đổi tín hiệu), với độ không chính xác khôi phục tín hiệu tiến gần đến σ_ε tuỳ ý; còn khi $R_\varepsilon(X) > C$ thì không có khả năng như thế.

Định lý Shannon xác định các khả năng giới hạn của sự hoà hợp nguồn tin tức liên tục với kênh truyền. Trong thực tiễn điều này đạt được bằng việc ứng dụng của các dạng chống nhiễu của điều chế và mã hoá.

14. MỐI QUAN HỆ GIỮA CÔNG SUẤT CỦA TÍN HIỆU VÀ BỀ RỘNG PHỔ CỦA NÓ

Từ công thức (1.41) có thể tạo ra một sự hiểu lầm, rằng khả năng thông qua C của kênh liên tục có thể tăng không giới hạn khi tăng B. Điều đó là không thể, bởi vì đồng thời với việc tăng B sẽ dẫn đến tăng công suất của tạp nhiễu, tức là tăng đại lượng P_n đứng trong mẫu số dưới dấu log trong biểu thức (1.41).

Công suất của nhiễu:

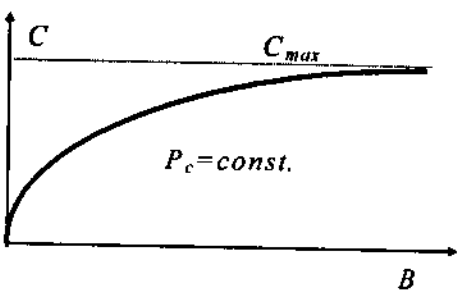
$$P_n = N_0.B \tag{1.48}$$

Ở đây N_0 - mật độ phổ công suất của nhiễu,

Đặt biểu thức này vào công thức (1.41), chúng ta nhận được:

$$C = B.\log\left(1 + \frac{p_c}{N_0 B}\right) \tag{1.49}$$

Phân tích biểu thức (1.49), chúng ta thấy rằng khi tăng dải thông B của kênh truyền, thì ban đầu làm tăng khả năng thông qua C của kênh, sau đó giá trị này tăng tiến tới giới hạn C_{max}, khi tăng B đến vô cùng (hình 1.11).



Hình 1.11. Giới hạn C_{max} khi tăng dải thông B của kênh.

Hình 1.12. Mối quan hệ giữa P_c và B.

Chúng ta nhận được một kết quả khác khi khảo sự phụ thuộc của công suất tín hiệu P_c vào bề rộng phổ của nó B - khi cho trước khả năng thông qua C của kênh và mật độ phổ nhiễu N_0. Biến đổi công thức (1.43), chúng ta nhận được:

$$P_c = N_0 B\left(2^{\frac{C}{B}} - 1\right)$$

Sự phụ thuộc này được biểu diễn trên hình 1.12. Từ đây thấy rằng công suất của tín hiệu đơn điệu giảm khi tăng bề rộng phổ của nó. Điều này cho phép truyền tin với một tốc độ cần thiết đi một khoảng cách rất xa với công suất tín hiệu rất nhỏ, nếu sử dụng tín hiệu với bề rộng phổ đủ rộng.

Khả năng giảm công suất của tín hiệu do việc tăng bề rộng phổ của nó được gọi là mối tương quan giữa công suất và bề rộng phổ của tín hiệu. Khả năng này được áp dụng khi gia công các hệ truyền tin vũ trụ, khi mà công suất của tín hiệu luôn luôn bị hạn chế do nguồn cung cấp trên các thiết bị vũ trụ.

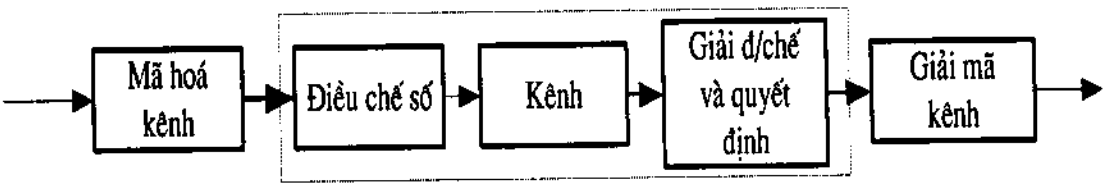
15. SO SÁNH KHẢ NĂNG THÔNG QUA CỦA CÁC KÊNH TƯƠNG TỰ VÀ RỜI RẠC

a). Mô hình kênh rời rạc

Khi xem xét các vấn đề của kênh truyền tin, chúng ta thường phải phân biệt các khái niệm: kênh rời rạc, kênh liên tục và kênh bán liên tục. Rõ ràng trong các định nghĩa khác nhau như vậy, các đặc trưng của kênh phải khác nhau.

Kênh rời rạc (RR) - (hình 1.13) được hiểu là dạng kênh mở rộng, bao gồm đầu vào kể từ đầu vào bộ điều chế số, còn đầu ra kể từ đầu ra của bộ giải điều chế và quyết định. Nói một cách khác, kênh rời rạc có đầu vào, đầu ra đều là tín hiệu rời rạc.

- Nếu bộ điều chế số thực hiện điều chế tín hiệu nhị phân (M = 2), ta có kênh rời rạc đơn giản nhất: kênh nhị phân đối xứng (Binary Symetric Channel) (xem mục 9).



Hình 1.13. Mô hình của kênh rời rạc (RR).

- Nếu bộ điều chế số thực hiện điều chế tín hiệu M mức (M ≥ 2), ta có kênh rời rạc với tín hiệu M mức (xem mục 17).
- Kênh liên tục (LT) là kênh truyền mà đầu vào, đầu ra đều là tín hiệu liên tục, tức là kênh truyền mà đầu vào ở sau bộ điều chế, còn đầu ra ở phía trước bộ giải điều chế (xem mục 17).
- Kênh bán liên tục (BLT) được hiểu là kênh truyền mà đầu vào là tín hiệu rời rạc, còn đầu ra là tín hiệu liên tục. Trong hệ thống với kênh bán liên tục việc gia công tối ưu các từ mã được thực hiện trong kênh liên tục (thu toàn cục) (xem mục 17).

b). *So sánh khả năng thông qua của các kênh tương tự và rời rạc*

Khả năng thông qua của kênh nhị phân được xác định theo công thức (1.30). Tốc độ truyền ký hiệu R của kênh nhị phân có giá trị R = $\frac{1}{\tau}$ = 2B. Đưa giá trị này vào công thức (1.30), chúng ta có:

$$\frac{C}{B} = 2.[1 + p.\log p + (1-p).\log(1-p)] \qquad (1.50)$$

Để xác định, chúng ta coi rằng trong kênh truyền rời rạc, việc truyền tin sẽ sử dụng phương pháp điều chế tần số rời rạc, khi đó xác suất lỗi được xác định theo công thức:

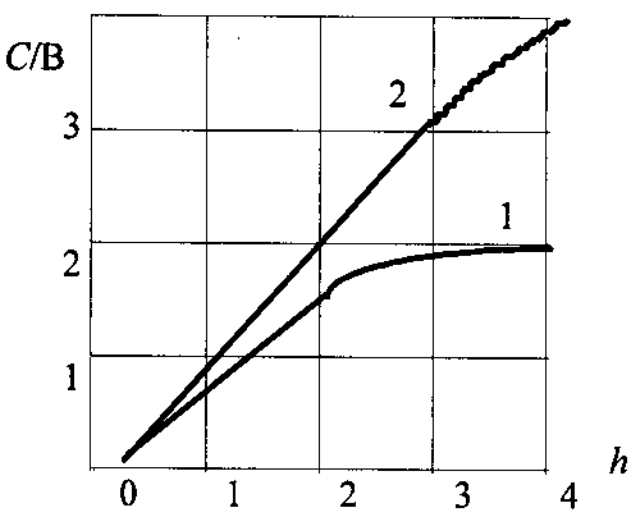
$$p = 0,5.[1 - \Phi(h)]$$

Đặt biểu thức này vào công thức (1.50), chúng ta nhận được sự phụ thuộc khả năng thông qua của kênh vào đại lượng h (hình 1.14, đường cong 1). Từ hình vẽ này, chúng ta thấy rằng khi h→ ∞, khả năng thông qua của kênh tiến tới 2B.

Khả năng thông qua của kênh liên tục được xác định theo công thức Shannon (1.41), khi đó chúng ta nhận được:

$$\frac{C}{B} = \log_2[1 + h^2] \qquad (1.51)$$

Sự phụ thuộc này cũng được trình bày trên hình 1.14, đường cong 2.



Hình 1.14. Sự phụ thuộc khả năng thông qua của kênh vào tham số h.

So sánh hai đồ thị, chúng ta thấy rằng khi h→ ∞, đối với kênh rời rạc $\frac{C}{B} \to 2$ còn đối với kênh liên tục $\frac{C}{B} \to \infty$, hơn nữa với giá trị bất kỳ của tham số h, khả năng thông qua của kênh liên tục đều vượt khả năng thông qua của kênh rời rạc. Điều này được giải thích như sau các tín

hiệu điều chế rời rạc khác với nhiễu trắng Gauss, còn phù hợp theo lý thuyết Shannon, khả năng thông qua của kênh sẽ lớn nhất ở các hệ thống mà tín hiệu tiến tới nhiễu trắng Gauss. Các tín hiệu tương tự theo cấu trúc của mình có thể tiến đến nhiễu trắng Gauss tốt hơn các tín hiệu rời rạc.

16. HIỆU QUẢ CỦA HỆ THỐNG TRUYỀN TIN

Thuật ngữ "hiệu quả" có nghĩa là "kinh tế ". Mức độ hiệu quả có thể được đánh giá theo đơn vị tiền tệ (chẳng hạn, giá thành xây dựng một km của đường liên lạc, thời hạn hoàn vốn, chi phí khai thác v.v..). Mặc dù đó là các chỉ tiêu quan trọng, nhưng chúng ta sẽ xác định mức độ hiệu quả của các hệ thống truyền tin bằng cách sử dụng các đặc trưng kỹ thuật của chúng. Chẳng hạn, một hệ thống truyền tin nào đó bảo đảm truyền tin trong dải âm tần với tốc độ 1200 bit/sec, còn một hệ thống truyền tin thứ hai bảo đảm tốc độ 9600 bit/sec, thì với quan điểm của chúng ta hệ thống truyền tin thứ hai được coi là hiệu quả hơn (với quan điểm sử dụng tốt hơn dải thông của kênh truyền). Mặc dù, hệ thống thứ hai có thể có giá thành đắt hơn hệ thống thứ nhất. Lại cũng có thể, với quan điểm của chỉ tiêu kỹ thuật thì hệ thống thứ hai có thể coi rằng kém hiệu quả hơn hệ thống thứ nhất, nếu việc so sánh diễn ra không theo khả năng thông qua của kênh, mà chẳng hạn, theo công suất kích thích vào đường truyền. Trong trường hợp thứ hai này, có thể hệ thống thứ hai đòi hỏi công suất tín hiệu lớn hơn.

Các hệ thống truyền tin hiện đại cần bảo đảm đạt được tốc độ truyền tin cao với dải tần số đã cho, cực tiểu công suất tín hiệu và xác suất méo cực tiểu.

Để đánh giá mức độ hiệu quả của các hệ thống truyền tin rất thường xuyên sử dụng ba chỉ tiêu hiệu quả sau:

- Chỉ tiêu hiệu quả: $\beta = \dfrac{R}{P_c / N_0}$; $\gamma = \dfrac{R}{B}$; $\eta = \dfrac{R}{C}$ (1.52)

Trong các công thức này: R - tốc độ truyền tin; P_c/N_0 - tỉ số công suất của tín hiệu trên mật độ phổ công suất nhiễu; B - dải thông tần của kênh truyền; C - khả năng thông qua của kênh truyền.

Tất cả các chỉ tiêu hiệu quả trên đều là các đại lượng không thứ nguyên, và được xác định với giả thiết rằng kênh truyền bảo đảm xác suất méo tín hiệu đủ nhỏ (được cho trước).

Các chỉ tiêu hiệu quả nêu trên có ý nghĩa vật lý đơn giản như sau: chỉ tiêu hiệu quả β - chỉ ra công suất của tín hiệu P_c được sử dụng như thế nào trong truyền tin khi cho trước tốc độ truyền R; chỉ tiêu hiệu quả γ - chỉ ra hiệu quả sử dụng dải thông kênh truyền như thế nào; chỉ tiêu hiệu quả η - chỉ ra hiệu quả sử dụng khả năng thông qua C của kênh truyền như thế nào.

Tất cả các chỉ tiêu hiệu quả trên đều có mối quan hệ tương hỗ với nhau. Để xác định mối quan hệ tương hỗ này, từ công thức (1.52) chúng ta loại bỏ R và nhận được:

$$\gamma = \frac{P_c.\beta}{N_0.B} = \frac{P_c}{P_n}.\beta \tag{1.53}$$

ở đây $\frac{P_c}{P_n} = h^2$ - tỉ số của công suất tín hiệu trên công suất nhiễu.

Khả năng thông qua của kênh liên tục (kênh Gauss) phụ thuộc vào đại lượng h^2 được xác định theo công thức Shannon (1.41). Đặt vào công thức này quan hệ (1.52) và (1.53), chúng ta nhận được:

$$\eta = \frac{\gamma.B}{B.\log_2\left(\frac{P_c}{P_n}+1\right)} = \frac{\gamma}{\log_2\left(\frac{\gamma}{\beta}+1\right)} \tag{1.54}$$

Chúng ta nghiên cứu giới hạn của sự phụ thuộc giữa β và γ trong công thức (1.54), bằng cách đặt η =1 (về mặt lý thuyết đại lượng có thể đạt được giới hạn phù hợp với lý thuyết Shannon đối với kênh, trong đó tín hiệu và nhiễu theo cấu trúc của mình tiến tới tạp âm trắng, đại lượng R có thể tiến gần đến C tuỳ ý, khi thực hiện mã hoá phù hợp). Sau biến đổi đơn giản, chúng ta nhận được:

$$\beta = \frac{\gamma}{2^{\gamma}-1} \tag{1.55}$$

Giới hạn nhận được của sự phụ thuộc giữa β và γ đối với kênh liên tục được trình bày trên hình 1. 15 (đại lượng β và γ được trình bày theo tỉ lệ logarit). Ở đây trình bày mối quan hệ giữa β và γ đối với các dạng điều chế rời rạc khác nhau. Khi tính toán các đại lượng này đã cho rằng xác suất méo tín hiệu rất nhỏ (xác suất méo tín hiệu chọn bằng $p = 10^{-5}$). Trong trường hợp này, nếu sử dụng công thức (1.29), chúng ta nhận được R ≈ C = $\frac{1}{T}$, ở đây T- độ dài của phần tử tín hiệu.

Dải thông tần hiệu quả của kênh truyền đối với các dạng điều chế khác nhau được xác định theo công thức:

$B = \frac{1}{T}$ - đối với tín hiệu ASK hai mức (M=2) biên độ.

$B = \frac{M}{T.\log_2 M}$ - đối với tín hiệu FSK với M mức tần số.

$B = \frac{1}{T.\log_2 M}$ - đối với tín hiệu PSK với M mức pha. (1.56)

Bảng 1. Hiệu quả của các dạng điều chế khác nhau

Chỉ tiêu	ASK	FSK	BPSK
Tốc độ truyền thông tin R	$\frac{1}{T}$	$\frac{1}{T}$	$\frac{1}{T}$
Dải thông tần hiệu quả B	$\frac{1}{T}$	$\frac{2}{T}$	$\frac{1}{T}$
$\gamma = \frac{R}{B}$	1 (0 db)	0,5 (-3 db)	1 (0 db)
h_0^2	39	19,5	9,75
$\beta = \frac{1}{h_0^2}$	$\frac{1}{39}$ (-15,9 db)	$\frac{1}{19,5}$ (-13 db)	$\frac{1}{9,75}$ (-9,9 db)
$\eta = \frac{\gamma}{\log_2\left(\gamma \cdot h_0^2 + 1\right)}$	0,19	0,15	0,29

Chúng ta xác định đại lượng β đối với trường hợp khảo sát (xác suất méo tín hiệu bé):

$$\beta = \frac{R}{P_c / N_0} = \frac{1/T}{P_c / N_0} = \frac{N_0}{E} = \frac{1}{h_0^2} \tag{1.57}$$

ở đây $E = P_c.T$ - năng lượng của tín hiệu.

Chúng ta tìm đại lượng h_0^2 đối với xác suất đã cho $p = 10^{-5}$, khi thu tối ưu các tín hiệu điều chế trên, theo công thức:

$$p_{ASK} = \frac{1}{2}\left[1 - \Phi\left(\frac{h_0}{\sqrt{2}}\right)\right] = 10^{-5}$$

$$p_{FSK} = \frac{1}{2}\left[1 - \Phi(h_0)\right] = 10^{-5}$$

$$p_{PSK} = \frac{1}{2}\left[1 - \Phi\left(\sqrt{2}h_0\right)\right] = 10^{-5} \tag{1.58}$$

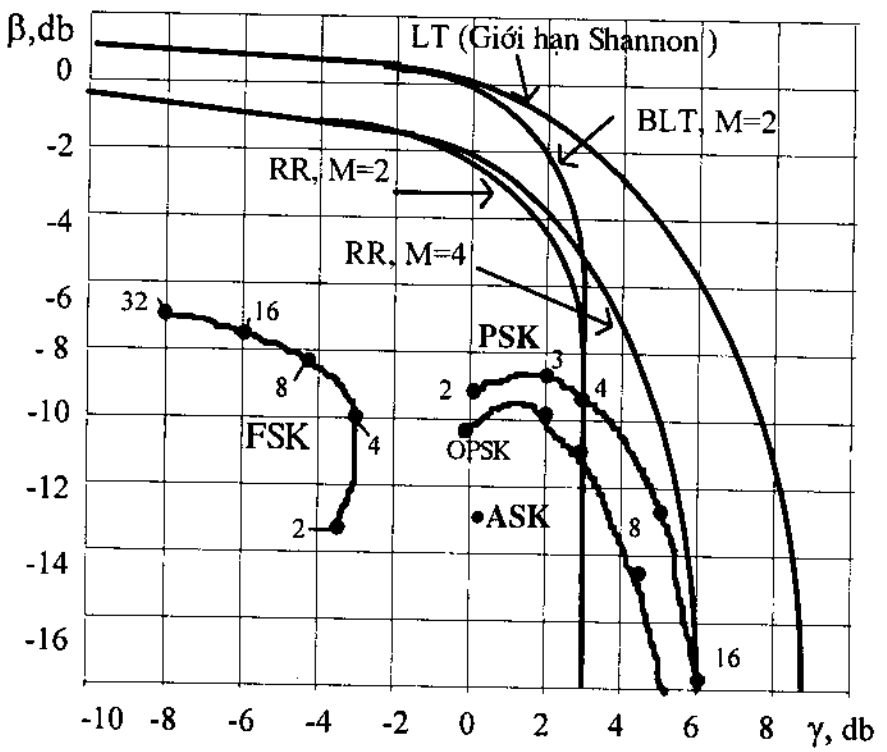
từ đây xác định được: $h_{0AM}^2 = 39$, $h_{0FSK}^2 = 19,5$, $h_{0PSK}^2 = 9,75$. Tất cả các tính toán trên đối với trường hợp M = 2, được trình bày trong bảng 1. Ở đây chỉ số hiệu quả η được xác định theo công thức (1.54).

Các giá trị của chỉ tiêu hiệu quả η (dòng cuối cùng của bảng) đã chỉ ra chúng hiệu quả như thế nào đối với các dạng điều chế số khác nhau.

Các công thức tính toán có thể dùng cho cả trường hợp tín hiệu nhiều mức (M mức). Kết quả tính toán được trình bày trong dạng đồ thị trên hình 1.15. Các số trên hình vẽ chỉ ra các giá trị M được sử dụng cho các tín hiệu điều chế.

Đối với chỉ tiêu hiệu quả β khi ASK cần giải thích thêm. Vấn đề ở chỗ là tín hiệu ASK khác với tín hiệu FSK, cũng như tín hiệu PSK, tồn tại các khoảng trống "thụ động" (khi điều chế ký hiệu 0, giá trị tức thời của công suất tín hiệu bằng 0). Điều đó giải thích rằng khi tính công suất của tín hiệu ASK theo trung bình, chứ không phải mức công suất đỉnh, đại lượng β tăng 3db. Điều này được thấy trên hình 1.15.

Từ hình vẽ thấy rằng khi tăng M đối với FSK đại lượng β - tăng, còn đại lượng γ - giảm, nghĩa là đối với tín hiệu FSK sử dụng công suất tốt, nhưng sử dụng dải thông kém. Đối với tín hiệu FSK nhiều mức tần số nên sử dụng trong trường hợp công suất tín hiệu bị hạn chế, còn dải thông không bị hạn chế, chẳng hạn, trong thông tin vũ trụ. Ngược lại, đối với tín hiệu PSK, khi tăng M, đại lượng γ - tăng, còn đại lượng β - giảm, nghĩa là nên sử dụng chúng cho trường hợp dải thông bị hạn chế, chẳng hạn, trong mạng điện thoại với PCM.



Hình 1.15. Tính hiệu quả của hệ thống truyền tin.

Trên hình 1.15. cũng trình bày đường cong cho hệ thống với tín hiệu rời rạc (M = 2), được sử dụng để gia công tối ưu tín hiệu trong kênh bán liên tục (BLT) và cho hệ thống với tín hiệu rời rạc (M = 2, và M = 4), được sử dụng để gia công tín hiệu trong kênh rời rạc. Từ hình vẽ cũng

thấy rằng các hệ thống này theo tính hiệu quả của mình tiến gần đến giới hạn Shannon, khi γ bị hạn chế vì tín hiệu rời rạc.

$$\gamma_{max} = \frac{R_{max}}{B} = \frac{\log_2 M}{T.B} \tag{1.59}$$

Có sự khác nhau giữa hệ thống với kênh liên tục (LT) và kênh rời rạc (RR) theo chỉ tiêu hiệu quả β được giải thích như sau: trong hệ thống với kênh bán liên tục (BLT) việc gia công tối ưu các từ mã được thực hiện trong kênh liên tục (thu theo toàn phần); trong hệ thống với kênh rời rạc (RR) việc gia công tối ưu các tín hiệu được diễn ra theo ký hiệu mã trong kênh liên tục (thu theo ký hiệu), còn việc giải mã tối ưu các từ mã được tiến hành trong kênh rời rạc. Chúng ta giả thiết rằng trong trường hợp này, cũng như các trường hợp khác, mã chống nhiễu được sử dụng phù hợp với lý thuyết Shannon trong kênh có nhiễu.

CHƯƠNG 2

CÁC MÃ KINH TẾ

(Các mã hoá nguồn)

Trong chương này, trước tiên chúng ta nghiên cứu các loại mã tiết kiệm, coi chúng là cơ sở của các mã nén số liệu; sau đó chúng ta nghiên cứu các mã nén số liệu, gọi là các mã kinh tế. Các mã kinh tế là xu hướng của các mã hoá nguồn, vì mục đích của chúng để biến đổi các tin tức nguồn thành các bản mã, nhưng luôn *xây dựng theo hướng kinh tế*. Chúng ta nghiên cứu thuật toán nén số liệu quan trọng nhất, phổ thông nhất là thuật toán nén văn bản LZ77, với đuôi của file nén .ZIP; thuật toán nén ảnh tĩnh JPEG và thuật toán nén ảnh động (video) MPEG; cuối cùng các phương pháp mã hoá (nén) tín hiệu tiếng nói.

1. CÁC KHÁI NIỆN CHUNG VỀ MÃ

Tin tức (tiếng nói, âm nhạc, hình ảnh truyền hình v.v..) khi được gửi đến người sử dụng, với việc sử dụng hệ thống truyền tin điện tử, trong phần lớn trường hợp, không trực tiếp truyền các tin tức này, mà thông thường để việc truyền theo kênh liên lạc được hiệu quả phải thông qua các phép biến đổi gọi là *mã hoá*. Đối với các nguồn rời rạc (bao gồm các ký tự, bản vẽ) cũng đòi hỏi phép biến đổi (mã hoá) tương tự như vậy; được gọi chung là các phép *mã hoá nguồn*.

Sau đây, chúng ta trình bày một số khái niệm chung về mã:

* *Mã hoá*: trong trường hợp chung đối với các tin tức rời rạc, được hiểu là *phép biến đổi* các tin của tập tin $X\{x_i\}$ với $i = 1 \div N$, thành các từ mã của tập mã $C\{\alpha_j\}$ cũng với $j = 1 \div N$, để cho phép mã hoá có tính thuận nghịch một - một. Yêu cầu chung của phép mã, đối với các tin rời rạc, là không được làm tổn hao thông tin. Tuy nhiên, cần nhớ rằng các phép mã hoá nguồn tương tự là các phép mã hoá có tổn hao thông tin, hay có gây méo. Trong các trường hợp như thế, tin tức được khôi phục không còn chính xác tuyệt đối như tin tức ban đầu, trước khi mã hoá.

* *Giải mã:* là phép biến đổi ngược với phép mã hoá, tức là biến đổi các từ mã của tập mã $C\{\alpha_j\}$ thành ra các tin của tập tin $X\{x_i\}$. Nếu phép mã hoá không làm tổn hao thông tin, thì phép giải mã sẽ khôi phục được chính xác tuyệt đối bản tin gốc (tất nhiên, cũng phải giả thiết thêm rằng kênh truyền không nhiễu).

* *Cơ số của bộ mã*, thường ký hiệu là *m*: là số lượng các *dấu mã*, hoặc *ký hiệu (symbol) mã* khác nhau được sử dụng trong bộ mã đó. Chẳng hạn, mã nhị phân chỉ sử dụng hai loại ký hiệu là "0" và "1", khi đó m = 2; mã cơ số 16 sử dụng 16 ký hiệu khác nhau là 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F, khi đó m = 16; v.v...

Giữa các bộ mã với các cơ số m khác nhau, có thể chuyển đổi cho nhau. Chẳng hạn $(13)_{10} = (1101)_2$; $(2F9A)_{16} = (0010\_1111\_1001\_1010)_2$.

* *Từ mã:* vì thông thường số ký hiệu của bộ mã ít hơn số ký hiệu của tập tin, cho nên khi mã hoá mỗi ký hiệu (mỗi tin) của tập tin được biến đổi thành một từ mã, là một tập hợp nào đó của các dấu mã (ký hiệu mã). Chẳng hạn, ký tự F được mã hoá theo mã ASCII chuẩn, cho từ mã dài 7 ký hiệu: $\alpha_F = 1000110$.

* *Chiều dài của từ mã*, thường ký hiệu là n: là số lượng các ký hiệu mã có trong một từ mã nào đó. Chẳng hạn, với bộ mã ASCII chuẩn cho mọi từ mã có chiều dài n = 7, còn với bộ mã ASCII mở rộng cho mọi từ mã có chiều dài n = 8.

Căn cứ vào chiều dài của các từ mã trong bộ mã, người ta phân chia thành bộ mã đều và bộ mã không đều. *Bộ mã đều*, được hiểu là bộ mã, mà mọi từ mã của nó đều có cùng độ dài. Chẳng hạn, bộ mã ASCII. *Bộ mã không đều*, là bộ mã, mà các từ mã khác nhau, nói chung, có chiều dài khác nhau, tức là $n_1 \neq n_2 \neq .... \neq n_N$. Bộ mã không đều thường được sử dụng để làm các mã kinh tế.

* *Mã kinh tế*: là phép mã hoá (biến đổi) tin tức với mục đích *giảm bớt khối lượng ký hiệu mã* và *nâng cao tốc độ truyền tin trên kênh*, hoặc là *làm giảm dải thông tần yêu cầu đối với kênh truyền*. Mã kinh tế còn được gọi là *mã tiết kiệm, mã không có dư*, và cũng còn gọi là *nén số liệu*. Phần lớn các phép mã hoá nguồn đều thực hiện theo xu thế mã kinh tế (ngoại trừ, chẳng hạn, ASCII), vì vậy nghiên cứu các mã kinh tế cũng là nghiên cứu các mã hóa nguồn.

Tồn tại một lớp mã, mà phép mã hoá không gây tổn hao thông tin, xây dựng theo quan điểm *mã kinh tế* dựa trên cơ sở của định lý mã hoá thứ nhất của Shannon đối với trường hợp kênh không nhiễu. Cũng tồn tại nhiều phương pháp mã kinh tế khác được xây dựng không tuân theo nguyên tắc của định lý này (xem chương 1).

2. CÁC PHƯƠNG PHÁP BIỂU DIỄN MÃ

Một bộ mã thu được sau phép mã hoá, có thể được biểu diễn bằng nhiều phương pháp khác nhau. Sau đây, chúng ta khảo sát hai phương pháp đơn giản nhất, còn các phương pháp khác được xem xét ở các phần thích hợp tiếp theo (xem chương 3).

a) Phương pháp bảng

Xét một tập tin có 5 tin, $X\{x_1, x_2, x_3, x_4, x_5\}$, đem mã hoá bằng bộ mã không đều, được tập mã $C\{\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5\}$. Có thể biểu diễn mối quan hệ giữa các tin của tập tin với các từ mã của tập mã như bảng 1.

Bảng 1: Mối quan hệ giữa các tin và từ mã tương ứng

xi	x1	x2	x3	x4	x5
αi	00	01	11	100	101

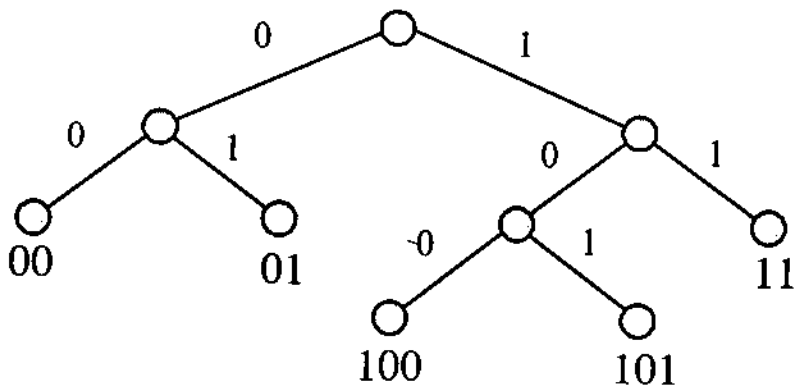
Phép biểu diễn này rất đơn giản, trực quan. Tuy nhiên, chỉ thích hợp cho các bộ mã ngắn. Cũng cần nhận xét rằng phép mã hoá như thế này, là phép biến đổi đơn trị, hay phép biến đổi không làm tổn hao thông tin, vì mỗi tin được mã hoá bằng một từ mã riêng biệt, vì vậy nó cho phép biến đổi ngược cũng đơn trị và khôi phục được chính xác tuyệt đối tin tức nguồn.

b) Phương pháp cây mã

Một bộ mã cũng rất thường được biểu diễn nhờ cây mã. Cây mã gồm *các nút* và *các nhánh*. Nút xuất phát của cây mã gọi là *nút gốc*. Từ nút gốc mọc ra 2 nhánh (nếu mã m phân sẽ có m nhánh, ở đây m = 2).

Nhánh dùng để ghi các ký hiệu mã, các nhánh đầu tiên để ghi các ký hiệu mã đầu tiên của các từ mã. Qui ước nhánh rẽ về bên trái của nút ghi ký hiệu 0, còn nhánh rẽ về bên phải ghi ký hiệu 1 (tất nhiên, cũng có thể qui ước ngược lại). Nhánh kết thúc ở nút bậc cao hơn. Từ nút bậc cao hơn, lại mọc ra 2 nhánh và lại qui ước ghi các ký hiệu mã như trên.

Các nhánh cứ tiếp tục mọc dài ra, tuỳ theo chiều dài của từ mã, cho đến khi dài bằng từ mã dài nhất thì thôi. Mỗi từ mã được xác định bằng các ký hiệu mã ghi trên các nhánh, bắt đầu từ nút gốc và kết thúc ở nút cuối cùng của nhánh. Trên hình 2.1a) biểu diễn cây mã cho bộ mã ở trên bao gồm các từ mã: 00 - 01- 11- 100 - 101.



Hình 2.1a). Cây mã biểu diễn bộ mã có 5 từ.

3. MÃ THỐNG KÊ TỐI ƯU (HAY TIẾT KIỆM) CỦA CÁC TIN TỨC RỜI RẠC, ĐỘC LẬP VỚI NHAU

Khi xây dựng định lý thứ nhất, Shannon đã đề nghị gọi mã đó là mã tối ưu, vì nó nâng cao năng suất của nguồn rời rạc, cũng gọi là thống kê, vì để thực hiện mã hoá cần phải biết trước xác suất xuất hiện của các tin ở đầu ra của nguồn (phải tính toán thống kê tin tức).

Như đã biết, *năng suất* (khả năng sản sinh tin) và *độ dư thừa* của nguồn rời rạc được xác định phù hợp theo công thức:

$$R(X) = \frac{H(X)}{\bar{\tau}} \qquad g_M = 1 - \frac{H(X)}{H(X)_{\max}}$$

Ở đây $\bar{\tau}$ - chiều dài trung bình của các tin, g_M - độ dư thừa của nguồn tin, còn $H(X)_{max} = \log N$ - entropy cực đại của nguồn, với N - kích cỡ các tin của nguồn.

Bây giờ chúng ta xét trước khi gửi bản tin lên kênh truyền, bản tin được mã hoá. Chẳng hạn, bằng bộ mã không đều và phép mã hoá không làm tổn hao thông tin. Lúc này mỗi tin x_i được mã bằng từ mã α_i, có chiều dài n_i. Chiều dài trung bình của các từ mã, ký hiệu là $\bar{n}$ – được định nghĩa:

$$\bar{n} = E\{n_i\} = \sum_{i=1}^{N} p(x_i).n_i \qquad (2.1)$$

Khi xác định công thức (2.1), chúng ta cho rằng xác suất xuất hiện các từ mã bằng xác suất xuất hiện các tin, hơn nữa điều đó hoàn toàn đúng. Ký hiệu E{.} - để chỉ toán tử kỳ vọng.

Mỗi từ mã có chiều dài (theo thời gian) $\tau_i = n_i\tau_0$, với τ_0- chiều dài mỗi ký hiệu và coi là như nhau, thì chiều dài trung bình (theo thời gian) của các từ mã $\bar{\tau} = \bar{n}.\tau_0$. Với bộ mã, chúng ta cũng đưa vào các định nghĩa *năng suất* (khả năng sản sinh từ mã) và *độ dư thừa* của bộ mã được xác định phù hợp theo công thức:

$$R(X) = \frac{H(X)}{\bar{n}\tau_0} \qquad g_C = 1 - \frac{H(X)}{\bar{n}} \qquad (2.2)$$

Trong công thức đầu, rõ ràng để tăng năng suất của nguồn tin cần xây dựng cơ chế mã hoá để chiều dài trung bình của bộ mã $\bar{n}$ phải nhỏ nhất có thể. Còn trong định nghĩa độ dư thừa g_C của bộ mã, thì $\bar{n}$ đóng vai trò $H(X)_{max}$ ở công thức xác định g_M .Tuy nhiên, để giảm độ dư thừa g_C, không sử dụng xu hướng tổ chức bộ mã để $H(X) \to H(X)_{max}$, vì H(X) vốn đã được xác định theo đặc trưng thống kê của tập tin, mà sử dụng cơ chế mã hoá để $\bar{n} \to H(X)$. Từ đây, tồn tại một phát biểu khác của định lý thứ nhất Shannon về chiều dài trung bình $\bar{n}$ của bộ mã:

Luôn luôn xây dựng được một phép mã hoá để chiều dài trung bình của bộ mã $\bar{n}$ có thể nhỏ tuỳ ý, nhưng không thể nhỏ hơn entropy của tập tin đem mã hoá.

Bất đẳng thức của định lý này có dạng:

$$H(X) \le \bar{n} \le H(X) + 1 \qquad (2.3)$$

Như vậy, có thể định nghĩa $\bar{n}_{min} = H(X)$. Một cơ chế mã hoá để $\bar{n} \to \bar{n}_{min}$ là cơ chế mã hoá tối ưu. Lúc này, bộ mã không có độ dư thừa.

Từ công thức (2.2), thấy rằng :

Để nâng cao năng suất của nguồn mã hoá (hoặc tốc độ truyền thông tin trên kênh) - R, cũng như để giảm độ dư thừa g_C của bộ mã, cả hai cần giảm chiều dài trung bình $\overline{n}$ của các từ mã. Đây chính là xu hướng của các mã kinh tế, cũng còn gọi là mã tiết kiệm của các tin tức rời rạc.

a) Nguyên tắc chung của cơ chế mã hoá tiết kiệm (không gây tổn hao thông tin):

Từ công thức (2.1), ta có: $\overline{n} = \sum_{i=1}^{N} p(x_i).n_i$, còn $H(X) = \sum_{i=1}^{N} p(x_i)\log\frac{1}{p(x_i)}$. Từ đây theo công thức (2.3) ta có:

$$\sum_{i=1}^{N} p(x_i).n_i \geq \sum_{i=1}^{N} p(x_i)\log\frac{1}{p(x_i)}$$

Để thực hiện được bất đẳng thức này, thì với mỗi số hạng của tổng (mỗi từ mã) cần thực hiện điều kiện: $n_i \geq \log\frac{1}{p(x_i)}$; từ đây chúng ta dẫn ra quan hệ: $n_i \sim \frac{1}{p(x_i)}$, ở đây dấu ~ có nghĩa là tỉ lệ. Cần phải thực hiện một cơ chế mã hoá theo nguyên tắc: *tin nào có xác suất xuất hiện $p(x_i)$ lớn, phải mã hoá bằng từ mã có chiều dài n_i nhỏ và ngược lại*; tức là phải thực hiện phép mã hoá để thoả mãn chuỗi bất đẳng thức:

$p(x_1) > p(x_2) > ... > p(x_N)$ - sắp xếp các tin của tập tin theo thứ tự của xác suất giảm dần (cũng có thể sắp xếp theo thứ tự ngược lại);

$n_1 < n_2 < ... < n_N$ - thực hiện mã hoá để được chuỗi bất đẳng thức về chiều dài này.

Sau đây, chúng ta khảo sát hai thuật toán thực hiện theo nguyên tắc này.

b) Mã Shannon - Fano

Thuật toán của mã này thực hiện theo ba bước sau:

1. Sắp xếp các tin của tập tin theo thứ tự của xác suất giảm dần.
2. Chia tập tin thành hai nhóm con, theo nguyên tắc có tổng xác suất các tin trong mỗi nhóm coi là tương đương nhau. Mỗi tin của nhóm con ở trên, ghi ký hiệu mã 1 (hoặc ngược lại, ghi ký hiệu mã 0), coi đó là ký hiệu mã đầu tiên của các từ mã thuộc nhóm con này. Mỗi tin của nhóm con ở dưới, ghi ký hiệu mã 0 (hoặc ngược lại, ghi ký hiệu mã 1), coi đó là ký hiệu mã đầu tiên của các từ mã thuộc nhóm con này.
3. Lặp lại bước 2 nhiều lần, cho đến khi mỗi nhóm con chỉ còn một tin, thì kết thúc bước mã hoá và xác định được các từ mã tương ứng.

Thí dụ 1: Để minh chứng cho thuật toán trên, chúng ta xét một thí dụ đơn giản. Cho một tập tin (đầy đủ) gồm 8 tin, với xác suất xuất hiện của các tin tương ứng như bảng 2a):

Bảng 2a): Các tin và xác suất tương ứng của chúng

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
$p(x_i)$	0,15	0,13	0,35	0,09	0,08	0,07	0,09	0,04

Chú ý: $\sum_{i=1}^{8} p(x_i) = 1$ - tập tin đầy đủ (hay toàn phần).

Nếu mã hoá tập tin này bằng *bộ mã đều*, thì mỗi từ mã có chiều dài n =3, vì $N = 2^3 = 8$ là số lượng các tin, cũng là số lượng các từ mã.

Entropy của tập tin: $H(X) = \sum_{i=1}^{8} p(x_i)\log\frac{1}{p(x_i)} = 2{,}69$,

Chiều dài trung bình các từ mã đều: $\bar{n} = \sum_{i=1}^{8} n_i p(x_i) = \sum_{i=1}^{8} n.p(x_i) = n = 3$,

Độ dư thừa của bộ mã đều, trong trường hợp này là:

$$g_{CD} = 1 - \frac{H(X)}{\bar{n}} = 1 - \frac{2{,}69}{3} \approx 0{,}103$$

Hơn nữa $g_{CD} = g_M$, tức là độ dư thừa của bộ mã bằng độ dư thừa của tập tin. *Mã hoá một tập tin bất kỳ bằng bộ mã đều, để bảo đảm tính không làm tổn hao thông tin, chỉ có thể làm cho* $g_{CD} \geq g_M$.

Xét ý nghĩa của độ dư thừa g_M ở công thức trên, có thể mở rộng như sau: về trung bình có thể nói cứ 100 ký tự (từ mã) thì có đến hơn 10 ký tự (từ mã) không chứa thông tin.

Bây giờ, chúng ta đem mã hoá tập tin này theo thuật toán Shannon- Fano, theo 3 bước vừa trình bày ở trên. Quá trình mã hoá được trình bày thành bảng 2b):

Bảng 2b): Quá trình mã hoá theo Shannon- Fano

Tin	$p(x_i)$	Các bước mã hoá	Từ mã	$n_i.\ p(x_i)$
x_3	0,35	00	00	0,7
x_1	0,15	01	01	0,3
x_2	0,13	100	100	0,39
x_4	0,09	101	101	0,27
x_7	0,09	1100	1100	0,36
x_5	0,08	1101	1101	0,32
x_6	0,07	1110	1110	0,28
x_8	0,04	1111	1111	0,16

Từ bảng 2b), chúng ta xác định được độ dài trung bình của các từ mã:

$$\overline{n} = \sum_{i=1}^{8} n_i . p(x_i) = 2,78,$$

còn độ dư thừa của bộ mã:

$$g_{CSF} = 1 - \frac{2,69}{2,78} = 0,032$$

Rõ ràng trong trường hợp này độ dư thừa g_{CSF} của bộ mã Shannon-Fano nhỏ hơn rất nhiều so với trường hợp sử dụng mã đều.

Bộ mã này, cũng như bộ mã Huffman xét sau đây, đều có tính PREFIX. Đặc tính này phát biểu như sau: *mọi từ mã có chiều dài dài hơn, đều không chứa ở phần đầu của chúng các từ mã có chiều dài ngắn hơn.* Nhờ đặc tính này mà bộ mã có tính phân tách được khi giải mã [Bạn đọc muốn tìm hiểu kỹ về tính chất này, có thể tham khảo thêm các tài liệu khác, chẳng hạn, [8]].

c) Mã Huffman

Thuật toán Huffman bao gồm 4 bước sau:

1. Sắp xếp các tin của tập tin theo thứ tự của xác suất giảm dần.
2. Hợp 2 xác suất của hai tin cuối bảng (2 tin có xác suất nhỏ nhất), một tin ghi ký hiệu mã 1, còn tin kia ghi ký hiệu mã 0 (tuỳ chọn). Xác suất tổng của 2 tin được coi là một tin mới và lại được sắp xếp các tin còn lại với tin mới theo thứ tự của xác suất giảm dần.
3. Lặp lại bước 2 nhiều lần, cho đến khi tổng xác suất các tin bằng 1, thì kết thúc quá trình mã hoá.
4. Từ mã ứng với mỗi tin được lập như sau:

* Đọc các ký hiệu mã theo chiều từ dưới lên trên, bắt đầu từ xác suất của tin tương ứng (tại vị trí bắt đầu bước hợp 2 xác suất).
* Ghi các ký hiệu mã của mỗi từ mã theo chiều từ phải qua trái (ghi theo trình tự ngược).

Thí dụ 2: Để minh chứng cho thuật toán này, chúng ta xét thí dụ ở bảng 3a) sau:

Bảng 3a): Các tin và xác suất tương ứng của chúng

Tin x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7
$p(x_i)$	0,1	0,06	0,34	0,07	0,23	0,01	0,19

Quá trình mã hoá theo Huffman được trình bày thành bảng 3b).

Từ bảng 3a,b), chúng ta xác định được entropy của tập tin và độ dài trung bình của các từ mã như sau:

$$H(X) = \sum_{i=1}^{7} p(x_i)\log_2\left(\frac{1}{p(x_i)}\right) = 2{,}3825 \qquad \bar{n} = \sum_{i=1}^{7} n_i.p(x_i) = 2{,}45$$

Độ dư thừa của tập tin và của bộ mã:

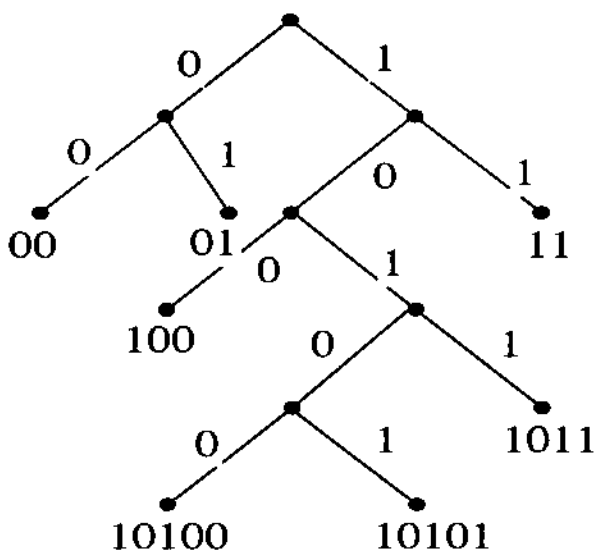
$$g_M = 1 - \frac{H(X)}{H(X)_{\max}} = 1 - \frac{2{,}3825}{2{,}8073} = 0{,}1513 \qquad g_{CH} = 1 - \frac{H(X)}{\bar{n}} = 1 - \frac{2{,}3825}{2{,}45} \approx 0{,}0276$$

Bảng 3b): Quá trình mã hoá theo Huffman

Các tin x_i	Xác suất $p(x_i)$	Các bước mã hóa					Từ mã α_1	$n_i.p(x_i)$
x_3	0,34	0,34	0,34	0,34	0,42	0,58	11	0,68
x_5	0,23	0,23	0,23	0,24	0,34	0,42	01	0,46
x_7	0,19	0,19	0,19	0,23	0,24		00	0,38
x_1	0,10	0,10	0,14	0,19			100	0,30
x_4	0,07	0,07	0,10				1011	0,28
x_2	0,06	0,07					10101	0,30
x_6	0,01						10100	0,05
								2,45

Cây mã của bộ mã ở thí dụ 2, như hình 2.1b)

Nếu đem mã hóa tập tin trên (thí dụ 2) theo mã đều, chúng ta cần sử dụng các từ mã có độ dài n = 3, vì $N = 2^3 = 8 > 7$. Do đó chiều dài trung bình của bộ mã $\bar{n} = n = 3$.



Hình 2.1b). Cây mã biểu diễn bộ mã có 7 từ (thí dụ 2).

Độ dư thừa của bộ mã đều, trong trường hợp này, được xác định như sau:

$$g_{CĐ} = 1 - \frac{H(X)}{\bar{n}} = 1 - \frac{2{,}3825}{3} = 0{,}206$$

Như vậy, $g_{CĐ} > g_M$, tức là phép mã hóa đều, trong trường hợp này, đã làm tăng độ dư thừa.

Cả hai thuật toán mã tiết kiệm xét ở trên đều có tác dụng làm giảm nhỏ độ dư thừa của bộ mã đi nhiều lần.

Nhược điểm của mã Huffman, cũng như mã Shannon- Fano:

- Để mã hoá cần phải biết trước xác suất (chính xác là tần suất) xuất hiện của các tin, điều này rõ ràng là không dễ dàng. Để khắc phục nhược điểm này có phương pháp Huffman *động*, trong đó tần suất các tin được xác định đúng dần dần cho đến khi xem xét hết bản tin. Tuy nhiên, hiệu quả nén không cao.
- Chiều dài cực tiểu của từ mã không thể nhỏ hơn 1, trong khi đó entropy của tin tức có thể nhận giá trị lớn hơn 0, bé tuỳ ý, chẳng hạn 0,1 hoặc 0,01 bit/ tin. Vì vậy, các mã này gần như luôn luôn *có độ dư thừa*.
- Các mã Huffman, mã Shannon-Fano chỉ đạt được tối ưu (không có độ dư thừa) khi xác suất các tin là bội nguyên âm của 2, tức là bằng: 1/2; 1/4; 1/8; v.v...

Hai dạng mã kinh tế vừa xét ở trên, ngày này không được sử dụng trực tiếp để nén số liệu. Tuy nhiên, chúng đưa ra ý tưởng đầu tiên về phép nén số liệu, hơn nữa bản thân chúng cũng vẫn còn được sử dụng làm mã thành phần cho nhiều loại mã nén, mà chúng ta xét dưới đây.

4. CÁC MÃ CÓ NHỚ

Các mã có nhớ thường được khảo sát ở hai dạng sau:

* *Các mã khối*;

* *Các mã có nhớ hữu hạn.*

a). Mã khối: Mã khối thực hiện chia véctơ số liệu (ở dạng nhị phân) ra thành các khối có độ dài hữu hạn. Sau đó xác định tần suất của các khối này, rồi đem mã hoá theo các phương pháp mã tiết kiệm đã xét ở mục 3. Kết quả cho ta một mã tiết kiệm hơn. Về mã khối ta nói rằng nó là mã khối bậc k, nếu tất cả các khối có độ dài k ký hiệu

Xét thí dụ sau: cho một véctơ số liệu X = (0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1). Trước hết chia dãy số liệu X ra thành các khối có độ dài 2: 01, 10, 10, 01, 11, 10, 01, 01. Sau đó coi chúng được hình thành từ các ký hiệu mới (01, 10, 11) có tần suất (4/8, 3/8, 1/8). Đem mã hoá dãy ký hiệu mới theo các phương pháp mã tiết kiệm ở mục 3, được kết quả như bảng 4.

Từ bảng mã này, chúng ta nhận được dãy từ mã như sau:

0, 10, 10, 0, 11, 10, 0, 0.

Nối các từ mã lại cùng nhau, chúng ta có dãy số liệu được gọi là dãy số liệu nén, ký hiệu B(X):

B(X) = (0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0)

Bảng 4. Bảng mã hoá

Khối	Từ mã
01	0
10	10
11	11

Xác định các tham số đặc trưng của bộ mã:

Entropy $H(X) = -\sum_{i=0}^{i} p(x_i).\log p(x_i) = 0{,}9887$ bit/ký hiệu;

Độ dài trung bình của bộ mã $\overline{n} = \sum_{i=1}^{3} n_i.p(x_i) = \frac{4}{8} + \frac{2\times 3}{8} + \frac{2\times 1}{8} = 1{,}5$ bit/ký hiệu

Độ dư thừa của bộ mã $g_C = 1 - \frac{H(X)}{\overline{n}} = 1 - \frac{0{,}9887}{1{,}5} = 1 - 0{,}659 = 0{,}341$

Tốc độ nén, được định nghĩa R = (số lượng bít sau nén)/(số lượng bít trước nén); trong trường hợp này bằng R = 12/16 = 0,75

b). Mã có nhớ hữu hạn: Khi mã hoá véctơ số liệu (x_1, x_2,..., x_N), sẽ sử dụng một quyển sách mã, bao gồm từ một vài loại mã không nhớ khác nhau. Mỗi bộ chọn của số liệu được mã hoá bằng mã không nhớ từ quyển sách mã, bằng cách xác định giá trị của số nào đó đứng trước số liệu chọn.

Ta xét một thí dụ truyền thống, véctơ số liệu X = ABRACADABRA. Trong chuỗi này rõ ràng có sự phụ thuộc thống kê mạnh giữa các ký tự tiếp sau, mà cần phải tính đến khi chọn phương pháp mã hoá.

Mã có nhớ *khi mã hoá mỗi ký tự chạy sẽ tính đến giá trị của ký tự đứng trước nó*. Như vậy, từ mã của ký tự chạy A sẽ khác nhau trong tổ hợp RA, DA, CA (mã này nhớ một ký tự).

Bảng 5. Bảng mã hoá

Ký tự, ký tự đứng trước	Từ mã
(A,-)	1
(B,A)	0
(C,A)	10
(D,A)	11
(A,R)	1
(R,B)	1
(A,C)	1
(A,D)	1

Kết quả mã hoá, được véctơ B(X) = (1011101111011) có độ dài 13 bit. Nếu đem mã hoá véctơ X (11 ký tự) bằng mã đều, cần sử dụng bộ mã có chiều dài các từ mã n = 4. Như vậy, véctơ X dài 44 bit. Tốc độ mã hoá R = 13/44 = 0,29.

Nếu đem mã hoá véctơ X bằng bộ mã tiết kiệm không nhớ (xét ở mục 3).

Bảng 6. Bảng mã hoá X theo mã không nhớ

Ký tự	$p(x_i)$		Từ mã	$n_i.p(x_i)$
A	5/11	0	0	5/11
B	2/11	1 0	10	4/11
R	2/11	1 1 0	110	6/11
C	1/11	1 1 1 0	1110	4/11
D	1/11	1 1 1 1	1111	4/11

Chuỗi bít sau mã hoá không nhớ $B_0(X)$ = (01011001110011110101100) dài 23 bit. Như vậy, so sánh giữa mã có nhớ và mã không nhớ, thì tốc độ mã trong trường hợp này là: R = 13/23 = 0,565. Rõ ràng mã có nhớ kinh tế hơn mã không nhớ.

5. CÁC MÃ SỐ HỌC

Khi mã hoá số học, sẽ khác với các phương pháp mã đã khảo sát trước đây, lúc này ký tự để mã hoá (hoặc nhóm ký tự) sẽ được thay thế bằng mã phù hợp của nó, *kết quả của phép mã hoá tất cả tin tức được trình bày bằng một, hoặc một cặp số thực trong khoảng từ 0 đến 1*. Theo trình tự mã hoá của văn bản gốc, khoảng ánh xạ của chúng được giảm nhỏ dần, còn số lượng của các chữ số hệ 10 (hoặc hệ 2), dùng để trình bày nó sẽ tăng dần. Các ký tự của văn bản gốc sẽ rút gọn trị số khoảng ban đầu từ giá trị xác suất của chúng, phù hợp theo mô hình xác định. Chúng ta khảo sát một thí dụ đơn giản của phương pháp mã số học. Chẳng hạn, đem mã hoá chuỗi ký tự RADIOVIZIR (10 ký tự).

Trước hết, xác định tần suất xuất hiện của các ký tự này, kết quả được trình bày thành bảng 6.

Việc xếp đặt các ký tự vào bảng có thể theo một trình tự bất kỳ, có thể theo trình tự xuất hiện của nó trong chuỗi ký tự, có thể theo sự tăng dần của tần suất xuất hiện các ký tự, điều này là tuỳ chọn, kết quả mã hoá sẽ khác nhau, nhưng hiệu quả đều như nhau.

a). Mã hoá

Trước khi mã hoá, khoảng xuất phát là khoảng [0,0 - 1,0), với độ rộng 1,0. Chú ý: dấu [- chỉ khoảng đóng, còn dấu) - chỉ khoảng mở.

Ký tự R được chọn làm ký tự mã hoá đầu tiên, khoảng của nó giữ nguyên như khoảng ban đầu, tức là khoảng [0,8 - 1,0), gọi là khoảng thứ nhất, với độ rộng 0,2.

Bảng 6. Kết quả xác định tần suất và khoảng tương ứng của các ký tự

Ký tự	Tần suất	Khoảng ban đầu
A	0,1	0 - 0,1
D	0,1	0,1 - 0,2
V	0,1	0,2 - 0,3
I	0,3	0,3 - 0,6
Z	0,1	0,6 - 0,7
O	0,1	0,7 - 0,8
R	0,2	0,8 - 1

Các ký tự tiếp theo được mã hoá theo cách khoảng của chúng được làm hẹp dần lại, hơn nữa khoảng của chúng được nằm trong khoảng trước đó.

Ký tự A được chọn làm ký tự được mã hoá thứ 2, khoảng của nó được làm hẹp lại theo cách: giá trị đầu và cuối của khoảng mới được xác định bằng cách bổ sung vào giá trị đầu của khoảng trước các giá trị được tính toán theo nguyên tắc: giá trị bổ sung cho đầu khoảng = độ rộng khoảng trước × giá trị đầu của khoảng xuất phát (xem bảng 6); còn giá trị bổ sung cho cuối khoảng = độ rộng khoảng trước × giá trị cuối của khoảng xuất phát (xem bảng 6). Trình tự tính toán như sau:

- giá trị khoảng trước = 1 - 0,8 = 0,2;
- giá trị đầu của khoảng xuất phát = 0;
- vậy giá trị bổ sung cho đầu khoảng = 0,2× 0,0 = 0,00;
- giá trị đầu khoảng = 0,8 + 0,00 = 0,80;
- giá trị cuối của khoảng xuất phát = 0,1;
- vậy giá trị bổ sung cho cuối khoảng = 0,2× 0,1 = 0,02;
- giá trị cuối khoảng = 0,8 + 0,02 = 0,82;

Khoảng mới của ký tự A (gọi là khoảng thứ 2) sẽ là: [0,80 - 0,82).

Ký tự D được chọn làm ký tự được mã hoá thứ 3, khoảng của nó được làm hẹp lại, để bổ sung vào khoảng thứ 2. Cách tính toán tương tự như trên. Kết quả khoảng thứ 3 sẽ là: [0,802 - 0,804).

Tiếp theo là các ký tự V, I, Z, O được mã hoá. Kết quả trình bày thành bảng 7.

Kết quả mã hoá, chúng ta nhận được khoảng: [0,8025002 0,8025008]. Để giải mã duy nhất chỉ cần biết một giới hạn khoảng: hoặc dưới, hoặc trên, có nghĩa là kết quả mã hoá chỉ cần giữ lại một giới hạn, chẳng hạn khoảng dưới - 0,8025002, hơn nữa chỉ quan tâm đến các số sau dấu phẩy, tức là số nguyên 8025002, là một số hệ 10 gồm 7 con số. Chuyển đổi dãy số này sang hệ cơ số 2, cần sử dụng 20 bit, vì 2^{19} = 780288, còn 2^{20} = 1560576, trong khi đó 780288 < 8025002

<1560576. Đem so sánh với bản tin ban đầu gồm 10 ký tự, đem mã hoá bằng mã ASCII cần sử dụng 10×8 = 80 bit, chúng ta thấy hệ số nén: $r = \frac{80}{20} = 4$.

Bảng 7. Quá trình mã hoá mã số học

Ký tự	Khoảng mã hoá		Độ rộng khoảng
khoảng ban đầu	[0,0	1,0)	1
R	[0,8	1,0)	0,2
A	[0,80	0,82)	0,02
D	[0,802	0,804)	0,002
V	[0,8024	0,8026)	0,0002
I	[0,80246	0,80252)	0,00006
Z	[0,802496	0,802502)	0,000006
O	[0,8025002	0,8025008)	0,0000006

b). Giải mã

Khi giải mã chúng ta giả thiết rằng (tất nhiên) biết giá trị đầu của khoảng cuối mã hoá, tức là 0,8025002, cũng như quá trình mã hoá phải biết bảng tần suất và khoảng ban đầu của các ký tự, tức là bảng 6. Chúng ta biết ngay rằng ký tự đầu tiên giải mã được là ký tự P, bởi vì số 0,8 đứng đầu kết quả đem giải mã. Bây giờ chúng ta lặp lại hoạt động của mã hoá:

khoảng ban đầu [0,0 1,0]

khoảng mã hoá của P [0,8 - 1)

Để tìm ra ký tự giải mã thứ 2, chúng ta đem kết quả giải mã, tức 0,8025002 trừ đi giá trị đầu khoảng của P, tức 0,8, để loại trừ ảnh hưởng mã hoá ký tự P. Kết quả được: 0,8025002 - 0,8 = 0,0025002. Đem chia kết quả này cho độ rộng khoảng ký tự P: 0,0025002/0,2 = 0,012501. Giá trị 0,0 này phản ảnh giá trị đầu khoảng (hoặc nằm trong khoảng) của ký tự A - [0 - 0,1). Như vậy, ký tự giải mã thứ 2 là A.

Để tìm ký tự đem giải mã thứ 3, chúng ta đem 0,012501 trừ đi giá trị đầu khoảng của ký tự A, để loại trừ ảnh hưởng mã hoá của ký tự A: 0,012501- 0,0 = 0,012501, rồi chia cho độ rộng khoảng của ký tự A: 0,012501/0,1 = 0,12501. Giá trị 0,1 này phản ảnh giá trị đầu khoảng của ký tự D - [0,1 - 0,2). Như vậy, ký tự giải mã thứ 3 là D.

Để tìm ký tự giải mã thứ 4, chúng ta đem 0,12501 trừ đi giá trị đầu khoảng của ký tự D, để loại trừ ảnh hưởng mã hoá của ký tự D: 0,12501 - 0,1 = 0,02501, rồi đem chia cho 0,1 là độ rộng khoảng của D: 0,02501/0,1 = 0,2501. Giá trị 0,2 phản ảnh giá trị đầu khoảng của ký tự V - [0,2 - 0,3). Như vậy, ký tự giải mã thứ 4 là V.

Tương tự, chúng ta có phép tính: (0,2501 - 0,2)/0,1 = 0,501. Giá trị 0,5 nằm trong khoảng (chứ không nhất thiết phải là giá trị đầu khoảng) của ký tự I - [0,3 - 0,6); (0,501 - 0,3)/0,3 = 0,67 → Z; (0,67 - 0,6)/0,1 = 0,7 → O.

Quá trình giải mã được trình bày thành bảng 8.

Bảng 8. Kết quả giải mã

Giá trị độ rộng khoảng mã hoá	Ký tự ở đầu ra giải mã	Khoảng ban đầu dưới — trên	Độ rộng khoảng
0,8025002	P	0,8 - 1,0	0,2
0,012501	A	0,0 - 0,1	0,1
0,12501	D	0,1 - 0,2	0,1
0,2501	V	0,2 - 0,3	0,1
0,501	I	0,3 - 0,6	0,3
0,67	Z	0,6 - 0,7	0,1
0,7	O	0,7 - 0,8	0,1

6. MỤC ĐÍCH NÉN SỐ LIỆU VÀ CÁC DẠNG CỦA HỆ THỐNG NÉN

Việc truyền và lưu giữ thông tin yêu cầu một chi phí lớn. Khối lượng thông tin càng nhiều, thì chi phí này càng đắt. Đáng tiếc là trong phần lớn trường hợp việc truyền, cũng như lưu giữ thông tin, chúng đã không được trình bày ở dạng nén. Phần lớn các số liệu khi lưu giữ chúng được trình bày ở dạng đơn giản dễ sử dụng nhất, chẳng hạn, ở dạng mã ASCII. Dạng trình bày đơn giản đó chiếm một dung lượng lớn khi lưu giữ và chiếm thời gian lớn khi truyền, hoặc không đủ dải thông để truyền chúng. Bởi vậy, nén số liệu là một trong các *xu hướng bức xúc nhất* trong kỹ thuật truyền tin hiện đại.

Như vậy, mục đích của nén số liệu là bảo đảm việc trình diễn số liệu, do nguồn tin sinh ra, ở dạng nén chặt, để kinh tế hơn khi truyền cũng như khi lưu giữ chúng.

Sau đây sẽ trình bày cấu trúc chung của các hệ thống nén số liệu:

Số liệu của nguồn→Mã hoá→Nén số liệu→(truyền, lưu giữ)→Giải nén→Giải mã→Khôi phục số liệu.

Trong sơ đồ này: mã hoá và giải mã được hiểu phép mã hoá nguồn, giải mã nguồn. Nén số liệu là phép biến đổi số liệu ở dạng nhị phân thành dạng số liệu được nén chặt. Còn giải nén là phép biến đổi ngược lại. *Trong nhiều trường hợp phép mã hoá nén và phép mã hoá nguồn là một.* Về bản chất nén cũng là một phép mã hoá kinh tế.

Hiện tại có hai dạng của hệ thống nén số liệu:

- Hệ thống nén không làm tiêu hao thông tin (nén không tiêu hao);

- Hệ thống nén làm tiêu hao thông tin (nén có tiêu hao).

a). Nén không tiêu hao thông tin

Trong hệ thống nén không làm tiêu hao thông tin, thì phép giải nén sẽ khôi phục lại chính xác tuyệt đối bản tin nguồn. Như vậy, cấu trúc hệ thống nén này như sau:

Véctơ số liệu X → *Mã hoá* X → *Nén* → B(X) → *Giải nén* → *Giải mã* → X

Véctơ số liệu X để nén, là tập tin $X = (x_1, x_2,..., x_N)$ có độ dài hữu hạn. Các tin x_i là các ký hiệu của văn bản, bản vẽ. Kích cỡ của tập tin N là hữu hạn, nhưng nó có thể lớn tuỳ ý. Như vậy, nguồn số liệu là tập tin X có kích cỡ là N các ký hiệu.

Mã hoá, được hiểu là phép mã hoá nguồn, thông thường là mã ASCII, mỗi ký hiệu x_i cho một từ mã dài 8bit. Dãy số liệu nhị phân (cũng sử dụng ký hiệu X) này được đưa vào nén- nén thực chất cũng là phép mã hoá- ở đầu ra cho dãy số liệu nén $B(X) = (b_1, b_2,..., b_K)$ kích cỡ K phụ thuộc vào X. Bởi vì hệ thống nén không làm tổn hao thông tin, cho nên với hai véctơ nguồn như nhau $X_L = X_M$, thì sau phép nén cũng nhận được hai véctơ nén như nhau $B(X_L) = B(X_M)$.

Khi giải quyết bài toán nén số liệu, một cách tự nhiên, dẫn đến vấn đề đánh giá tính hiệu quả của các hệ thống nén khác nhau. Thông thường sử dụng hai tham số để đánh giá tính hiệu quả của hệ thống nén sau:

1) Hệ số nén, ký hiệu r, được xác định như một tỉ số:

r = (kích cỡ số liệu nguồn [bit])/ (kích cỡ số liệu nén [bit])

Nếu r = 2, có nghĩa là khối lượng bit ở bản nén B(X) bằng $\frac{1}{2}$ số lượng bit ở văn bản gốc X. Hệ số nén r càng lớn, hệ thống nén làm việc càng tốt.

Cùng với hệ số nén, tính hiệu quả của hệ thống nén còn được đánh giá bằng tham số đặc trưng cho tốc độ nén.

2) Tham số đặc trưng cho tốc độ nén, ký hiệu R, được xác định như một tỉ số:

$R = \frac{K}{N}$ = (kích cỡ số liệu nén [bit])/ (kích cỡ số liệu nguồn [bit])

Một hệ thống nén có hệ số nén r càng lớn, thì tốc độ nén càng nhỏ. Vì $r = \frac{1}{R}$

b). Hệ thống nén với tiêu hao thông tin

Trong hệ thống nén với tiêu hao thông tin (nén có tiêu hao), thì sau khi giải nén, cũng như giải mã nguồn việc khôi phục thông tin sẽ không chính xác như ở bản tin đầu. Sơ đồ cấu trúc của hệ thống nén có tiêu hao thông tin như sau:

Tin tức S→*Mã hoá nén có tiêu hao* (X→*lượng tử hoá*→X^q) →$B(X^q)$ →*Giải mã nén* X^*→ *tin tức* S^*

Trong hệ thống này, tin tức S là tín hiệu liên tục theo thời gian (tín hiệu tương tự). Như đã biết, quá trình mã hoá nguồn tương tự là quá trình làm tiêu hao lượng thông tin (do quá trình lấy mẫu và lượng tử hoá). Ở quá trình mã hoá nén cũng gây tiêu hao thông tin. Ta xét quá trình gây tiêu hao thông tin ở quá trình mã hoá nén. Gọi X là một mẫu của tín hiệu đã được số hóa, có độ lớn bất kỳ X = (x_1, x_2,..., x_N), sau quá trình lượng tử hoá, độ lớn các mẫu này được làm tròn bằng nguyên lần bước lượng tử hoá, được biểu diễn là X^q. Còn ở bước tiếp theo được thực hiện như nhau với mọi mẫu, tức là với hai mẫu như nhau $X_L^q = X_M^q$, thì quá trình nén cho số liệu nén B(X_M^q) = B(X_L^q). Ở quá trình giải mã nén sẽ khôi phục lại mẫu $X^* = (x_1^*, x_2^*, ..., x_N^*)$ gần đúng với mẫu X, tức là $X^* \approx X$ theo nghĩa khoảng cách trung bình bình phương. Sau giải mã nguồn tin tức được khôi phục cũng chỉ gần giống với tin tức gốc, tức là $S^* \approx S$.

Ở hệ thống nén với tiêu hao được đặc trưng bởi hai tham số: *tốc độ nén R* và *độ méo D*, được xác định:

$$R = \frac{K}{N} = (\text{số lượng bit sau nén/ Số lượng bit trước nén})$$

$$D = \frac{1}{L}\sum_{i=0}^{L-1}(X_i - X_i^*)^2$$, với L- số cặp mẫu được so sánh

Tham số R đặc trưng cho tốc độ nén, còn tham số D đặc trưng cho độ lệch trung bình bình phương giữa các mẫu X^* và X.

Nếu hai hệ thống nén có tiêu hao thông tin, một hệ thống có tham số R_1, D_1 còn hệ thống kia có tham số R_2, D_2. Hệ thống thứ nhất sẽ tốt hơn nếu $R_1 < R_2$, $D_1 < D_2$. Tuy nhiên, rất đáng tiếc là không thực hiện được điều như vậy. Vì tốc độ nén R và độ méo D là hai tham số quan hệ nghịch đảo với nhau. Bởi vậy, với hệ thống nén có tiêu hao, hoặc chọn cực tiểu tốc độ nén khi cho trước độ méo, hoặc chọn cực tiểu độ méo khi cho trước tốc độ nén.

Trong thực tiễn việc chọn lựa hệ thống nén không tiêu hao, hay hệ thống nén có tiêu hao là tuỳ thuộc vào số liệu được nén. Khi nén các tin tức rời rạc như: văn bản, chương trình máy tính, bản vẽ v.v.. thì phải ứng dụng hệ thống nén không tiêu hao; bởi vì cần khôi phục chính xác số liệu ban đầu. Khi nén số liệu là tín hiệu tiếng nói, âm nhạc, video v.v.., thì ngược lại phải chọn hệ thống nén có tiêu hao (phải sử dụng các phép mã hoá có tiêu hao thông tin), bởi vì phải chấp nhận một độ méo cho phép để bảo đảm tốc độ nén nhỏ đi vài lần. Vì vậy, hệ thống nén có tiêu hao bảo đảm hệ số nén cao hơn nhiều so với hệ thống nén không tiêu hao.

Sau đây chúng ta trình bày một số thí dụ, để biểu diễn các thủ tục nén, các phương pháp đơn giản của mã kinh tế và hiệu quả nén số liệu.

Thí dụ 1: Giả sử có một nguồn phát sinh ảnh số với 512×512 phần tử (pixel) có 256 màu. Mỗi màu được biểu diễn bằng một số từ tập hợp (0, 1, 2,..., 255). Biểu diễn toán học của ảnh này là ma trận 512×512, mỗi phần tử ảnh nằm trong tập hợp (0, 1, 2,..., 255).

Mỗi phần tử thuộc tập hợp (0, 1, 2,..., 255) lại được biểu diễn trong dạng nhị phân với 8bit cho một phần tử. Như vậy, kích cỡ số liệu nguồn chiếm $512\times512\times 8 = 2^{21}$ hay là 2,1 Mbit.

Một đĩa cứng có dung lượng 1Gbit, thì có thể chứa 5000 ảnh (một videoclip dài 5 phút chẳng hạn), nếu nó không được nén. Nếu các ảnh này được nén, chẳng hạn với hệ số nén r = 10, thì đĩa này có thể chứa được 1 giờ video phim.

Chúng ta lại giả thiết tiếp rằng muốn truyền bức ảnh này theo đường dây điện thoại với vận tốc 14000 bit/sec, thì thời gian để truyền xong bức ảnh sẽ là $(21.10^6/14.10^3)/60 = 3$ phút. Khi nén với hệ số r = 40, thì thời truyền sẽ chỉ là 5 sec.

Thí dụ 2: Chọn nguồn số liệu để nén là một mảnh ảnh kích thước 4×4 pixel, bao gồm 4 màu: R = đỏ, 0 = da cam, Y = xanh biếc, G = xanh lá cây.

Bảng 9. Bảng phân phối màu của mảnh ảnh

R	R	O	Y
R	O	O	Y
O	O	Y	G
Y	Y	Y	G

Biểu diễn mảnh ảnh này theo dòng, còn mỗi màu được gắn một chỉ số, chẳng hạn, R = 3, O = 2, Y = 1, G = 0, kết quả ta nhận được một véctơ số liệu:

X = (3, 3, 2, 1, 3, 2, 2, 1, 2, 2, 1, 0, 1, 1, 1, 0)

Để nén số liệu, trước hết xác định tần suất của mỗi số, sau đó đem mã hoá X theo các phương pháp mã kinh tế trình bày ở mục 3.

Bảng 10. Bảng mã hoá theo Shannon- Fano

Chỉ số màu	Từ mã
3	001
2	01
1	1
0	000

Sử dụng bảng mã hoá, thay thế mỗi phần tử của véctơ X bằng từ mã tương ứng của bảng 10 (mã không nhớ).Ta nhận được véctơ nén số liệu B(X), véctơ B(X) có dạng sau:

B(X) = (0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1 0, 0, 0, 1, 1, 1, 0, 0, 0)

Hệ số nén trong trường hợp này r = 32/31= 1,03.

Thí dụ 3: Chúng ta so sánh hai loại mã khác nhau, khi thực hiện nén cùng một véctơ số liệu:

X = ABRACADABRA.

Véctơ X có 11 ký tự, nếu đem mã hoá bằng bộ mã đều phải sử dụng từ mã có chiều dài n = 4. Kết quả véctơ X (ở dạng nhị phân) dài 44 bit.

Mã thứ nhất là loại mã không nhớ. Bảng mã hoá (sau khi đã xác định tần suất của các ký tự, rồi mã hoá theo 1 trong 2 phương pháp mã ở mục 3), được dẫn ra như sau:

Bảng 11. Bảng mã hoá 1

Ký tự	Từ mã
A	0
B	10
R	110
C	1110
D	1111

Mã thứ hai, khi mã hoá mỗi ký tự chạy phải nhớ đến ký tự đứng trước đó. Như vậy, từ mã đối với ký tự chạy A sẽ khác nhau trong các trường hợp RA, DA, CA (tương tự đối với các ký tự khác, *mã này có đặc tính nhớ một ký đứng trước*) Từ mã phù hợp với véctơ X, tương ứng với hai loại mã như sau:

$B_1(X) = 01011001110011110101100.$

$B_2(X) = 10111011111011.$

Như vậy, hệ số nén khi sử dụng mã 1 (mã không nhớ) là: 44/23 = 1,9, trong khi đó hệ số nén của mã thứ (mã có nhớ) là: 44/14 = 3,14. Sử dụng mã thứ 2 có ưu điểm hơn, mặc dầu nó phức tạp hơn.

Bảng 12. Bảng mã hoá 2

Ký tự, ký tự đứng trước	Từ mã
(A,-)	1
(B,A)	0
(C,A)	10
(D,A)	11
(A,R)	1
(R,B)	1
(A,C)	1
(A,B)	1

7. CÁC PHƯƠNG PHÁP NÉN SỐ LIỆU KHÔNG TIÊU HAO. PHƯƠNG PHÁP LEMPEL- ZIV

Thực tế tất cả phương pháp nén số liệu không tiêu hao đều thuộc họ các thuật toán từ công trình của hai nhà bác học người Israel là Abraham Lenpel và Jacob Ziv, được xuất bản vào năm 1977. *Bản chất của phương pháp này là các đoạn trong văn bản để nén được thay thế bằng các chỉ số (mang thông tin về vị trí, chiều dài của đoạn) mà chúng đã xuất hiện trước đây trong văn bản đó.*

Họ các thuật toán này được gọi là phương pháp Lempel- Ziv và ký hiệu là nén -LZ. Các five văn bản sử dụng nén - LZ có đuôi .ZIP. Gần đây chúng ta thấy ngoài thuật toán nén mà file có đuôi ZIP (Win.ZIP), còn xuất hiện chuẩn nén số liệu, mà file có đuôi RAR (Win.RAR)

7.1. Giới thiệu nén số liệu nhờ ZIP

Nén số liệu nhờ ZIP, được tạo ra bởi các tác giả Jean-loup Gailly, Mark Adler và Richard Wales. ZIP là một chương trình được phổ biến tự do, được viết bằng ngôn ngữ C chạy trên hệ điều hành Unix và một vài hệ điều hành khác. ZIP có chức năng tương đương PKZIP, được truy cập rộng rãi miễn phí chạy trong hệ điều hành Windows, được gia công bởi PKWARE, Inc. Thuật toán ZIP là kỹ thuật nén số liệu được sử dụng thường xuyên nhất đã được tích hợp trong hệ điều hành Windows, Unix, Macintosh và các hệ điều hành khác.

Thuật toán LZ77 và các phương án của nó dựa trên yếu tố là các từ và các đoạn bên trong của dòng văn bản (hoặc cấu trúc của ảnh, trong trường hợp GIF), có sự lặp lại theo xác suất. Khi tìm thấy sự lặp lại thì từ đó (hoặc đoạn đó) được thay thế bằng một đoạn mã ngắn hơn. Chương trình nén cần tìm ra các đoạn lặp lại như thế và xây dựng các đoạn mã để thay thế trực tiếp cho các đoạn này. Trong các đoạn mã tiếp theo, sử dụng sự lặp lại để gia công các chuỗi mới. Thuật toán cần phải xây dựng sao cho chương trình giải nén có thể xây dựng theo qui tắc ảnh ngược với mã hoá, để có thể khôi phục lại số liệu ban đầu.

Trước khi xem xét chi tiết thuật toán LZ77, chúng ta hãy khảo sát một thí dụ đơn giản. Ở đây nêu ra một câu không có ý nghĩa, nhưng tiện cho phân tích như sau:

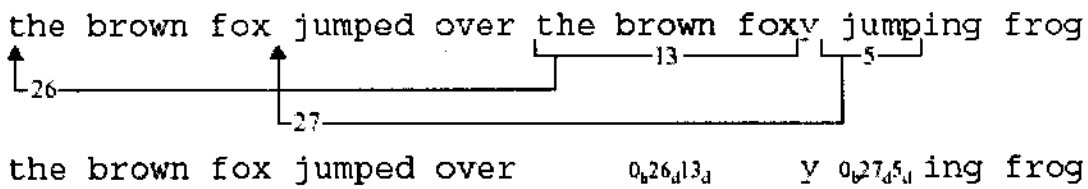
the brown fox jumped over the brown foxy jumping frog,

độ dài của câu là 53 octet (53byte) hoặc là 424 bit. Thuật toán gia công câu này từ trái qua phải. Mỗi kí tự không được nén được gia công thành 9 bit nhị phân, bao gồm một bit tiêu đề, có giá trị 1 và 8bit mã ASCII của kí tự. Trong quá trình thực hiện tiếp theo, thuật toán tìm ra chuỗi lặp lại. Mỗi khi bắt gặp sự lặp lại, thuật toán thực hiện đếm số lượng kí tự, bao nhiêu tuỳ ý, trong chuỗi lặp lại và ghi nhận chừng ấy kí tự. Trong thí dụ này, chuỗi lặp lại dài nhất là *the brown fox*. Chuỗi này được thay thế bằng chỉ số chỉ khoảng cách đến chuỗi đứng trước và số liệu về độ dài của chuỗi lặp lại. Trong trường hợp này chuỗi đứng trước của chuỗi lặp lại *the brown fox* có khoảng cách 26 kí tự (kể cả ký tự trống) và độ dài của chuỗi lặp lại có 13 kí tự. Đối với thí dụ

này, sẽ chọn hai phương án để mã hoá: 8bit chỉ khoảng cách và 4bit chỉ độ dài hoặc 8bit chỉ khoảng cách và 6bit chỉ độ dài; 2bit tiêu đề chỉ phương án được lựa chọn: giá trị 00 cho phương án 1, giá trị 01 cho phương án 2. Như vậy, chuỗi lặp lại *the brown fox* được mã hoá thành dạng: $<00_{id}><26_{kc}><13_d>$, hoặc là 00_00011010_1101.

Phần còn lại của tin tức nén được kể từ chữ cái *y*, chuỗi $<00_{id}><27_{kc}><5_d>$ được sử dụng để thay thế cho chuỗi lặp lại *jump*, bao gồm 5 kí tự, kể cả kí tự trống ở phía trước; chuỗi tin tức nén còn lại là *ing frog*,.

Biểu diễn phù hợp của phép nén tin tức trình bày trên hình 2.2. Tin tức sau khi nén bao gồm 35 kí tự 9bit và hai đoạn mã, cho kết quả tổng 35×9 + 2×14 = 343bit. So sánh với 424 bit ban đầu, thì hệ số nén bằng $r = \frac{424}{343} = 1{,}24$.



Hình 2.2. Thí dụ sử dụng sơ đồ LZ77.

7.2. Thuật toán nén

Thuật toán nén theo sơ đồ LZ77 và các phương án của nó sử dụng hai cửa sổ đệm. *Cửa sổ đệm trượt sau* chứa đựng được N kí tự của nguồn, đã gia công trước đó, và *cửa sổ đệm để điều khiển chọn* chứa đựng được L kí tự tiếp theo (L<N), chúng cần được gia công tiếp theo (hình 2.3a).

Thuật toán cố gắng tìm ra chuỗi *hai* hoặc số lượng kí tự *nhiều hơn* từ *cửa sổ đệm để điều khiển chọn* trùng lặp với chuỗi kí tự trong dòng của *cửa sổ đệm trượt sau*. Sự trùng được phát hiện bằng cách đem *chuỗi ký tự* ở cửa sổ đệm để điều khiển chọn *so sánh lần lượt với tất cả các ký tự* còn nằm trong cửa sổ đệm trượt sau. Nếu không tìm thấy sự trùng lặp, thì kí tự cuối cùng thuộc *cửa sổ đệm để điều khiển chọn* được chọn ra như kí tự 9 bit, chính kí tự này được đặt vào *cửa sổ đệm trượt sau*, còn chính kí tự cuối cùng ở cửa sổ này bị đẩy bật ra. Nếu sự trùng lặp được phát hiện, thì trường<*khoảng cách*> được thiết lập, rồi các cửa sổ tiếp tục lướt qua các kí tự tiếp theo để tìm ra chuỗi trùng lặp có độ dài lớn nhất, khi phát hiện thấy kết thúc chuỗi ký tự trùng lặp, thì trường <*độ dài*> và trường <*tiêu đề*> được thiết lập. Sau đó chuỗi trùng lặp được mã hoá thành dạng ba giá trị: <*tiêu đề*>,< *khoảng cách*>,< *độ dài*>. Dòng gồm K kí tự cuối cùng ở *cửa sổ đệm trượt sau* bị đẩy bật ra, còn *dòng mã* được dịch vào cửa sổ này. Trường <*khoảng cách*>, khi được chọn là 8bit, cũng có nghĩa khoảng cách xa nhất mà nó có thể mã hoá là 255 ký tự. Tuy nhiên, khoảng cách thực xa nhất còn phụ thuộc vào chiều dài cửa trượt sau, tức N. Trường <*độ*

dài> được chọn tối ưu bằng *kỳ vọng* của độ dài chuỗi ký tự lặp lại của ngôn ngữ đang xét. Tuy nhiên, điều này khó thực hiện được. Bởi vậy, trường này được chọn theo hai phương án bằng 4 - cho phương án 1, bằng 6 - cho phương án 2. Nếu chọn bằng 4 như thí dụ minh hoạ, cũng có nghĩa *chuỗi ký tự trùng dài nhất* mà nó có thể mã hoá là 15. Nếu trường này được chọn quá dài cũng phát sinh dư thừa, khi mã hoá các chuỗi trùng lặp ngắn. Độ rộng L của cửa sổ điều khiển chọn cũng cần được chọn tối ưu và bằng *kỳ vọng* của độ dài chuỗi ký tự lặp lại của ngôn ngữ đang xét.

Ở' thí dụ minh hoạ khoảng cách ngắn nhất từ chuỗi ký tự đem so sánh đến chuỗi ký tự được so sánh là 26 (chuỗi ký tự thứ nhất), nếu cửa sổ đệm trượt sau được chọn với độ dài ngắn hơn 26, chẳng hạn, N = 25, thì không có chuỗi ký tự trùng lặp nào được phát hiện. Và như vậy, nếu cửa sổ đệm trượt sau được chọn với *độ dài càng dài* thì có thể càng nhiều kiểu trùng lặp được phát hiện, nhưng thời gian mã hoá sẽ lâu hơn.

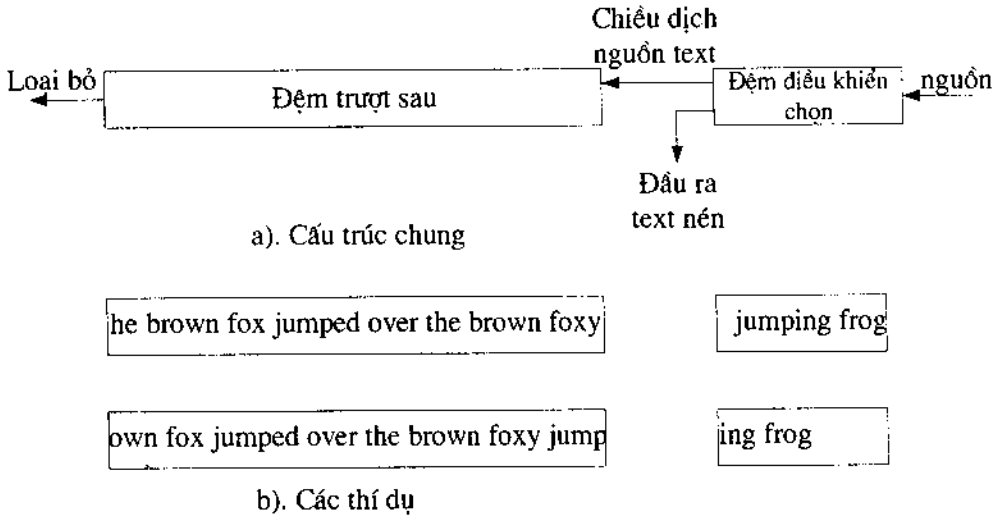
Trên hình 2.3b trình bày hoạt động của sơ đồ này đối với câu văn bản lấy làm thí dụ minh hoạ. Trên hình vẽ biểu diễn *cửa sổ đệm trượt sau* chứa đựng được 39 kí tự (N = 39), còn *cửa sổ đệm để điều khiển chọn* chứa được 13 kí tự (L = 13).

Trong hình 2.3 biểu diễn đã gia công được 40 kí tự đầu tiên, còn lại 39 ký tự cuối trong dạng không nén được nhìn thấy trong cửa sổ trượt sau. Phần còn lại của số liệu nguồn nằm trong *cửa sổ đệm để điều khiển chọn*. Thuật toán nén xác định sự lặp lại của các kí tự tiếp theo, đặt 5 kí tự từ *cửa sổ đệm để điều khiển chọn* vào *cửa sổ trượt sau* và đưa ra mã phù hợp của chuỗi. Trạng thái của các đệm sau hoạt động này được trình bày trong hình vẽ dưới.

Sơ đồ LZ77 là hiệu quả và mô phỏng tự nhiên của số liệu được đưa vào. Nhưng nó cũng có những nhược điểm nhất định. Thuật toán sử dụng cửa sổ đệm trượt sau có kích thước giới hạn để tìm ra sự trùng lặp với văn bản sau đó. Đối với khối văn bản rất dài khi so sánh với kích thước của cửa sổ, thì nhiều kiểu trùng lặp sẽ bị bỏ qua. Kích thước của cửa số có thể tăng, nhưng điều đó dẫn đến các trả giá sau: 1) tăng thời gian thực hiện của thuật toán, bởi vì cần thiết phải thực hiện so sánh các chuỗi ký tự trong *cửa sổ đệm để điều khiển chọn* với các chuỗi ký tự cũ trong *cửa sổ đệm trượt sau*. 2) tăng độ dài của trường <khoảng cách> bởi vì cần thiết để chỉ chuỗi trước đó có độ dài lớn hơn.

Cần xây dựng thuật toán có khả năng tìm ra các đoạn *trùng lặp giống nhau*, dù chúng đứng ở *bất kỳ chỗ nào* trong văn bản (trùng lặp bội), cũng như nhiều dạng trùng lặp khác.

Thuật toán nén nào, càng tìm ra được *nhiều kiểu trùng lặp* (hay nói cách khác là càng *không bỏ sót* một kiểu trùng lặp nào), và tìm ra cách mã hóa chúng bằng *đoạn mã ngắn nhất* có thể, nhưng vẫn đủ thông tin để giải nén, thì thuật toán đó có tỉ số nén càng cao.



Hình 2.3. Sơ đồ LZ77.

Nếu chúng ta để ý đến các thuật toán hàm băm (Hash function) - xem chương 4, thì sẽ thấy chúng là các thuật toán "nén" vô địch. Bởi vì, hàm băm có khả năng biến đổi một *bản tin* có độ dài bất kỳ (nhưng hữu hạn), thành *bản tóm lược* có độ dài chỉ 128, 160, hoặc 256 ký hiệu. Hơn nữa, nói chung, bản tin nào có tương ứng bản tóm lược đó. Bởi vì các phép biến đổi của hàm băm không làm tổn hao thông tin. Tuy nhiên, "giải nén" là không thể. Vì thế, các hàm băm trở thành các thuật toán "nén" không thực tế.

Chúng ta xét một thí dụ khác từ Việt văn. Đoạn văn bản sau đây được trích từ cuốn "Chuyện Đông, chuyện Tây" trang 226 của tác giả An Chi, do NXB Trẻ 2006:

"Trước hết, về ngữ âm, không phải bao giờ yếu tố vay mượn cũng được phát âm y chang như tiếng gốc, vì một lẽ đơn giản là nói chung thì hệ thống ngữ âm của ngôn ngữ đi vay mượn thường khác với hệ thống ngữ âm của ngôn ngữ được vay mượn"

Để đơn giản chúng ta diễn giải quá trình nén của đoạn: "*nói chung thì hệ thống ngữ âm của ngôn ngữ đi vay mượn thường khác với hệ thống ngữ âm của ngôn ngữ được vay mượn*". Đoạn này dài 111 ký tự, chiếm số bít (chưa nén): 888.

Trong đoạn văn bản này có 2 đoạn trùng lặp:

1. _hệ thống ngữ âm của ngôn ngữ đ . Đoạn này dài 31 ký tự, kể cả ký tự trống ở đầu, khoảng cách đến chuỗi ký tự trùng là 56 ký tự. Đoạn mã có dạng: 00_<56>_<31> = 00_00111000_11111.

2. _vay mượn. Đoạn này dài 9 ký tự, kể cả ký tự trống ở đầu, khoảng cách đến chuỗi ký tự trùng là 58 ký tự. Đoạn mã có dạng: 00_<58>_<9> = 00_00111010_01001.

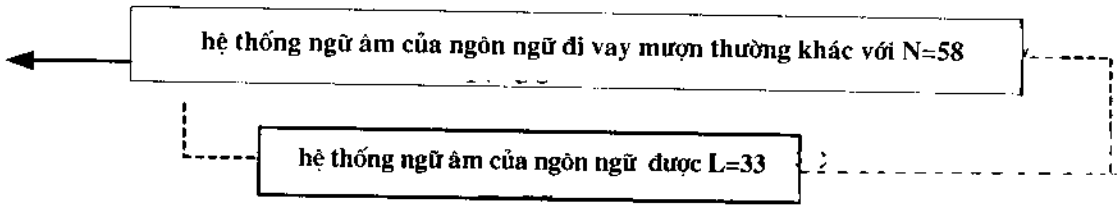
Số ký tự được nén 40, chiếm số bít: 2×15 = 30,

Số ký tự không được nén 111 - 40 = 71, chiếm số bít: 71×9 = 639,

Tổng số bit sau nén: 639 + 30 = 669,

Tỉ số nén là: $r = \frac{888}{669} = 1,327$

Các cửa sổ nén được chọn, chẳng hạn, N = 58, L = 33. Hình ảnh cửa sổ hiện thời khi nén đoạn thứ nhất như hình 2.3.a)



Hình 2.3a). Cửa sổ điều khiển chọn hiện thời phát hiện đoạn ký tự trùng thứ nhất.

7.3. Thuật toán giải nén

Giải nén theo sơ đồ LZ77 thực hiện rất đơn giản. Thuật toán giải nén cần phải lưu giữ N kí tự cuối cùng của đầu ra phục hồi. Khi bắt gặp chuỗi được nén, thuật toán giải nén sử dụng các giá trị của trường <khoảng cách> và <độ dài>, để thay thế các mã này bởi dòng văn bản thực.

8. MÃ HOÁ (NÉN) THEO CHIỀU DÀI LẶP LẠI

Mã hoá (nén) theo chiều dài lặp lại (Run Length Code - RLE) có thể đạt hiệu quả cao khi nén các sơ đồ, bức ảnh tĩnh hai màu (đen, trắng), mà chúng được coi là tập hợp của các đường thẳng (các dòng), trong đó có các đoạn cùng màu. Mã hoá (nén) theo chiều dài lặp lại được coi là một cơ sở của thuật toán nén ảnh tĩnh JPEG.

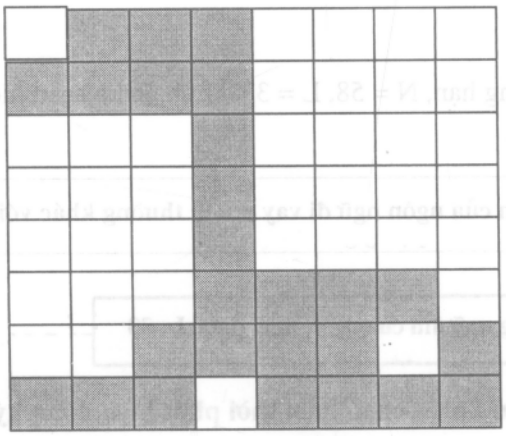
Ý tưởng của phương pháp mã hoá (nén) theo chiều dài lặp lại như sau: thay thế cho việc mã hoá số liệu đặc biệt, bằng việc mã hoá các số phù hợp với đoạn độ dài, mà ở đoạn đó số liệu giữ nguyên không thay đổi.

Giả sử chúng ta cần nén một mảnh ảnh 2 màu (đen, trắng) có kích thước (8×8) phần tử, như hình 2.4.

Quét mảnh ảnh này theo dòng và mã hoá mỗi màu bằng một con số hệ 2 phù hợp là 0 và 1. Kết quả nhận được véctơ số liệu hệ 2 như sau:

X = (01110000_11110000_00010000_0001000_0001000_00011110_00111101_11101111).

Ở đây, để tiện theo dõi, véctơ X được phân thành 8 nhóm, tương ứng với 8 dòng. Véctơ X dài 64bit. Chia véctơ X thành các đoạn, mà ở đó số liệu không đổi, xác định độ dài của chúng bằng các số hệ 10. Kết quả nhận được véctơ mới r = (1, 3, 4, 4, 7, 1, 7, 1, 7, 1, 7, 4, 3, 4, 1, 4, 1, 4). Xác định tần suất của các số trong véctơ này được p(4) = 6/18; p(1) = 6/18; p(7) = 4/18; p(3) = 2/18. Đem mã hoá véctơ r theo phương pháp mã không nhớ ở mục 3. Kết quả mã hoá được trình bày trong bảng 13.



Hình 2.4. Mảnh ảnh đen trắng (8×8) phần tử.

Để chỉ ra chuỗi số liệu đem mã hoá bắt đầu từ 0 (màu trắng), trong chuỗi số liệu nén ta thêm tiếp đầu ngữ 0. Kết quả ta được:

B(r) = (0101110011010110101101001110100100), dài 37 bit.

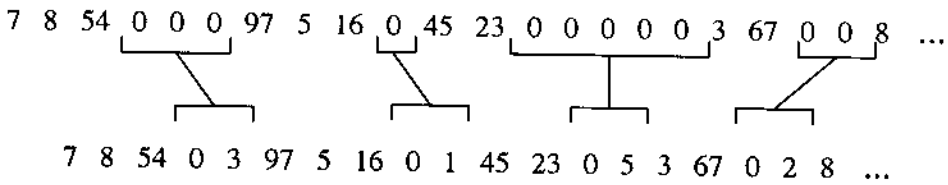
Tỉ số nén trong trường hợp này là: $r = \frac{64}{37} = 1{,}73$. Tuy nhiên, phía giải nén phải hoàn toàn biết kích thước tấm ảnh (bao nhiêu phần tử).

Bảng 13. Bảng mã hoá

Độ dài đoạn	Tần suất	Từ mã
4	6/18	0
1	6/18	10
7	4/18	110
3	2/18	111

Khi nén ảnh đen-trắng với kích thước lớn, phương pháp này có thể đạt hiệu quả nén cao.

Dưới đây trình bày một thí dụ khác của việc sử dụng mã hoá theo chiều dài lặp lại, khi mà trong chuỗi số liệu số bắt gặp các đoạn với một số lượng lớn các giá trị 0. Khi bắt gặp đoạn số liệu 0, dài bất kỳ, nó sẽ được mã hoá bằng 2 số. Số đầu là 0, đóng vai trò là cờ để chỉ bắt đầu mã hoá độ dài dòng số 0, còn số thứ 2- chỉ số lượng số 0. Nếu về trung bình số lượng số 0 trong đoạn lớn hơn 2, thì thuật toán này cho hiệu quả nén; còn ngược lại, nếu nhiều đoạn chỉ có số 0 đơn, thì có thể dẫn đến làm tăng kích thước của dòng số liệu.



Còn một phương pháp khác để nén tín hiệu ảnh, tín hiệu âm thanh bằng phương pháp nén không tiêu hao, đó là phương pháp mã hoá vi sai.

9. CÁC MÃ VI SAI

Nguyên tắc làm việc của các mã vi sai dựa trên một sự kiện là đối với nhiều loại số liệu, sự khác nhau giữa các mẫu lân cận là không nhiều. Chẳng hạn, không nên chờ đợi sự khác nhau nhiều giữa các pixel lân cận của tấm ảnh số.

Sau đây, chúng ta xem xét một thí dụ đơn giản, để chứng tỏ ưu điểm của phương pháp mã vi sai (bằng cách mã hoá hiệu giữa các mẫu) khi so sánh với mã không nhớ đơn giản (giá trị các mẫu không phụ thuộc lẫn nhau).

Bằng biểu diễn 8bit (256 mức) của ảnh số, thu được ở 10 pixel liên tiếp, có các mức biểu diễn trong hệ thập phân, như sau: 144, 147, 150, 146, 141, 142, 138, 143, 145, 142.

Nếu mã hoá các mức của các pixel này bằng mã đều, với mỗi mức 8bit, ta thu được chuỗi số liệu 80 bit.

Bây giờ chúng ta tính hiệu giá trị giữa các mẫu, bắt đầu từ mẫu thứ 2, ta thu được chuỗi số liệu mới tương ứng như sau:

144,	147,	150,	146,	141,	142,	138,	143,	145,	142.
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
144,	3,	3,	- 4,	- 5,	1,	- 4,	5,	2,	-3.

Chuỗi số liệu ban đầu dễ dàng được khôi phục từ chuỗi số liệu hiệu, bằng cách tính tổng:

144,	144+3,	147+3,	150- 4,	146 - 5,	141+1,	142 - 4,	138+5,	143+2,	145-3
⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓	⇓
144,	147,	150,	146,	141,	142,	138,	143,	145,	142.

Để mã hoá chuỗi số liệu hiệu, chúng ta mã hoá như sau: số thứ nhất mã hoá bằng 8 bit, các số sau mã hoá bằng 4bit, trong đó bit đầu chỉ dấu (+ = 1, – = 0), còn 3bit sau để mã hoá giá trị (vì giá trị các hiệu ≤ 7).

Kết quả mã hoá, cho chúng ta chuỗi số liệu có 8 + 9×4 = 44bit, gần như ngắn bằng một nửa, điều này không thể có khi mã hoá từng mẫu riêng biệt. Hệ số nén trong trường hợp cụ thể này là r = 80/ 44 ≈ 2.

Phương pháp mã vi sai được sử dụng rộng rãi trong các trường hợp, khi mà chuỗi số liệu, một cách tự nhiên, có các giá trị lân cận nhau không khác nhau đáng kể, còn chính các giá trị ấy có thể lớn bao nhiêu tuỳ ý. Thuộc loại các chuỗi số liệu như thế gồm có tín hiệu âm thanh, đặc biệt tín hiệu tiếng nói sau khi số hoá; đối với tín hiệu hình ảnh thì các pixel lân cận nhau thông thường có độ sáng và màu sắc gần giống nhau. Trong khi đó, phương pháp này không áp dụng được cho các dạng số liệu từ mã hoá văn bản, hình vẽ, hoặc bất kỳ số liệu nào, mà các giá trị lân cận nhau không phụ thuộc lẫn nhau.

10. CÁC PHƯƠNG PHÁP NÉN CÓ TIÊU HAO THÔNG TIN

Như đã nhận xét trước đây, trong thực tiễn tồn tại hai hệ thống nén số liệu: hệ thống nén số liệu không tiêu hao thông tin và hệ thống nén số liệu có tiêu hao thông tin.

Khi nén số liệu không tiêu hao thông tin, thì việc khôi phục tin tức đạt được độ chính xác tuyệt đối. Tuy nhiên, hiệu quả nén không cao, hệ số nén trong các trường hợp này ít khi vượt 3 đến 5 (ngoại trừ trường hợp khi mà tần suất lặp lại của các đoạn số liệu rất lớn).

Đối với các nguồn tin tức dạng tương tự (như âm thanh, hình ảnh), các phép biến đổi chúng không bao giờ đạt được độ chính xác tuyệt đối. Chẳng hạn, thứ nhất phép biến đổi tin tức từ nguồn nguyên thuỷ thành tín hiệu điện, do dải động, dải thông tần của *các thiết bị định dạng* đều bị hạn chế, nên phép biến đổi này không hoàn chỉnh. Thứ hai, tiếp đến *phép mã hoá nén*, trong đó có quá trình lấy mẫu, chính quá trình này đã đưa entropy của nguồn từ vô cùng về giá trị hữu hạn, quá trình lượng tử hoá bản chất là quá trình làm xấp xỉ. Vì vậy, phép mã hoá nén là phép biến đổi có tiêu hao thông tin. Thứ ba, quá trình truyền tin trên kênh luôn luôn tồn tại *các loại nhiễu khác nhau*, chúng làm mất mát một phần thông tin của tin tức truyền. Thứ tư, *các thiết bị thu* và *các thiết khôi phục* đều là các thiết bị không bao giờ hoàn chỉnh. Cuối cùng, sự cảm nhận của *tai*, của *mắt* con người cũng bị hạn chế, luôn luôn có ngưỡng về âm lượng, âm sắc, về màu sắc. Vì tất cả các lý do như vậy, chúng ta không mong đợi các phép biến đổi là hoàn chỉnh và việc khôi phục tin tức dạng tương tự là chính xác tuyệt đối.

Hệ thống nén có tiêu hao thông tin cho hệ số nén đôi khi lớn gấp hàng chục lần so với hệ thống nén không tiêu hao.

Phần lớn các phương pháp nén số liệu có tiêu hao đều dựa trên việc mã hoá không phải chính số liệu, mà là số liệu từ phép biến đổi tuyến tính nào đó của chúng. Chẳng hạn, các hệ số của phép biến đổi Fourier rời rạc (Discrete Fourier Transformation = DFT), các hệ số của phép biến đổi cosine rời rạc (Discrete Cosine Transformation = DCT), biến đổi Khapa, Yolsa ...

Để hiểu được vì sao trong kỹ thuật nén có tiêu hao thông tin lại sử dụng phép biến đổi không phải là chính số liệu, mà là phép biến đổi các hệ số của phép biến đổi tuyến tính nào đó của số liệu và tính hiệu quả cao của nó, chúng ta nghiên cứu chuẩn nén ảnh đen trắng JPEG (Joint Photographers Experts Group).

11. THUẬT TOÁN NÉN ẢNH TĨNH JPEG

Thuật toán phổ thông nén ảnh tĩnh JPEG có thể diễn tả đầy đủ nguyên lý nén có tiêu hao thông tin trên cơ sở của mã hoá biến đổi.

Ý tưởng cơ bản của mã hoá biến đổi có thể diễn giải đơn giản như sau: giả thử chúng ta cần xử lý một tín hiệu số, được biểu diễn bằng các mẫu của chúng. Nếu bỏ đi ở mỗi mẫu một nửa loạt nhị phân (chẳng hạn, 4 loạt từ mẫu 8 loạt), thì dung lượng nén giảm đi một nửa, nhưng cũng mất đi một nửa lượng thông tin.

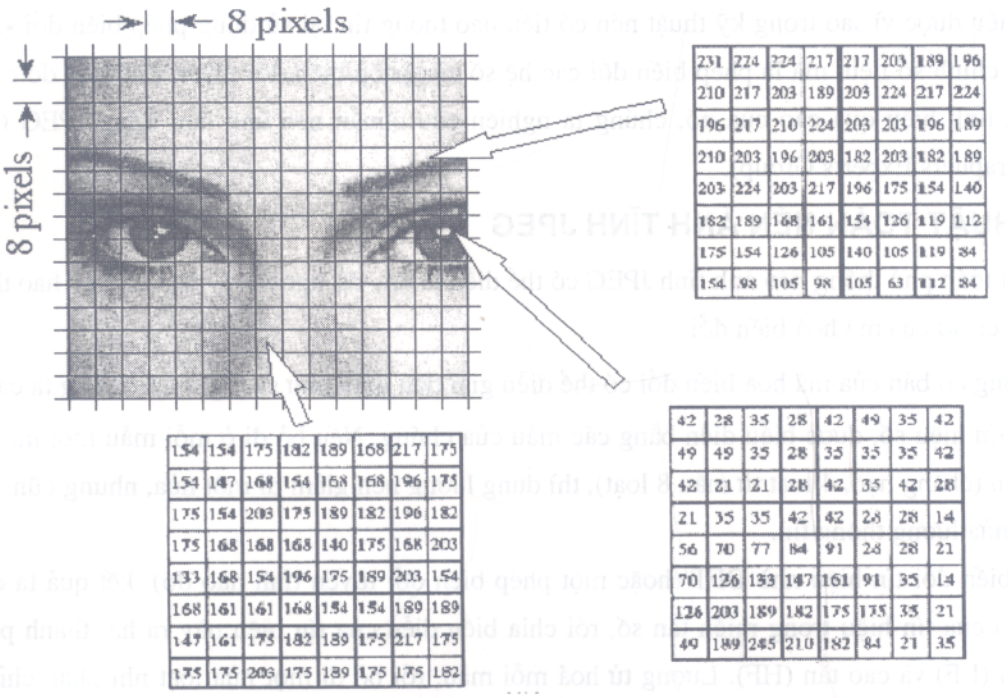
Nếu biến đổi tín hiệu nhờ DFT (hoặc một phép biến đổi tuyến tính nào đó), kết quả ta được các mẫu của tín hiệu trong miền tần số, rồi chia biểu diễn của tín hiệu này ra hai thành phần-thấp tần (LF) và cao tần (HF). Lượng tử hoá mỗi mẫu, rồi bỏ đi một nửa loạt nhị phân chỉ của các mẫu HF. Kết quả dung lượng nén giảm đi một phần ba, còn lượng thông tin bị mất chỉ 5%. Điều này được giải thích như sau: các thành phần thấp tần của phần lớn các tín hiệu đều chứa phần lớn năng lượng và mang số lượng lớn thông tin của chúng, còn ngược lại, các thành phần HF chỉ chứa một phần nhỏ năng lượng và mang một phần nhỏ thông tin của chúng (Xem chương 5). Thuộc loại các tín hiệu như thế này là tín hiệu âm thanh, tín hiệu hình ảnh.

Chúng ta nghiên cứu thuật toán nén JPEG với ảnh đen-trắng, mỗi phần tử (pixel) của nó có mức phân giải độ sáng là 256 mức, tức là mỗi phần tử biểu diễn bằng loạt 8bit nhị phân, hay 1byte. Độ sáng cực đại của ảnh là 255, tương đương với màu trắng, độ sáng cực tiểu của ảnh là 0, tương đương với màu đen. Các màu trung gian, chiếm giá trị từ 1 đến 254, như biểu diễn trên hình 2.5.

11.1.Bước 1. Chia tấm ảnh thành các ô vuông có kích thước 8×8

Làm việc của thuật toán nén JPEG được bắt đầu với việc chia tấm ảnh thành các ô vuông có kích thước 8×8 = 64 pixel (mỗi pixel là một hình vuông, có cạnh bằng độ rộng của dòng quét). Chúng ta chọn như vậy, vì nếu chọn kích thước ô vuông nhỏ thì hiệu quả nén sẽ không lớn (nếu kích thước ô là 1×1, không có hiệu quả nén); còn nếu chọn kích thước ô quá lớn, thì trong mỗi ô màu sắc thay đổi quá lớn, hiệu quả nén cũng kém.

Trên hình 2.5. biểu diễn các giá trị độ sáng của 3 ô ở các vị trí khác nhau của tấm ảnh, độ sáng mỗi pixel được biểu diễn bằng các số trong hệ 10. Mỗi ô là một ma trận gồm 8×8 = 64 phần tử. Sau này, mỗi ô sẽ được gia công và mã hoá không phụ thuộc lẫn nhau.



Hình 2.5. Phân tấm ảnh thành các ô vuông (8 × 8) pixel, được biểu diễn bằng các số hệ 10.

11.2. Bước 2. Biến đổi cosine rời rạc

Bước thứ 2 của thuật toán nén là áp dụng với mỗi ô phép biến đổi DCT. Để thực nghiệm nén ảnh, người ta đã thử nghiệm với nhiều phép biến đổi khác nhau, chẳng hạn, biến đổi Karhunen-Loéve, cho hệ số nén cao, nhưng rất khó thực hiện trong thực tế; biến đổi DFT thực hiện đơn giản, nhưng hiệu quả nén không cao. Cuối cùng chọn biến đổi cosine rời rạc- DCT. Khác với các biến đổi khác, DCT chỉ sử dụng hàm thực, với mỗi thành phần cosine. Cũng như mọi phép biến đổi khác, DCT cho phép biến đổi biểu diễn điểm ảnh theo không gian 2 chiều về cường độ sáng, ký hiệu x(i, j), thành biểu diễn điểm ảnh theo không gian 2 chiều về phổ tần số, ký hiệu X(u, v) và ngược lại.

DCT thuận được biểu diễn bằng thuật toán sau:

$$X(u,v) = \frac{2}{N}.C(u).C(v).\sum_{i=1}^{N-1}\sum_{j=1}^{N-1} x(i,j) \times \cos(\frac{(2i+1)\pi.u}{2N}).\cos(\frac{(2j+1)\pi.v}{2N}) \qquad (2.4)$$

ở đây N = 8, còn hệ số C(u), C(v) = $\frac{1}{\sqrt{2}}$ nếu u, v ≡ 0, ngoài ra lấy bằng 1.

Như vậy, kết quả áp dụng DCT đối với ô ảnh kích thước (8×8) pixel, nhận được phổ tần số hai chiều cũng với kích thước (8× 8) mẫu. Nói một cách khác, từ 64 số biểu diễn các mẫu độ sáng của pixel, biến đổi thành 64 số biểu diễn các mẫu của phổ tần số DCT.

Đối với DCT kích cỡ (8× 8), người ta gọi các thành phần hàm cosine là hàm cơ sở được xác định theo công thức:

$$B(i,j) = \cos\left[\frac{(2i+1)\pi.u}{16}\right] \times \cos\left[\frac{(2j+1)\pi v}{16}\right] \tag{2.5}$$

trong công thức này cặp (u,v) đóng vai trò tham biến, với $0 \le (u, v) \le 7$; cặp biến số (i, j) lấy các giá trị nguyên trong khoảng $0 \le (i, j) \le 7$. Ứng với mỗi giá trị của cặp (u,v), hàm cơ sở B(i, j) xác định được tất cả (8×8) = 64 giá trị. Chẳng hạn, với cặp (u,v) = (6,4), $B(i,j) = \cos(\frac{(2i+1)3\pi}{8}) \times \cos(\frac{(2j+1)\pi}{4})$. $B(i,j)$ tạo thành ma trận [B] và nó có dạng:

$$[B] = \begin{bmatrix} ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \\ ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \end{bmatrix}$$

Trên hình 2.6 trình diễn hình ảnh 2 chiều của 6 hàm cơ sở ứng với 6 cặp giá trị (u,v) khác nhau. Chẳng hạn, hình ảnh ở giữa bên phải, ứng với ma trận [B] vừa xác định ở trên.

Thành phần *một chiều* phù hợp với toạ độ (0,0) được biểu diễn ở đỉnh trái trên cùng của hình vẽ, còn hàm cơ sở *tần số cao nhất* phù hợp với toạ độ (7,7) được biểu diễn ở đỉnh phải dưới cùng. Hàm cơ sở đối với toạ độ (0,1) là hàm cosine theo toạ độ j và hằng số theo toạ độ i; còn hàm cơ sở với toạ độ (1,0) cũng như trên, nhưng quay đi 90°, tức là hàm cosine theo toạ độ i và hằng số theo toạ độ j.

Kết quả, chẳng hạn, phổ DCT với chỉ số (u,v) = (0, 0) đơn giản là trung bình của tổng tất cả các phần tử của ô ảnh, và là thành phần *một chiều* . Thành phần với chỉ số (1, 0) biểu diễn trung bình độ sáng của tất cả các ô ảnh theo chiều ngang, còn theo chiều đứng trọng số lớn nhất được trao cho phần tử (0, 1) v.v.. Có thể nhận thấy rằng khi (u,v) càng lớn (về phía dưới, phải) của phổ DCT, càng biểu diễn thành phần cao tần của phổ.

Bảng 14a): Giá trị cường độ chói Y trong biểu diễn hệ số 10 từ 0 đến 255

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Giá trị cường độ chói Y biểu diễn ban đầu, chẳng hạn, trong dạng hệ số 10 từ 0 đến 255 (8bit cho một pixel), như bảng 14a), cần chuyển sang biểu diễn từ -128 đến +127, để phù hợp với biên độ của các thành phần màu, mà chúng ta xét sau này (xem mục 11.8). Như vậy, mỗi phần tử của bảng 14a) cần trừ đi 128, kết quả cho bảng 14b).

Bảng 14b): Giá trị cường độ chói Y trong biểu diễn hệ số 10 từ -128÷127

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Bảng 14c). Kết quả DCT - gọi là phổ DCT

$$\begin{bmatrix} -415 & -30 & -61 & 27 & 56 & -20 & -2 & 0 \\ 4 & -22 & -61 & 10 & 13 & -7 & -9 & 5 \\ -47 & 7 & 77 & -25 & -29 & 10 & 5 & -6 \\ -49 & 12 & 34 & -15 & -10 & 6 & 2 & 2 \\ 12 & -7 & -13 & -4 & -2 & 2 & -3 & 3 \\ -8 & 3 & 2 & -6 & -2 & 1 & 4 & 2 \\ -1 & 0 & 0 & -2 & -1 & -3 & 4 & -1 \\ 0 & 0 & -1 & -4 & -1 & 0 & 1 & 2 \end{bmatrix}$$

Chúng ta có một nhận xét quan trọng là phổ DCT nhận được có các giá trị lớn nhất (tương ứng với thành phần tần số thấp) thuộc đỉnh bên trái của bảng 14c). Chúng chứa phần lớn năng lượng của tín hiệu. Hơn nữa, thành phần của phổ DCT ứng với tọa độ (u, v) = (0, 0), có giá trị lớn nhất trong bảng, là thành phần một chiều (DC-Direct Current). Các phép biến đổi sau này, cần phải bảo đảm sự chính xác cao đối với thành phần một chiều. Bất kỳ sự sai sót nào trong quá trình biến đổi đối với thành phần một chiều đều làm giảm chất lượng ảnh khôi phục. Vì thành phần

một chiều phản ánh cường độ sáng trung bình của ô ảnh. Ngoài thành phần một chiều, các thành phần khác gọi là thành phần xoay chiều (AC- Alternating Current) của phổ DCT. Các thành phần thuộc đỉnh phải dưới cùng (tương ứng với thành phần tần số cao nhất) có giá trị tương đối nhỏ.

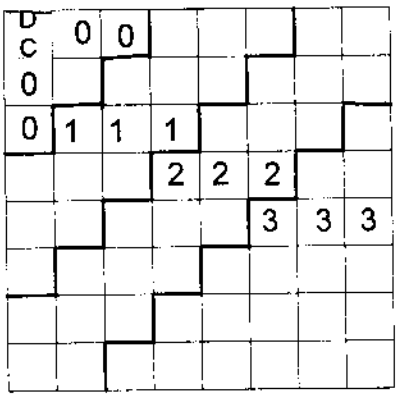
11.3. Bước 3. Lượng tử hoá

Bước tiếp theo của thuật toán JPEG là lượng tử hoá. Từ bảng 14c) là kết quả nhận được sau DCT, chúng ta nhận thấy quá nửa bảng, phần phổ cao tần, có giá trị 0 hoặc có giá trị rất nhỏ chỉ 1 hoặc 2. Các phổ cao tần này, trong quá trình xử lý tiếp theo có thể bỏ đi, hoặc ít nhất được làm tròn đến giá trị nguyên nhỏ.

Bước lượng tử thực hiện chia các hệ số của phổ DCT cho các số trong bảng các mức lượng tử, bảng 15a), phù hợp theo:

$$X_q(u,v) = \text{giá trị nguyên của làm tròn}\left(\frac{X(u,v)}{q(u,v)}\right)$$

trong đó: X(u,v) - giá trị phổ DCT cho ở bảng 14c). $X_q(u,v)$ - kết quả lượng tử hoá, như bảng 15b). *q(u,v)*- số mức lượng tử hoá, một thí dụ trình bày trong bảng 15a).



Hình 2.7 . Phân vùng lượng tử.

Nếu các số trong bảng 15a) càng lớn, nghĩa là số mức lượng tử lớn, thì kết quả của phép chia, sau khi đã làm tròn, càng có nhiều giá trị 0, cũng tức là nén càng mạnh và mức tiêu hao càng thấy rõ. Hơn nữa, bảng 15a) là bảng mức lượng tử của cường độ chói Y. Chúng được xác định *từ thực nghiệm*, trên cơ sở phân vùng lượng tử, như hình 2.7). Có bốn vùng lượng tử, đánh số từ 0 đến 3. Theo thứ tự vùng, độ chính xác lượng tử giảm dần. Người ta sử dụng QN0x để ký hiệu cho các bảng khác nhau mức lượng tử. Theo giá trị tăng dần của x (x = 1, 2, 3,...), giá trị mức lượng tử trong các vùng khác nhau tăng dần, so với các giá trị mức lượng tử đầu tiên, cũng có nghĩa là các hệ số DCT giảm dần. Chẳng hạn, QN04 có cấu trúc 0: 2: 2: 4, nghĩa là hệ số DCT ở vùng 0 giữ nguyên, còn vùng 2 và 3 chia cho 2, còn vùng 3 chia cho 4. Phía nén sử dụng QN0x nào, cần truyền cho phía giải nén biết, để sử dụng thích hợp.

Rõ ràng, việc lựa chọn giá trị các số trong bảng mức lượng tử sẽ phụ thuộc đáng kể vào hệ số nén- số lượng các giá trị 0 (sau làm tròn) trong phổ lượng tử hoá, cũng như chất lượng bức ảnh sau khôi phục. Đây là bước duy nhất của thuật toán làm tiêu hao thông tin.

Bảng 15a): Thí dụ bảng mức lượng tử hóa cường độ chói Y

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

Như vậy, kết quả làm tròn phổ DCT sau lượng tử hoá sẽ nhận mức độ méo ít hoặc nhiều phổ DCT, cũng như bức ảnh sau khôi phục.

Bảng 15b): Kết quả lượng tử phổ DCT

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

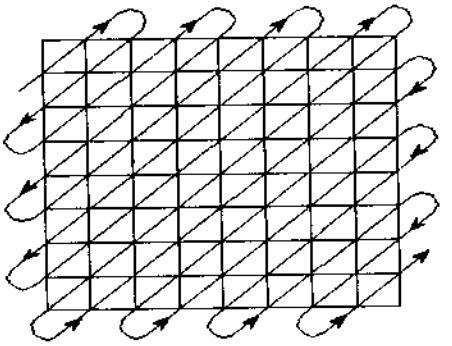
Chẳng hạn: -26 = -415/16 = -25,9 → làm tròn bằng -26, gọi là hệ số một chiều (DC), các số hạng còn lại gọi là các hệ số xoay chiều (AC)

11.4. Bước 4. Quét zích-zắc

Bước tiếp theo của thuật toán JPEG là biến đổi ma trận 8×8 của phổ DCT sau lượng tử hoá thành chuỗi nối tiếp. Nhưng phải làm sao cho có thể nhóm được các số có giá trị lớn thành một nhóm, tiếp theo là nhóm của các số 0 ở phổ DCT sau lượng tử hoá. Thủ tục này gọi là *quét zích zắc* như hình 2.8.

Kết quả của quá trình *quét zích zắc* này sẽ biến đổi ma trận 8 × 8 của phổ DCT sau lượng tử hoá thành chuỗi nối tiếp gồm 64 số, trong đó phần lớn là số 0. Với một chuỗi như vậy, có thể nén hiệu quả bằng mã hoá theo *chiều dài lặp lại (RLE).*

Trong thí dụ này, kết quả quét zích zắc, nhận được chuỗi như bảng 15c).



Hình 2.8. Thủ tục *quét zích zắc*.

Bảng 15c). Chuỗi số liệu nhận được sau bước quét zích-zắc.

−26,

−3, 0,

−3, −2, −6,

2, −4, 1, −4,

1, 1, 5, 1, 2,

−1, 1, −1, 2, 0, 0,

0, 0, 0, −1, −1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0,

0, 0, 0, 0,

0, 0, 0,

0, 0,

0

11.5. Bước 5. Mã hóa theo chiều dài lặp lại

Bước thứ 5 của thuật toán JPEG là mã hoá chuỗi nhận được theo mã *chiều dài lặp lại* - RLE, để giảm nhỏ chuỗi số 0. Lúc này, chuỗi số liệu trong bảng 15c) được đơn giản bằng từ mã EOB (End Off Block) để chỉ chuỗi 38 số 0 ở cuối, như bảng 16a). Sau đó, mã RLE thực hiện mã hóa các số liệu trong dạng <chiều dài chuỗi 0, giá trị>, gọi là *một từ mã RLE*.

Chẳng hạn, sau hệ số DC bằng -26, tiếp đến hệ số AC bằng -3, vì không có số 0 nào phía trước, nên từ mã RLE có dạng <0,-3>, tiếp đến hệ số AC bằng -3 có một số 0 phía trước, nên từ mã RLE có dạng <1,-3>. Kết quả thu được như bảng 16b).

11.6. Bước 6. Mã hóa entropy

Bước cuối cùng của thuật toán JPEG là thực hiện mã hoá các từ mã RLE nhận được ở bước 5, có dạng <chiều dài chuỗi 0, giá trị>, bằng một mã kinh tế đã biết nào đó, chẳng hạn mã Huffman hoặc mã số học. Chúng ta xét quá trình sử dụng mã Huffman.

Trong bước này, để mã hóa mỗi từ mã RLE ở trên, gồm hai đoạn mã:
<chiều dài chuỗi số 0> và <giá trị>. Để mã hóa <chiều dài chuỗi số 0> cần sử dụng hai bảng tìm kiếm (bảng 17a,b)), để tìm từ mã Huffman tương ứng với <chiều dài chuỗi 0>. Bảng 17b) xây dựng theo nguyên tắc mã tiết kiệm. Để xác định các từ mã Huffman này, người ta phải mô phỏng với vô số các ô ảnh với đặc tính phân bố độ sáng khác nhau, để tìm ra thống kê <chiều dài chuỗi 0> và xác định tần suất của chúng.

Bảng 17a). Phân loại khoảng giá trị các hệ số AC (trích)

Phân loại	Khoảng giá trị	
1	-1	1
2	-3,-2	2, 3
3	-7,-6,-5,-4	4, 5, 6, 7
4	-15,...,-8	8 ,..., 15
5	-31,...,-16	16, ..., 31
6	-63,...,-32	32, ..., 63
7	-127,...,-64	64,...,127
8	-255,...,-128	128,...,255
...	...	...

Các từ mã Huffman cho các hệ số DC khác nhau, cũng được xác định từ bảng tìm kiếm, như bảng 17c).

Các <giá trị> được mã hóa trực tiếp, tức là các giá trị dương hệ số 10 mã hóa theo hệ số 2 trực tiếp. Chẳng hạn, $(1)_{10} \rightarrow (1)_2$, $(2)_{10} \rightarrow (10)_2$, $(3)_{10} \rightarrow (11)_2$,...

Để mã hóa các giá trị âm, người ta thực hiện bằng cách thêm ký hiệu 0 trước chuỗi mã của giá trị. Còn chính 'giá trị' đã bỏ đi dấu âm, được mã hóa như giá trị dương ở trên. Chẳng hạn:

$(-1)_{10} \rightarrow (01)_2$, Tương tự như vậy, $(-2)_{10} \rightarrow (010)_2$, $(-4)_{10} \rightarrow (0100)_2$,...

Bảng 17b). Các từ mã Huffman của các hệ số AC (trích)

Chiều dài chuỗi số 0	Phân loại	Độ dài từ mã	Từ mã
0	1	2	00
0	2	2	01
0	3	3	100
0	4	4	1011
0	5	5	11010
0	6	6	111000
0	7	7	1111000
.	.	.	.
1	1	4	1100
1	2	6	111001
1	3	7	1111001
1	4	9	111110110
.	.	.	.
2	1	5	11011
2	2	8	11111000
.	.	.	.
3	1	6	111011
3	2	9	111110111
.	.	.	.
4	1	6	111010
5	1	7	1111010
6	1	7	1111001
7	1	8	11111001
...	...	...	...

Bây giờ, chúng ta xét thí dụ quá trình mã hóa các từ mã RLE ở bảng 16b).

Trước hết là hệ số DC bằng - 26. Từ bảng 17c), chúng ta xác định được hệ số này được phân loại 5, với từ mã Huffman tương ứng 1110, còn <giá trị> được mã hóa bằng 011010. Kết quả có đoạn mã 1110_011010.

Tiếp đến từ mã RLE < 0,-3>. Từ bảng 17a), chúng ta xác định được hệ số này được phân loại 2, chiều dài chuỗi số 0 đứng trước là 0, cho nên từ bảng 17b) xác định được từ mã Huffman tương ứng là 01, còn <giá trị> được mã hóa bằng 011. Kết quả có đoạn mã 01_011.

Bảng 17c). Các từ mã Huffman cho các hệ số DC

Khoảng giá trị các hệ số DC	Phân loại	Từ mã cho tín hiệu Y
0	0	100
-1 1	1	00
-3,-2 2, 3	2	01
-4,-7 4, 7	3	101
-8,-15 8, 15	4	110
-16,-31 16, 31	5	1110
-32,-63 32, 63	6	11110
-64,-127 64, 127	7	111110
-128,-255 128, 255	8	1111110

Tiếp đến, chẳng hạn, từ mã RLE < 1,-3 >. Từ bảng 17a), chúng ta xác định được hệ số này được phân loại 2, chiều dài chuỗi số 0 đứng trước là 1, cho nên từ bảng 17b) xác định được từ mã Huffman tương ứng là 111001, còn <giá trị> được mã hóa bằng 011. Kết quả có đoạn mã 111001_011.

Tiếp đến, chẳng hạn, từ mã RLE < 5,-1 >. Từ bảng 17a), chúng ta xác định được hệ số này được phân loại 1, chiều dài chuỗi số 0 đứng trước là 5, cho nên từ bảng 17b) xác định được từ mã Huffman tương ứng 1111010, còn <giá trị> được mã hóa bằng 0. Kết quả có đoạn mã 1111010_0.

Bảng 16): Đơn giản khối số liệu bằng từ mã EOB a), kết quả mã hóa theo RLE b) và kết quả mã hóa theo Huffman c).

a)
-26,
-3, 0,
-3, -2, -6,
2, -4, 1, -4,
1, 1, 5, 1, 2,
-1, 1, -1, 2, 0, 0,
0, 0, 0, -1, -1, EOB

b) Các từ mã RLE

<-26> <0,-3> <1,-3> <0,-2> <0,-6> <0,2> <0,-4>

<0,1> <0,-4> <0,1> <0,1> <0,5> <0,1> <0,2>

<0,-1> <0,1> <0,-1> <0,2> <5,-1> <0,-1> EOB

c) Chuỗi mã nhận được sau mã hóa Huffman

1110_011010, 01_011, 111001_011 , 01_01 ,100 _0110, 01_1 , 100_010

00_1 , 100_010 , 00_1 , 00_1 , 100_101 , 00_1 , 01_10 , 00_0 , 00_1 ,00_0 ,

01_10 , 1111010_0 ,00_0 , 1010

Từ mã EOB được mã hóa bằng từ mã ngắn nhất 1010.

Kết quả nhận được chuỗi số liệu, như bảng 16c), có kích thước giảm hơn nhiều so với kích thước chuỗi số liệu ban đầu. Trong thí dụ này, chuỗi số liệu sau nén dài 100bit, còn chuỗi số liệu trước nén là: 8×8×8 = 512bit. Như vậy, hệ số nén r = 512/100 = 5,12.

Chúng ta cũng có thể hỏi vì sao phải xây dựng các mã phức tạp như vậy? Câu trả lời là để *chuỗi mã nhận được ngắn nhất và các từ mã RLE có tính Prefix, bảo đảm cho khả năng giải mã có tính phân tách và đơn trị.*

Hai bước cuối cùng của thuật toán JPEG thường được gọi bước nén thứ hai, ở đây thực hiện bằng mã kinh tế không gây tiêu hao thông tin, với sự tính toán về đặc trưng cấu trúc số liệu để thực hiện giảm dung lượng của nó. Các bảng tra cứu- bảng 17, cần lưu giữ ở cả hai phía- phía nén và phía giải nén.

11.7. Giải nén theo thuật toán JPEG diễn ra theo trình tự ngược, với các thuật toán ngược tương ứng.

Bảng 18a): Khôi phục kết quả lượng tử hóa phổ DCT

$$\begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Sau khi mở số liệu nén lần 2 theo thuật toán giải mã của mã thống kê tương ứng (chẳng hạn, giải mã theo Huffman, RLE), rồi xếp đặt chuỗi số liệu nối tiếp vào khối kích thước 8×8 theo thủ tục quét zích zắc, như bảng18a.

Giải lượng tử hoá các số này (bảng18a), nhờ lưu giữ bảng mức lượng tử hoá, bằng cách nhân tương ứng với các số của bảng. Kết quả nhận được bảng 64 số liệu sau giải lượng tử hoá, như bảng18b).

Thí dụ: - 416 = - 26 × 16.

Thực hiện biến đổi cosine rời rạc ngược (DCT^{-1}) với các số liệu trong bảng 18b), thu được các số liệu gần với ban đầu nhất - bảng 18c).

Bảng 18b): Giải lượng tử hóa

$$\begin{bmatrix} -416 & -33 & -60 & 32 & 48 & -40 & 0 & 0 \\ 0 & -24 & -56 & 19 & 26 & 0 & 0 & 0 \\ -42 & 13 & 80 & -24 & -40 & 0 & 0 & 0 \\ -56 & 17 & 44 & -29 & 0 & 0 & 0 & 0 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Bảng 18c): Kết quả DCT ngược

$$\begin{bmatrix} -68 & -65 & -73 & -70 & -58 & -67 & -70 & -48 \\ -70 & -72 & -72 & -45 & -20 & -40 & -65 & -57 \\ -68 & -76 & -66 & -15 & 22 & -12 & -58 & -61 \\ -62 & -72 & -60 & -6 & 28 & -12 & -59 & -56 \\ -59 & -66 & -63 & -28 & -8 & -42 & -69 & -52 \\ -60 & -60 & -67 & -60 & -50 & -68 & -75 & -50 \\ -54 & -46 & -61 & -74 & -65 & -64 & -63 & -45 \\ -45 & -32 & -51 & -72 & -58 & -45 & -45 & -39 \end{bmatrix}$$

Đem các phần tử trong bảng 18c), cộng với 128, được kết quả như bảng 18d). Đem so sánh kết quả giải nén của thuật toán JPEG với các số liệu ban đầu trước khi nén (bảng 14a)), để chúng ta thấy được mức độ méo ít nhiều.

Chẳng hạn, với tọa độ (i,j) = (0,0) từ bảng 14a) có $Y_{(0,0)} = 52$, còn từ bảng 18d có $Y_{(0,0)}$ sau khôi phục bằng 60. Như vậy, lệch giữa hai giá trị là 8. Tương tự cho các giá trị khác.

Bảng 18d): Kết quả khôi phục số liệu, sau giải nén

$$\begin{bmatrix} 60 & 63 & 55 & 58 & 70 & 61 & 58 & 80 \\ 58 & 56 & 56 & 83 & 108 & 88 & 63 & 71 \\ 60 & 52 & 62 & 113 & 150 & 116 & 70 & 67 \\ 66 & 56 & 68 & 122 & 156 & 116 & 69 & 72 \\ 69 & 62 & 65 & 100 & 120 & 86 & 59 & 76 \\ 68 & 68 & 61 & 68 & 78 & 60 & 53 & 78 \\ 74 & 82 & 67 & 54 & 63 & 64 & 65 & 83 \\ 83 & 96 & 77 & 56 & 70 & 83 & 83 & 89 \end{bmatrix}$$

Rõ ràng các số liệu khôi phục không khác nhiều so với các số liệu ban đầu trước khi nén. Thuật toán JPEG là thuật toán nén có tiêu hao. Tuỳ theo hệ số nén khác nhau mà mức độ tiêu hao khác nhau.

Trên hình 2.9. trình bày theo trình tự bức ảnh ban đầu (bên trái), bức ảnh được nén với hệ số nén bằng 10 (ở giữa) và bức ảnh được nén với tỉ số r = 45 (phải cùng).

Như vậy, các bước của thuật toán nén JPEG như sau:

1.Chia bức ảnh thành các ô có kích thước (8×8) pixel.

2. Với mỗi ô sử dụng DCT.

3. Làm tròn các số của phổ DCT theo bảng mức lượng tử hoá cho trước.

4. Biến đổi ma trận của các hệ số được làm tròn thành chuỗi số liệu nối tiếp (quét zích zắc).

5. Mã hoá chuỗi số liệu nối tiếp bằng mã độ dài lặp lại, để giảm nhỏ chuỗi số 0, thu được các từ mã RLE.

6. Mã hoá chuỗi số liệu nhận được ở bước 5 bằng mã kinh tế Huffman, nhờ các bảng tìm kiếm.

Hình 2.9. So sánh các bức ảnh

Giải mã cùng từng ấy bước, nhưng theo trình tự ngược với các toán tử ngược.

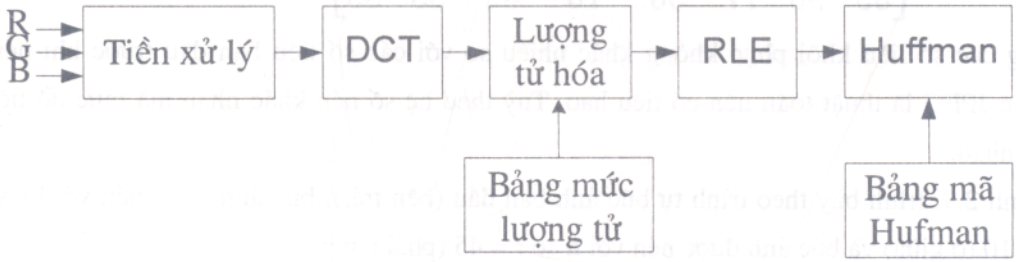
Thuật toán nén JPEG có các ưu điểm sau:

- Có khả năng nén trong dải rộng (từ 2 đến 200),
- Có thể làm việc với ảnh có độ phân giải bất kỳ.
- Thuật toán nén và giải nén đối xứng nhau.

Tuy nhiên, thuật toán nén JPEG cũng có các nhược điểm: có thể xuất hiện các quầng sáng ở các vị trí có thay đổi màu mạnh- gọi là hiệu ứng Gibbs, cũng có thể bị phân rã ảnh thành các ô (8×8) khi hệ số nén cao.

11.8. Các bước thuật toán nén ảnh màu JPEG

Thuật toán nén ảnh màu JPEG gồm 6 bước như hình 2.10a).



Hình 2.10 a). Các bước nén ảnh màu của thuật toán JPEG

1.Bước tiền xử lý: Gồm hai quá trình:

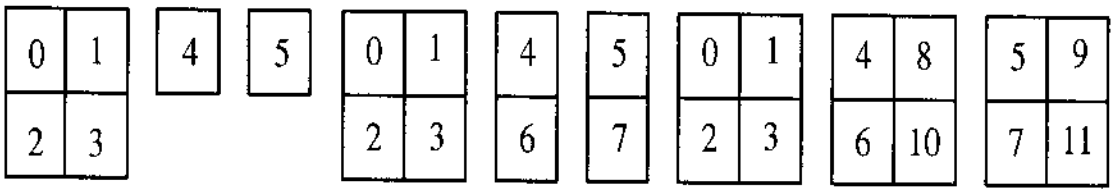
√ Biến đổi không gian màu RGB thành không gian màu YUV hay YCbCr, theo thuật toán:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ 0{,}5 & -0{,}4187 & -0{,}0813 \\ 0{,}168 & -0{,}3313 & 0{,}5 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Từ thuật toán trên, chúng ta xác định được:

- Tín hiệu độ chói Y: $Y = 0{,}299R + 0{,}587G + 0{,}144B$
- Tín hiệu hiệu màu Cb: $Cb = 0{,}5R - 0{,}14187G - 0{,}0813B$
- Tín hiệu hiệu màu Cr: $Cr = 0{,}168R - 0{,}3313G + 0{,}5B$

Các tín hiệu hiệu màu có biên độ âm cực đại -128 trong hệ thống sử dụng 8bit biểu diễn giá trị các pixel, trong khi đó tín hiệu độ chói Y có biên độ từ 0 đến 255. Do đó, để đơn giản cho bộ DCT, tín hiệu độ chói Y cần dịch mức xuống -128, tức là giá trị mỗi phần tử cần trừ đi 128. Kết quả, độ chói Y có mức biên độ trong khoảng (-128 ÷127).



a) 4:2:0(4:1:1) b) 4:2:2 c) 4:4:4

Cấu tạo của Macro block theo các tiêu chuẩn màu

Hình 2.10b) Cấu tạo của Macro block theo các tiêu chuẩn màu khác nhau

√ Chia bức ảnh thành các khối kích thước (16×16) pixel, gọi là Macro block - MB. Mỗi MB gồm 4 block các mẫu tín hiệu chói Y và hoặc 2, hoặc 4, hoặc 8 block các mẫu tín hiệu hiệu màu-CrCb. Số các khối của tín hiệu hiệu màu phụ thuộc vào tiêu chuẩn màu trong nén video.

Mỗi khối (8×8) pixel được lấy từ MB của một ảnh theo chiều từ trái qua phải, từ trên xuống dưới. Sau này chúng được nén độc lập nhau. Như vậy, ảnh phải có kích thước là bội số của MB. Gọi là *bước tiền xử lý*.

Về bước *tiền xử lý*, chúng ta có thể nói đôi điều như sau:

- Độ chói (Luminance), ký hiệu Y, là một thực thể khách quan do cường độ sáng quyết định và có thể đo được. Mắt người cảm nhận về độ chói theo nghĩa sáng nhiều hay sáng ít và nhạy cảm về sự thay đổi cường độ sáng. Vấn đề cảm nhận được màu sắc phụ thuộc vào các yếu tố cấu tạo của mắt người. So với độ chói, mắt người ít nhạy cảm hơn với sự thay đổi của màu sắc. Vì vậy, các kỹ thuật nén ảnh màu cần chú ý đến sự bảo toàn độ chói nhiều hơn đến sự bảo toàn màu sắc.

2. *Bước 2.* Biến đổi cosine rời rạc - DCT. Đã xuất hiện một vài tiêu chuẩn JPEG sử dụng biến đổi Wavelet rời rạc thay cho DCT.

3. *Bước 3.* Lượng tử hoá

$$Xq(u,v) = \text{giá trị nguyên của làm tròn}\left(\frac{X[u,v]}{q[u,v]}\right)$$

trong q[u,v] số mức lượng tử hoá ở toạ độ (u,v). q[u, v] tạo thành bảng cho trước, có bảng riêng cho tín hiệu độ chói và bảng riêng cho tín hiệu hiệu màu. Phia giải nén cần phải biết khi nén đã sử dụng các bảng mức lượng tử nào.

4. *Bước 4.* Quét zích zắc

5. *Bước 5.* Mã hoá theo chiều dài lặp lại - RLE

6. *Bước 6.* Mã hoá theo Huffman hoặc mã số học (bước 5,6 gọi là các bước nén không tiêu hao).

12. THUẬT TOÁN HỒI QUY (SÓNG)

Thuật toán nén *hồi quy* (hay truy toán) trong tiếng Anh gọi là: *wavelet*- tạm dịch là sóng gợn. Thuật toán định hướng vào ảnh đen trắng cũng như ảnh màu với phép biến đổi từ từ. Hệ số nén có thể nhận được và biến đổi trong giới hạn rộng từ 5 đến 100 lần. Khi thử nghiệm với hệ số nén lớn, tại những ranh giới đột ngột, hoặc đặc biệt diễn ra theo đường chéo sẽ xuất hiện " hiệu ứng cầu thang" - các bậc thang với độ sáng khác nhau trong một vài pixel.

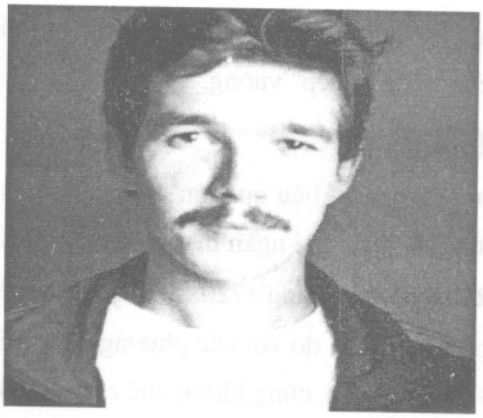
Ý tưởng của thuật toán như sau: *thay cho việc mã hoá bức ảnh riêng biệt bằng việc tính toán liên tiếp các giá trị về sự khác biệt giữa các phần tử ảnh lân cận nhau của tấm ảnh,* mà thông thường chúng sẽ nhận giá trị gần 0.

Chẳng hạn, hai số a_{2i} và a_{2i+1} luôn luôn có thể biểu diễn dưới dạng: $b_i^1 = \frac{a_{2i} + a_{2i+1}}{2}$ và $b_i^2 = \frac{a_{2i} - a_{2i+1}}{2}$. Tương tự như vậy, chuỗi a_i có thể được cặp đôi rồi chuyển thành chuỗi $b_i^{1,2}$.

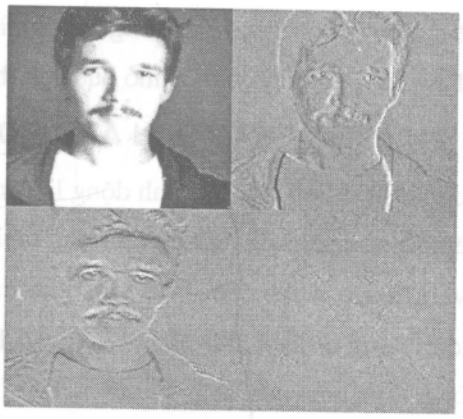
Chúng ta khảo sát một thí dụ. Giả sử chúng ta cần nén dòng số liệu gồm 8 giá trị độ sáng của các pixel (a_i)= (220, 221, 212, 218, 217, 214, 210, 202). Chúng ta nhận được chuỗi b_i^1 =(215,5; 215; 215,5; 206) và chuỗi b_i^2 = (4,5; -3; 1,5; 4). Chúng ta nhận thấy rằng các giá trị b_i^2 khá gần giá trị 0. Lập lại tính toán này đối với chuỗi b_i^1 tương tự như chuỗi a_i. Thao tác này được thực hiện theo kiểu truy toán, từ đó xuất hiện tên gọi của thuật toán. Từ dãy số liệu b_i^1 =(215,5; 215; 215,5; 206), chúng ta nhận được (215, 25; 210,75) và (0,25; 4,75). Các hệ số nhận được được làm tròn đến giá trị nguyên và nén, chẳng hạn theo thuật toán Huffman, có thể coi như kết quả mã hoá cuối cùng. Trong thí dụ này, chúng ta thực hiện biến đổi hai lần rồi được làm tròn thành

giá trị nguyên. Biến đổi *wavelet* có thể thực hiện từ 4 đến 6 lần, để cho phép nhận được hệ số nén cao.

Thuật toán cũng tiến hành tương tự cho các số liệu dạng hai chiều.



Hình 2.11a). Ảnh ban đầu.



Hình 2.11b). Bốn khối B^1, B^2, B^3, B^4.

Chẳng hạn, chúng ta có một hình vuông từ 4 điểm với độ sáng: $a_{2i,2j}$; $a_{2i+1,2j}$; $a_{2i,2j+1}$; $a_{2i+1,2j+1}$, thì được tính:

$$b^1_{i,j} = (a_{2i,2j} + a_{2i+1,2j} + a_{2i,2j+1} + a_{2i+1,2j+1})/4,$$

$$b^2_{i,j} = (a_{2i,2j} + a_{2i+1,2j} - a_{2i,2j+1} - a_{2i+1,2j+1})/4,$$

$$b^3_{i,j} = (a_{2i,2j} - a_{2i+1,2j} + a_{2i,2j+1} - a_{2i+1,2j+1})/4,$$

$$b^4_{i,j} = (a_{2i,2j} - a_{2i+1,2j} - a_{2i,2j+1} + a_{2i+1,2j+1})/4.$$

Sử dụng các công thức này đối với ảnh 512×512, chúng ta nhận được sau lần biến đổi thứ nhất 4 ma trận có kích thước 256×256 phần tử, như hình 2.11 a).- ảnh ban đầu và hình 2.11b).- sau lần biến đổi thứ nhất.

Trong khối thứ nhất dễ dàng nhận ra đó là bản copy với kích thước nhỏ của ảnh gốc. Khối thứ hai - các hiệu trung bình của cặp giá trị các pixel theo chiều ngang. Khối thứ ba - các hiệu trung bình của cặp giá trị các pixel theo chiều đứng. Còn khối thứ bốn - các hiệu trung bình của cặp giá trị các pixel theo đường chéo. Tương tự như trên, có thể lặp lại phép biến đổi đối với ma trận thứ nhất để nhận được 4 ma trận kích thước 128×128.

Khi lặp lại phép biến đổi 3 lần, sẽ nhận được 4 ma trận kích thước 64×64 và 3 ma trận 128×128, 3 ma trận 256×256. Phép nén tiếp theo nhận ra rằng trong các ma trận hiệu có một số lượng lớn các số 0, hoặc gần đến giá trị 0, chúng sẽ được nén hiệu quả sau khi lượng tử hoá.

Ưu điểm của thuật toán này là ở chỗ có thể dễ dàng cho phép thực hiện khả năng "hiện hình" từ từ bức ảnh khi truyền qua mạng. Ngoài ra, bởi vì thay cho bức ảnh ban đầu, thực tế chúng ta lưu giữ bản copy của nó, và làm đơn giản hoá trình bày ảnh "thô thiển" theo tiêu đề.

Khác với thuật toán JPEG thực hiện gia công các khối 8×8, thuật toán này thực hiện gia công các khối 2×2, 4×4, 8×8, v.v... Do trong quá trình biến đổi các hệ số của các khối được lưu giữ riêng, dễ dàng bỏ đi được việc đập nhỏ bức ảnh thành các "mảnh ghép" vuông.

13. CÁC PHƯƠNG PHÁP NÉN ẢNH ĐỘNG (VIDEO)

Vấn đề cơ bản khi làm việc với ảnh động là vấn đề dung lượng số liệu quá lớn. Chẳng hạn, khi ghi hình lên đĩa compac với chất lượng trung bình, có thể ghi được vài ngàn bức ảnh, hơn 10 giờ âm nhạc và nửa giờ video. Tín hiệu video hệ màu (RGB) số hoá dạng (720×576) pixel và 25 ảnh/sec tạo ra dòng dữ liệu 240 Mbit/sec (hoặc 1,8 Gbit/phút). Khi đó với các phương pháp nén thông thường, thực hiện nén từng ảnh riêng biệt (JPEG thuộc số đó), cũng không thể cứu vãn nổi tình thế, bởi vì ngay cả khi giảm dung lượng số liệu đi 10 ÷ 20 lần, thì nó vẫn còn quá lớn để có thể sử dụng trong thực tiễn.

Khi thực hiện nén ảnh động cần phải để ý một vài dạng dư thừa của chúng:

- Các vùng có sự gắn kết chặt (đồng màu) của ảnh sẽ không có sự thay đổi màu đáng kể giữa các pixel lân cận. Đây là một đặc tính được sử dụng ở mọi phương pháp nén ảnh có tiêu hao.
- Độ dư thừa trong mặt phẳng màu đã phản ảnh cao mức độ liên kết cường độ các màu khác nhau của thành phần ảnh và các thành phần độ sáng của nó.
- Sự tương tự giữa các ảnh- yếu tố này được sử dụng khi tốc độ 25 ảnh/sec (đó là tần số cực tiểu để không nhận ra sự nhấp nháy của ảnh, khi các ảnh có thay đổi) sẽ không khác nhau đáng kể giữa các ảnh lân cận nhau.

Vào giữa các năm 80 của thế kỷ trước, nhiều trường đại học và phòng thí nghiệm của các hãng phim phương tây đã nghiên cứu để tạo ra thuật toán nén ảnh video số. Vào năm 1988 đã ra đời nhóm MPEG (Moving Pictures Experts Group), đến năm 1990 bắt đầu xây dựng chuẩn MPEG-1, đến tháng giêng 1992 thì hoàn thành.

Thuật toán MPEG đã được thừa hưởng nhiều kết quả nghiên cứu trước đó, trước hết phải kể đến thuật toán JPEG - là thuật toán vạn năng trong nén ảnh tĩnh. Trong đó có sử dụng DCT, nó cho phép bỏ đi tuỳ ý các thành phần cao tần của ảnh (cũng đồng nghĩa với việc thay đổi tỉ số nén trong một dải rộng) mà không ảnh hưởng nhiều đến chất lượng ảnh khôi phục. Ngoài ra còn sử dụng chuyển không gian màu RGB thành không gian màu YCbCr.

13.1. Thuật toán nén video

Công nghệ nén video trong **MPEG** được phân ra thành hai phần:

- Giảm nhỏ độ dư thừa thông tin của ảnh video theo thời gian thay đổi, điều này dựa trên thực tế là các ảnh cận kề nhau theo thời gian không thay đổi đáng kể.
- Nén từng ảnh riêng biệt

a). *Giảm nhỏ độ dư thừa theo thời gian thay đổi*

Để thoả mãn các yêu cầu trái ngược nhau và làm tăng độ mềm dẻo của thuật toán, trong chuỗi ảnh- các thành phần của ảnh động- được chia ra thành bốn loại ảnh:

• Intracode pictures- ảnh mã hoá trong (ta gọi là ảnh **I**): là ảnh chuẩn cho 2 loại ảnh kia, được nén không phụ thuộc vào các ảnh khác. Bởi vậy, trong chuỗi ảnh video, hoặc nhóm ảnh video, ảnh **I** đứng đầu tiên, nó cung cấp thông tin đầu cho các ảnh kia. Ảnh **I** cho phép truy cập ngẫu nhiên, nhưng nó có tỉ lệ nén thấp nhất.

• Predictive code pictures- ảnh mã hoá dự đoán trước (ta gọi là ảnh **P**): là ảnh được nén với việc sử dụng sự tham khảo đến một ảnh trước đó (past reference image), hoặc ảnh I hoặc ảnh P.

Các giá trị của ảnh P được xác định như sau:

$$MB_{ij} \text{ (ảnh } \mathbf{P}) = MB_{ij} \text{ (ảnh hiện thời) - } MB_{ij} \text{ (ảnh tham khảo quá khứ), với mọi i, j.} \quad (2.6)$$

Ở đây dấu (–) chỉ phép tính trừ của hệ cơ số 10, với các pixel tương ứng cùng tọa độ trong mỗi MB, còn trong hệ cơ số 2, nó là phép cộng mod2. Nếu lấy ảnh hiện thời bằng ảnh tham khảo quá khứ, thì các giá trị ảnh $P \equiv 0$, tức là ảnh P chỉ gồm toàn giá trị 0.

Các vùng của ảnh hiện thời tương tự như các vùng của ảnh tham khảo trước được ghi nhận bằng véctơ dịch chuyển. Ảnh **P** được sử dụng làm ảnh tham khảo cho việc bù chuyển động của ảnh B, hoặc ảnh P khác.

• Bidirectionally predicted pictures- ảnh mã hoá dự đoán hai chiều (ta gọi là ảnh **B**) được nén với việc sử dụng sự tham khảo đến hai ảnh: một ảnh phía trước (ảnh tham khảo quá khứ - past reference image), một ảnh phía sau (ảnh tham khảo tương lai - future reference image). Bản thân ảnh B không được sử dụng làm ảnh tham khảo cho ảnh P hoặc ảnh B khác.

Các giá trị của ảnh B được xác định như sau:

$$MB_{ij}\text{(ảnh B)} = MB_{ij} \text{ (ảnh hiện thời) - } 0{,}5\times[\, MB_{ij} \text{ (ảnh tham khảo quá khứ) +}$$
$$+ MB_{ij} \text{ (ảnh tham khảo tương lai)], với mọi i, j.} \quad (2.7)$$

Ở đây phép (+) và phép (–) trong hệ cơ số 10, với các pixel tương ứng cùng tọa độ trong mỗi MB, còn trong hệ cơ số 2, chúng là phép cộng mod2. Nếu lấy ba ảnh này bằng nhau, thì các giá trị ảnh $B \equiv 0$, tức là ảnh B chỉ gồm toàn giá trị 0.

Các vùng của ảnh hiện thời tương tự như các vùng của ảnh tham khảo trước (hoặc sau) được ghi nhận bằng véctơ dịch chuyển. Hệ số nén của ảnh B cao nhất so với ảnh P và ảnh I

• Các ảnh **DC**- được nén độc lập với mức độ tiêu hao cao, chỉ sử dụng khi cần tìm kiếm nhanh. Nó giống ảnh I, tuy nhiên, chỉ có thành phần một chiều ở đầu ra khi thực hiện DCT. Đây là loại ảnh đơn màu.

Các ảnh **I** bảo đảm khả năng truy nhập tuỳ ý đến một ảnh bất kỳ, là dạng các điểm đầu vào của dòng dữ liệu để giải nén.

Các ảnh loại **P** sử dụng tham khảo đến một ảnh **I** hoặc ảnh **P** phía trước, nó có tác dụng nâng cao mức độ nén toàn bộ.

Các ảnh **B** sử dụng tham khảo đến hai ảnh, nằm phía trước và phía sau, bảo đảm nâng cao mức độ nén nhất, còn chính nó không được sử dụng làm ảnh tham khảo.

Các ảnh P và B không phải là các ảnh hoàn chỉnh mà chúng chỉ chứa đựng sự khác biệt giữa ảnh hiện thời và ảnh tham khảo trước nó (cho ảnh P) hay cả ảnh tham khảo trước và sau nó (cho ảnh B).

Chuỗi các ảnh trong phim có thể, chẳng hạn, **I B B P B B P B B P B B P B B I B B P**..., hoặc là **I P B P B P I P B P B**...

Người ta chọn tần suất của ảnh loại **I** xuất phát từ yêu cầu về thời gian truy nhập tuỳ ý và độ tin cậy của dòng dữ liệu khi truyền qua kênh có nhiễu, chọn tỉ số giữa các ảnh loại **P** và **B** xuất phát từ mức độ nén cần thiết và mức độ phức tạp của giải nén, bởi vì để giải nén ảnh loại **B** cần biết ảnh tham khảo trước cũng như ảnh tham khảo sau nó. Vì vậy, thứ tự xuất hiện ảnh (cũng như thứ tự tái tạo ảnh) khác với thứ tự truyền dẫn ảnh. Chẳng hạn, thứ tự xuất hiện ảnh là: I_1 B_2 B_3 P_4 B_5 B_6 P_7 B_8 B_9 I_{10}... Thứ tự truyền các ảnh sẽ là: I_1 P_4 B_2 B_3 P_7 B_5 B_6 I_{10} B_8 B_9 ... Lớp ảnh của dòng dữ liệu MPEG phải gắn thông tin về số thứ tự ảnh để giúp cho trình tự tái tạo ảnh đúng.

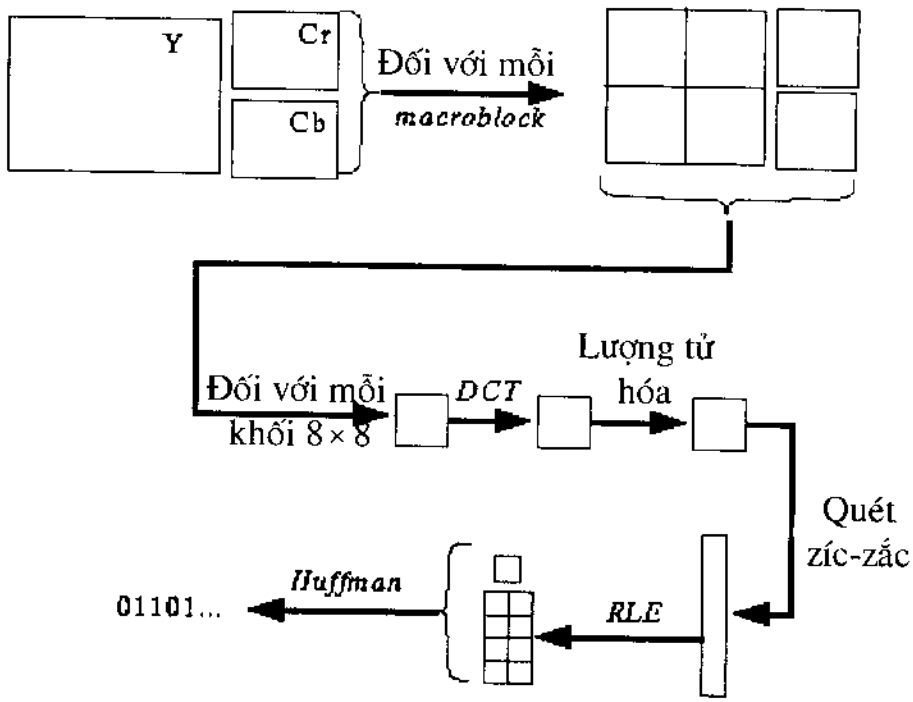
b). Nén các ảnh riêng biệt

Thực tiễn tồn tại nhiều thuật toán nén ảnh tĩnh. Trong số đó thường sử dụng thuật toán dựa trên cơ sở DCT. Thuật toán nén ảnh riêng biệt của MPEG giống như thuật toán nén ảnh tĩnh- JPEG. Mỗi ma trận cơ sở của không gian màu **YCbCr** được nén độc lập nhau nhờ DCT. Đối với ảnh P và ảnh B với các vùng ảnh hiện thời giống các vùng ảnh tham khảo, khi mã hóa sẽ thực hiện thay thế bằng các véctơ dịch chuyển đại diện cho chúng.

Quá trình nén mỗi ma trận cơ sở, hoàn toàn giống như ở thuật toán JPEG, gồm 6 bước đã biết. Tuy nhiên, bước tiền xử lý thì khác nhau, tùy thuộc vào từng loại ảnh. Sau đây, chúng ta biểu diễn các bước của quá trình nén từng loại ảnh.

b1). Nén ảnh I

Trên hình 2.12a) biểu diễn các bước nén ảnh I, bao gồm các bước:



Hình 2.12a). Các bước nén ảnh I.

- *Tiền xử lý*: Gồm các bước biến đổi không gian màu RGB thành không gian màu YCbCr. Chia tấm ảnh thành các MacroBlock. Các MB (trong trường hợp cụ thể này) thực hiện theo tiêu chuẩn 4:2:0 (4:1:1). Đối với mỗi ma trận cơ sở (8×8) thực hiện nén riêng biệt theo 6 bước của thuật toán JPEG.

- Sáu bước của thuật toán nén ảnh tĩnh JPEG đối với mỗi ma trận cơ sở (8×8) → DCT → Lượng tử hóa → Quét zich zắc → Mã RLE → Mã Huffman.

- Các bước nén ảnh I, hoàn toàn như các bước nén của thuật toán JPEG.

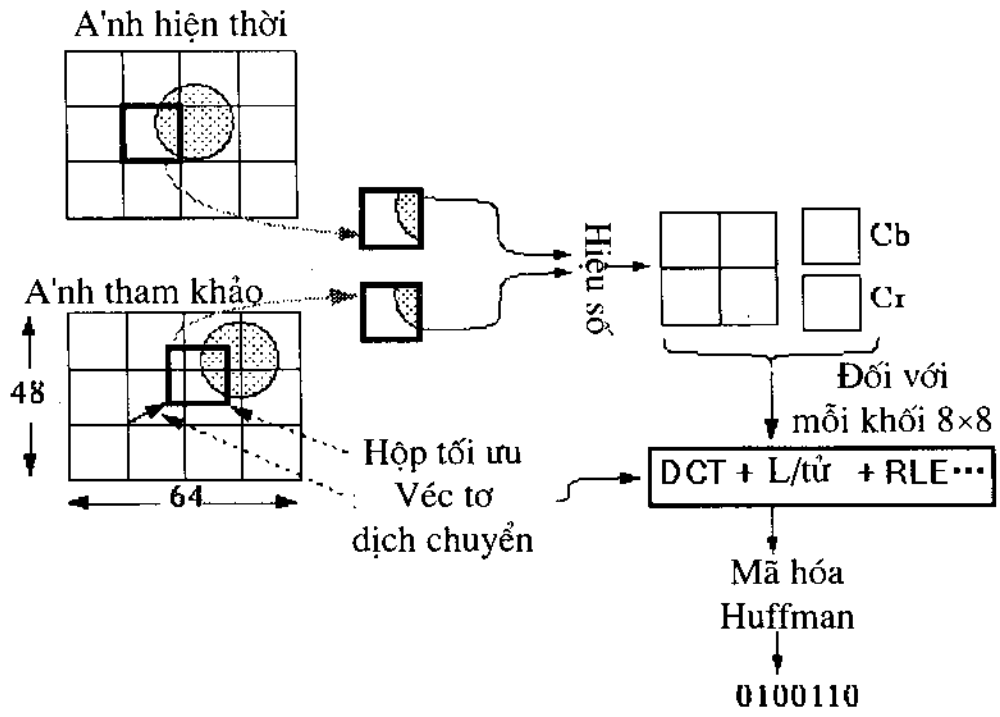
b2). Nén ảnh P

Trên hình 2.12b) biểu diễn các bước nén ảnh P. Cần chú ý rằng ảnh P là ảnh mã hóa dự đoán có tham khảo đến một ảnh trước đó (hoặc ảnh I, hoặc ảnh P).

√. Khuôn *hình tròn* trong ảnh hiện thời dùng để ký hiệu vùng ảnh dịch chuyển, mà khi mã hóa được thay thế bằng véctơ dịch chuyển thích hợp.

√. Khung *hình vuông viền đậm màu* trong ảnh hiện thời (target image) dùng để ký hiệu MB điển hình sẽ thực hiện trong các bước mã hóa.

√. Khung *hình vuông viền đậm màu* trong ảnh tham khảo quá khứ (past reference image) dùng để xác định *tối ưu* véctơ dich chuyển, gọi là hộp tối ưu.



Hình 2.12b). Các bước nén ảnh P.

- Bước tiền xử lý: Gồm các bước:

√. Xác định "hiệu số" giữa các MB tương ứng của ảnh hiện thời với các MB tương ứng của ảnh tham khảo theo công thức (2.6).

√. Xác định các vùng dịch chuyển của ảnh hiện thời so với ảnh tham khảo, rồi thay thế chúng bằng các véctơ dịch chuyển (ở mức MB).

√. Sáu bước nén tiếp theo, đối với mỗi ma trận cơ sở (8×8), hoàn toàn như ở thuật toán JPEG

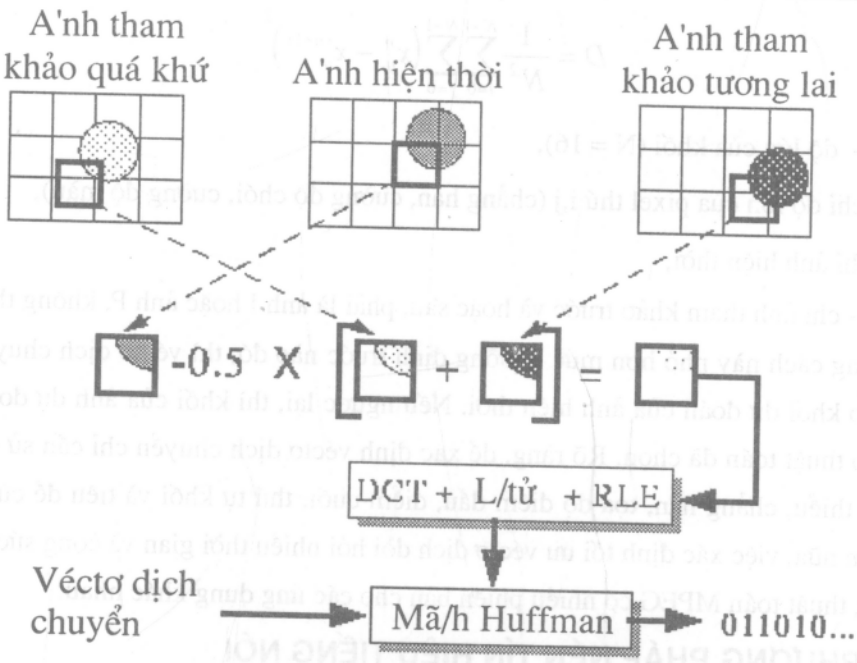
b3). Nén ảnh B

Trên hình 2.12c) biểu diễn các bước nén ảnh B. Cần chú ý rằng ảnh B là ảnh mã hóa dự đoán có tham khảo đến hai ảnh, một ảnh trước đó (hoặc ảnh I, hoặc ảnh P), một ảnh sau đó (hoặc ảnh I, hoặc ảnh P).

Bước tiền xử lý: Gồm các bước:

√. Xác định "hiệu số" giữa các MB tương ứng của ảnh hiện thời với các MB tương ứng của 2 ảnh tham khảo theo công thức (2.7).

√. Xác định các vùng dịch chuyển của ảnh hiện thời so với 2 ảnh tham khảo, rồi thay thế chúng bằng các véctơ dịch chuyển (ở mức MB).



Hình 2.12c). Các bước nén ảnh B.

Quá trình nén đối với ảnh B có thể sử dụng đến hai loại véctơ dịch chuyển. Như vậy, quá trình nén ảnh B tương tự quá trình nén ảnh P, ngoại trừ việc sử dụng véctơ dịch chuyển. Đối với ảnh B có thể sử dụng hoặc một loại véctơ dịch chuyển cho ảnh tham khảo quá khứ, hoặc một loại véctơ dịch chuyển cho ảnh tham khảo tương lai, hoặc cả hai loại véctơ dịch chuyển.

Giải nén: Đối với ảnh loại P và ảnh loại B, khi giải nén, để khôi phục ảnh hiện thời cần:

√. Khôi phục các ảnh tham khảo;

√. Sử dụng các toán tử ngược với các toán tử khi nén, để giải nén cho ảnh dự đoán;

√. Khôi phục các vùng đã được thay thế bằng các véctơ dịch chuyển;

√. Khôi phục ảnh hiện thời, từ kết quả của ba bước trên.

c). Sử dụng véctơ của khối dịch chuyển

Một khả năng đơn giản để xác định các vùng tương tự của ảnh hiện thời so với các vùng đã biết của ảnh tham khảo trước (hoặc sau) là sự tính toán mỗi khối của ảnh hiện thời từ mỗi khối của ảnh tham khảo trước (hoặc sau). Nhưng hiệu quả hơn cả là thuật toán tìm kiếm véctơ, mà nó chỉ ra khối của ảnh hiện thời đã dịch chuyển theo quan hệ với khối của ảnh tham khảo trước (hoặc sau) như thế nào.

Thuật toán chỉ ra đối với MB của ảnh hiện thời, chúng ta cần tìm ra khối tương tự với nó nhất ở các ảnh tham khảo, theo nghĩa là một số đo nào đó. Chẳng hạn, theo cực tiểu của tổng các bình phương của hiệu giá trị độ lớn các pixel ở hai khối tương ứng từ khối đang xét của ảnh tham khảo trước (hoặc ảnh sau) đến các khối của một vùng lân cận nào đó của ảnh hiện thời:

$$D = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(x_{ij}^{n} - x_{ij}^{(n\pm1)}\right)^2$$

Ở đây: N - độ lớn của khối (N = 16),

x_{ij}- chỉ độ lớn của pixel thứ i,j (chẳng hạn, cường độ chói, cường độ màu),

n - chỉ ảnh hiện thời,

n ±1- chỉ ảnh tham khảo trước và hoặc sau, phải là ảnh I hoặc ảnh P, không thể là ảnh B.

Nếu khoảng cách này nhỏ hơn mức ngưỡng định trước nào đó, thì véctơ dịch chuyển được lưu giữ thay cho khối dự đoán của ảnh hiện thời. Nếu ngược lại, thì khối của ảnh dự đoán được nén độc lập theo thuật toán đã chọn. Rõ ràng, để xác định véctơ dịch chuyển chỉ cần sử dụng một số tham số tối thiểu, chẳng hạn, tọa độ điểm đầu, điểm cuối, thứ tự khối và tiêu đề của véctơ dịch chuyển. Hơn nữa, việc xác định tối ưu véctơ dịch đòi hỏi nhiều thời gian và công sức.

Hiện thời, thuật toán MPEG có nhiều phiên bản cho các ứng dụng khác nhau.

14. CÁC PHƯƠNG PHÁP NÉN TÍN HIỆU TIẾNG NÓI

Một khối lượng lớn tin tức được truyền trong các hệ thống truyền tin ngày nay là tín hiệu tiếng nói của con người. Bởi vậy, các phương pháp *mã hoá hiệu quả* cũng như các phương pháp *nén hiệu quả* tín hiệu tiếng nói được *đặc biệt quan tâm nghiên cứu.*

Lịch sử phát triển kỹ thuật xử lý tín hiệu tiếng nói đã được bắt đầu từ rất lâu. Có thể coi được bắt đầu từ khi xây dựng đường cáp điện thoại xuyên Đại tây dương, cũng như khi xây dựng đường cáp điện thoại từ Moskva- Vladivostok.

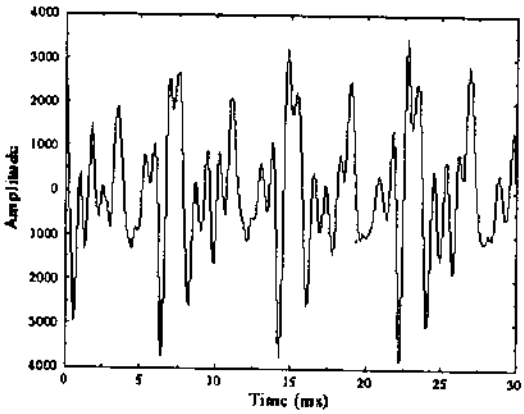
Chúng ta nghiên cứu các đặc tính cơ bản của tiếng nói, coi đó là đối tượng của mã kinh tế cũng như để truyền theo kênh liên lạc và thử giải thích xem những đặc tính nào, mà dựa trên đó có khả năng nén nó.

14.1. Các đặc điểm của tín hiệu tiếng nói

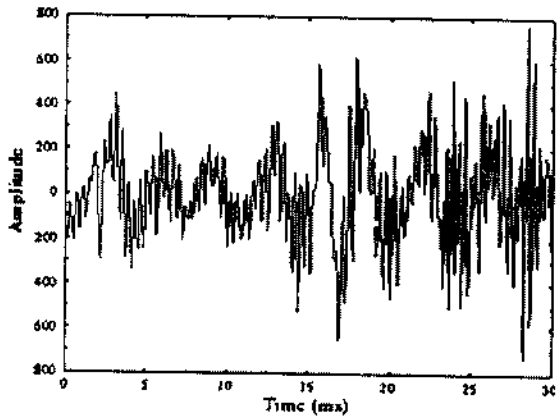
Tiếng nói là một *dao động dạng phức tạp*, nó phụ thuộc vào cách phát âm của từ, âm sắc, âm điệu của giọng nói, vào giới tính, vào tuổi tác của người nói. Phổ tần số của tiếng nói rất rộng (chẳng hạn, từ 50 đến 10.000 Hz), nhưng để truyền tín hiệu tiếng nói theo đường dây điện thoại chỉ cần các thành phần tần phổ trong khoảng 0,3 đến 3,4 kHz. Với giới hạn của dải phổ này đã làm xấu đi các thành phần của hàng loạt âm (chẳng hạn, các âm rít mà năng lượng chủ yếu của nó tập trung ở thành phần cao tần của phổ), nhưng đụng chạm rất nhỏ đến các thành phần chính yếu của tín hiệu. Giới hạn phổ tần dưới 300 Hz cũng làm xấu đi các thành phần âm trầm.

Trên các hình dưới đây biểu diễn các đoạn của tín hiệu tiếng nói, bao gồm nguyên âm- hữu thanh (hình 2.13a) và phụ âm- vô thanh (hình 2.13b) và các phổ tương ứng của các tín hiệu này (hình 2.14a); 2.14b). Từ đây thấy rõ ranh giới trong đặc trưng của các tín hiệu phù hợp, và tất nhiên cũng thấy rõ cả trong trường hợp thứ nhất, cũng như trong trường hợp thứ hai, bề rộng phổ

của tín hiệu đều không vượt khỏi 3,5 kHz. Ngoài ra, cũng thấy rằng mức phổ ứng với thành phần âm tần (tương ứng sự biến thiên chậm theo thời gian) vượt trội so với thành phần cao tần (tương ứng sự biến thiên nhanh theo thời gian). Đặc tính không bằng phẳng này của phổ là một yếu tố để ta có thể nén tín hiệu của nó.

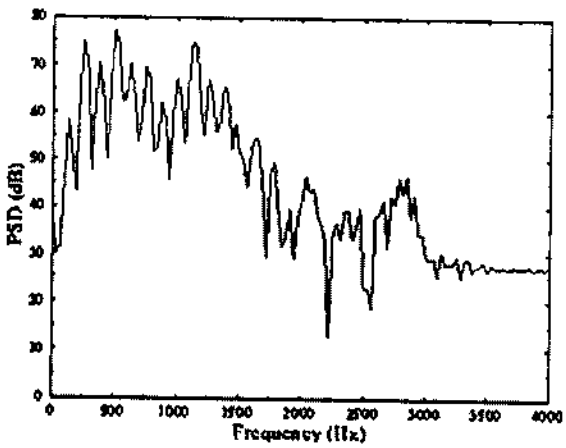


Hình 2.13a). Dạng sóng của nguyên âm.

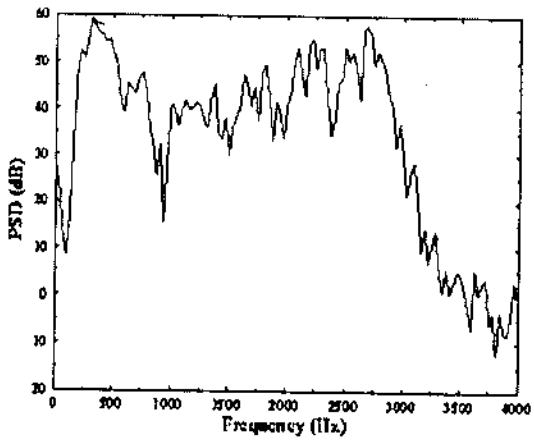


Hình 2.13b). Dạng sóng của phụ âm.

Điều đặc biệt thứ hai của tín hiệu tiếng nói, như có thể nhận xét từ thí dụ này là sự không đồng đều trong phân bố xác suất (mật độ xác suất) các giá trị tức thời của tín hiệu. Các mức biên độ tín hiệu nhỏ có phân bố xác suất lớn hơn các mức biên độ lớn.



Hình 2.14a).Dạng phổ của nguyên âm.

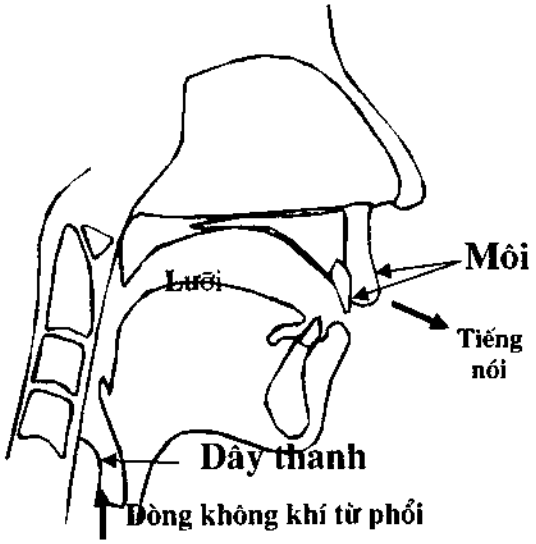


Hình 2.14b). Dạng phổ của phụ âm.

Đặc biệt này nhận thấy rõ ràng ở các đoạn có độ dài lớn mức tích cực không cao của tín hiệu tiếng nói (mức tích cực = tỉ số của thời gian có tín hiệu/ tổng thời gian (có tín hiệu + thời gian nghỉ)). Yếu tố này, như đã biết, cũng bảo đảm khả năng của mã kinh tế - khi xác suất lớn sẽ được mã hoá bằng từ mã ngắn và ngược lại.

Còn một đặc biệt nữa của tín hiệu tiếng nói là tính động theo thời gian tồn tại chúng: các đặc tính và các tham số của tín hiệu trên các đoạn khác nhau, sẽ khác nhau một cách đáng kể. Đồng thời kích thước của khoảng tĩnh chiếm khoảng vài trăm milisec. Đặc tính này của tín hiệu gây

khó khăn đáng kể cho mã hoá kinh tế nó và bắt buộc các hệ thống nén làm việc bằng mô phỏng, nghĩa là giá trị của các tham số tín hiệu ở mỗi khoảng phải được điều chỉnh.



Hình 2.15. Cơ chế hình thành tiếng nói.

Cuối cùng, điều quan trọng đặc biệt để tổ chức nén tín hiệu tiếng nói là hiểu biết về cơ chế vật lý hình thành tiếng nói. Sơ đồ đơn giản của nó được trình bày trên hình 2.15.

Tiếng nói được hình thành khi dòng không khí từ phổi được đẩy đi qua dây thanh và thanh quản. Thanh quản được bắt đầu từ dây thanh và kết thúc ở môi và có độ dài trung bình khoảng 15 ÷ 17 cm. Thanh quản, do đó mang đặc tính cộng hưởng trong việc hình thành tiếng nói, bộ các thành phần tần số đặc trưng đối với mỗi con người, được gọi là *giọng nói* (formant). Các tần số và các dải thông của *giọng nói* có thể được điều khiển khi thay đổi hình dạng của thanh quản, chẳng hạn, thay đổi vị trí của lưỡi. *Giọng nói là đặc trưng riêng biệt của mỗi con người, nhờ nhận dạng nó chúng ta có thể phân biệt người này với người kia.* Phần quan trọng của nhiều bộ mã hoá/giải mã tiếng nói là sử dụng mô hình thanh quản như *bộ lọc với các tham số thay đổi có tính thời gian ngắn.* Bởi vì hình dạng của thanh quản có thể thay đổi tương đối chậm (thường cho rằng thanh quản có thể thay đổi vị trí của lưỡi từ 20 ÷ 30 lần trong một giây), thì các tham số của bộ lọc như thế cần phải được thay đổi tương đối thưa thớt (thường không quá 20 milisec hoặc ít hơn cho một lần).

Như vậy, thanh quản tạo ra dòng không khí đẩy nó qua dây thanh. Phụ thuộc vào khả năng kích thích để phát sinh, khi đó âm thanh có thể chia thành ba loại:

1). Các nguyên âm (hữu thanh), được phát sinh khi các dây thanh rung động do bị mở, đóng, để ngắt chính dòng không khí từ phổi tới thanh quản. Sự kích thích thanh quản khi đó diễn ra gần như chuỗi xung có chu kỳ. Tốc độ (tần số) của sự mở và đóng của thanh quản xác định độ

cao của âm phát sinh. Nó có thể được điều khiển bằng thay đổi hình dạng và cường độ các dây thanh, cũng như bằng thay đổi áp suất dòng không khí chảy qua.

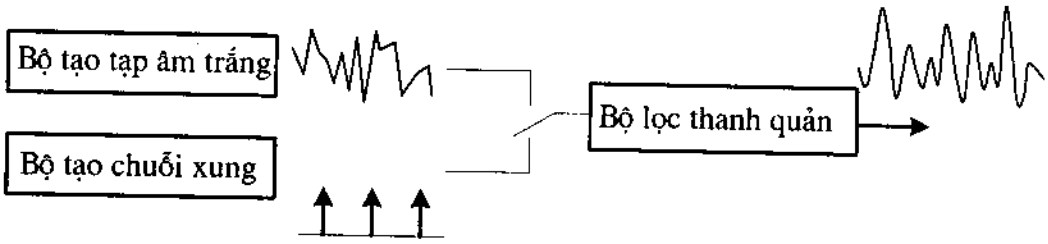
Các nguyên âm có mức độ phụ thuộc cao theo chu kỳ của âm cơ bản, với chu kỳ 2÷ 20 msec. Tính chu kỳ dài hạn này được thấy rõ trên hình 2.13a), ở đây trình bày một đoạn của tín hiệu tiếng nói từ nguyên âm.

2). Các phụ âm (vô thanh), *chẳng hạn: s, x, f,* được phát sinh khi kích thích thanh quản bởi dòng hỗn loạn tựa tạp âm, do dòng không khí có tốc độ chảy qua các dây thanh mở. Các âm như thế được trình bày trên hình 2.13b), thực tế không thấy tính chu kỳ dài hạn do dao động của các dây thanh, tuy nhiên, có thấy sự tương quan ngắn hạn do ảnh hưởng của thanh quản.

3). Các âm có đặc trưng nổ, *chẳng hạn: k, kh, g, l,* được phát sinh khi đóng thanh quản với áp suất còn dư của không khí bên ngoài trước khi mở.

Một vài âm, *chẳng hạn: z, v,* được hình thành không giống như các miêu tả ở trên đây, chúng có thể được xem như một âm hỗn hợp.

Như vậy, quá trình hình thành tiếng nói có thể được xem như *sự lọc của thanh quản với sự thay đổi theo thời gian của các tham số tín hiệu được kích thích, cũng với sự thay đổi các đặc trưng của nó* (xem hình 2.16).



Hình 2.16. Mô phỏng cơ chế hình thành tiếng nói.

Như vậy, không xem xét đến tính cực kỳ đa dạng của việc phát sinh tín hiệu tiếng nói, *hình dạng và các tham số của thanh quản, cũng như các khả năng và các tham số kích thích đủ một dạng và được thay đổi tương đối chậm,* trên hình 2.13a) và 2.13b) thấy rõ rằng tín hiệu tiếng nói có mức độ cao của dự đoán kỳ hạn lâu và kỳ hạn ngắn do tính chu kỳ dao động của dây thanh và đặc tính cộng hưởng của thanh quản. Phần lớn các bộ mã hoá/giải mã tín hiệu tiếng nói đều sử dụng điều dự đoán này, cũng như sự thay đổi chậm các tham số của mô hình hệ thống hình thành tiếng nói để giảm nhỏ tốc độ mã. Ngày nay, mô hình bộ lọc thanh quản được thực hiện bằng bộ lọc số, là bộ lọc toàn điểm cực, có hàm truyền đạt:

$$H(z) = \frac{1}{A(z)} = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}}$$

với hệ số khuếch đại G và các hệ số a_i thay đổi phù hợp.

Như vậy, tất cả các khả năng của mã hóa kinh tế (nén) tín hiệu tiếng nói có thể chia thành ba loại, được miêu tả dưới đây:

14.2. Các mã dựa theo hình dạng tín hiệu

Các bộ mã hoá/giải mã dựa theo hình dạng tín hiệu là các bộ mã hoá/giải mã tín hiệu tiếng nói đơn giản nhất, nói chung chúng không sử dụng các thông tin về quá trình hình thành tín hiệu đem mã hoá, mà đơn giản chỉ là cố gắng để tín hiệu được khôi phục gần giống với tín hiệu ban đầu nhất. Về mặt lý thuyết, chúng bất biến đối với đặc trưng của tín hiệu được đưa tới đầu vào của bộ mã hoá và chúng có thể được sử dụng để mã hoá các tín hiệu bất kỳ, không nhất thiết phải là tín hiệu tiếng nói. Các bộ mã này là đơn giản nhất theo nguyên lý hoạt động, cũng như theo cấu trúc của thiết bị, nhưng không bảo đảm mức độ nén cao (hay tốc độ bit thấp).

a). Một phương án đơn giản của bộ mã dựa theo hình dạng tín hiệu được gọi là **điều chế xung mã** (Pulse Code Modulation- **PCM**), bao gồm ba quá trình:

- Lấy mẫu tín hiệu (rời rạc tín hiệu theo thời gian): thay thế tín hiệu liên tục theo thời gian bởi các mẫu của chúng (xem chương 6). Với chu kỳ lấy mẫu T_s thoả mãn điều kiện của định lý lấy mẫu (còn gọi là định lý Hary-Nyquist hoặc Kachennhicốp). Định lý lấy mẫu phát biểu: *một tín hiệu s(t) liên tục, có phổ hữu hạn với tần số cực đại F_{max}, hoàn toàn được xác định bởi các mẫu của chúng lấy trong các khoảng thời gian* $T_S \le \frac{1}{2F_{\max}}$ (hay $f_s \ge 2F_{\max}$) ở đây T_s - chu kỳ thời gian lấy mẫu, còn $f_s = 1/T_s$ - tần số lấy mẫu, và F_{max} là tần số cao nhất của tín hiệu đem lấy mẫu.

- Lượng tử hoá biên độ các xung mẫu. Quá trình này chia biên độ các xung mẫu ra thành nhiều mức lượng tử và là quá trình làm tròn các giá trị của biên độ các xung mẫu. Kết quả của bước: biên độ các xung mẫu bằng nguyên lần số mức lượng tử. Có hai kiểu lượng tử hoá: *đều* và *không đều*.

- Mã hoá là bước biến đổi biên độ các xung mẫu đã được làm tròn thành các từ mã nhị phân có độ dài đều.

Với tín hiệu tiếng nói lấy bề rộng phổ từ tần số 300 Hz đến tần số 3.400 Hz, thì ba quá trình trên được thể hiện như sau: tần số lấy mẫu $f_s \ge 2F_{\max}$, được chọn F_s = 8000Hz (8000 mẫu/sec, T_s = 125μsec). Nếu thực hiện lượng tử hoá đều, và để nhận được chất lượng tiếng nói lý tưởng, cần nhiều hơn 4000 mức lượng tử, nói cách khác mỗi xung được mã hoá bằng từ mã dài 12 bit ($N = 2^{12} = 4096$); khi đó tốc bit sẽ là:

$$R = 8000 \text{ mẫu/sec} \times 12 \text{ bit/mẫu} = 96000 \text{ bit/sec} = 96 \text{ Kbit/sec}$$

Nếu sử dụng lượng tử hoá không đồng đều (ở mức tín hiệu nhỏ sẽ sử dụng nhiều mức lượng tử hơn ở mức lớn, để sai số tương đối lượng tử hoá sẽ không đổi cho tất cả các mức của tín hiệu). Quá trình như thế gọi là *nén dải động của tín hiệu*. Với sự tính toán đặc trưng thống kê của tín

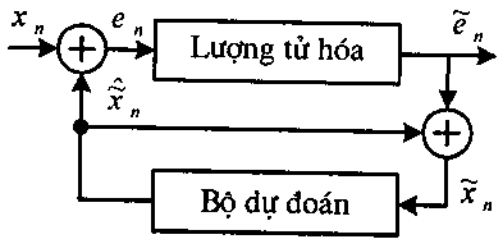
hiệu tiếng nói (theo sự phân bố của các giá trị tức thời), cũng như đặc trưng nghe phi tuyến- phân biệt tốt hơn ở các mức tín hiệu yếu, đặc trưng nén có dạng đường logarit. Hiện thời tồn tại hai luật nén: luật- μ của USA và luật- A của châu Âu. Nhờ quá trình *nén dải động của tín hiệu* này, mà số mức lượng tử hoá $N = 2^8 = 256$, nên mỗi xung mẫu được mã hoá thành từ mã dài 8 bit, khi đó tốc bit sẽ là:

$$R = 8000 \text{ mẫu/sec} \times 8 \text{ bit/mẫu} = 64000 \text{ bit/sec} = 64 \text{ Kbit/sec}$$

Như vậy, giá trị *64 Kbit/sec* được lấy làm giá trị xuất phát để so sánh mức hiệu quả và chất lượng mã hoá tín hiệu tiếng nói.

b). Một phương án tiếp theo, cho phép giảm tốc độ bit, là dự đoán các giá trị chạy của các mẫu theo theo một vài giá trị trước của nó và mã hoá không phải chính các mẫu mà là *các sai lệch dự đoán của nó*- hiệu giữa giá trị chân thực của mẫu chạy và giá trị dự đoán của nó. Nếu độ chính xác của dự đoán đạt được cao, thì sai lệch của dự đoán mẫu tiếp theo sẽ nhỏ hơn đáng kể so với chính các giá trị mẫu và để mã hoá nó cần một số bit ít hơn. Như vậy, bậc dự đoán sự biến thiên của tín hiệu đem mã hoá càng lớn, thì hiệu quả nén nó càng cao.

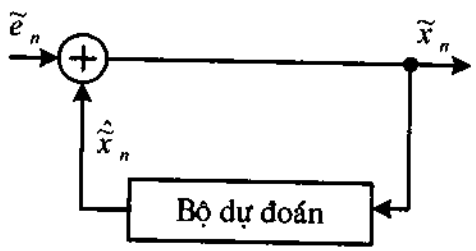
Phương án này được gọi là **điều chế xung mã vi sai** (Differential Pulse Code Modulation- **DPCM**), khi đó mã hoá không phải chính các giá trị mẫu của tín hiệu, mà là sai lệch của chúng từ một vài mẫu dự đoán.



$$e_n = x_n - \hat{\tilde{x}}_n = x_n - \sum_{i=1}^{p} a_i \tilde{x}_{n-i}$$

$\tilde{e}_n - e = \tilde{x}_n - x_n =$ sai số lượng tử hoá

Hình 2.17. Bộ mã hoá DPCM.



$$\tilde{x}_n = \hat{\tilde{x}}_n + \tilde{e}_n$$

Hình 2.18. Bộ giải mã DPCM.

Gọi x_n là giá trị mẫu hiện thời và $\hat{x}_n$ là giá trị mẫu dự đoán của x_n, được định nghĩa như sau:

$$\hat{x}_n = \sum_{i=1}^{p} a_i x_{n-i} \tag{2.8}$$

Như vậy, $\hat{x}_n$ là tổ hợp tuyến tính của p mẫu trước, a_i là các hệ số của bộ dự đoán, các hệ số a_i được chọn để tối thiểu hoá hàm sai lệch nào đó giữa $\hat{x}_n$ và x_n. Phép dự đoán theo (2.8) được gọi là dự đoán bậc p. Nếu lấy p =1, ta có dự đoán bậc 1 là đơn giản nhất, nhưng kém chính xác nhất. Trên hình 2.17 và hình 2.18 trình bày sơ đồ bộ mã hoá và giải mã DPCM.

Khi chọn bậc dự đoán thích hợp, thì kết quả mã hoá theo DPCM cho 5 - 6 bit/mẫu hoặc tốc độ bit R = 40 - 48 Kbit/sec.

c). Hiệu quả của DPCM có thể được nâng cao thêm, nếu dự đoán và lượng tử hoá tín hiệu sẽ được thực hiện không phải trên cơ sở của các đặc trưng trung bình nào đó của nó, mà phải tính toán các giá trị chạy của chúng theo sự thay đổi theo thời gian, có nghĩa là *phải mô phỏng*. Nếu tốc độ thay đổi theo thời gian của tín hiệu lớn, thì phải tăng số bước lượng tử hoá, ngược lại, khi tín hiệu thay đổi chậm theo thời gian, thì phải giảm số bước lượng tử hoá. Khi đó sai số dự đoán sẽ nhỏ, kéo theo giảm số bit cho một mẫu. Phương án mã hoá như thế được gọi là **điều chế xung mã vi sai tự thích nghi** (Adaptive DPCM= **ADPCM**). Sử dụng ADPCM với 4bit/ mẫu hay tốc độ bit 32 Kbit/sec, bảo đảm chất lượng tiếng nói như PCM 64 Kbit/sec, nhưng tốc độ bit giảm đi còn một nửa.

Ngày nay, chuẩn mã hoá ADPCM được sử dụng rộng rãi để nén tín hiệu tiếng nói trong các hệ thống Cityphone, hệ thống thông tin vi tế bào DECT, hệ thống điện thoại số không dây, v.v..

Ngày nay, cũng tồn tại các chuẩn ADPCM với tốc độ bit 40, 24, 16 Kbit/sec (hai trường hợp sau cho chất lượng tiếng nói kém hơn ADPCM 32 Kbit/sec).

d). Điều chế Delta *(DM)*: Điều chế delta là một trường hợp của DPCM, trong đó mỗi từ mã chỉ có 1bit. Đây là phương án mã hoá đơn giản nhất hiện có. Tốc độ bit R bằng tần số lấy mẫu f_s. Nhưng để phối hợp với các hệ thống có từ mã bit cao, DM phải tăng tần số lấy mẫu một cách đáng kể. Chẳng hạn, với hệ thống 32 kênh, tốc độ lấy mẫu phải là 50 Mega xung mẫu/sec. Để cải thiện các nhược điểm của DM, người ta thực hiện điều chế Delta tự thích nghi (ADM).

Như vậy, nén tín hiệu tiếng nói trên cơ sở mã hoá dựa theo hình dạng tín hiệu, trong trường hợp tốt nhất giảm được tốc độ bit đi hai, ba lần. Nếu với các phương pháp này tiếp tục giảm tốc độ bit, thì chất lượng tín hiệu tiếng nói sẽ giảm nghiêm trọng.

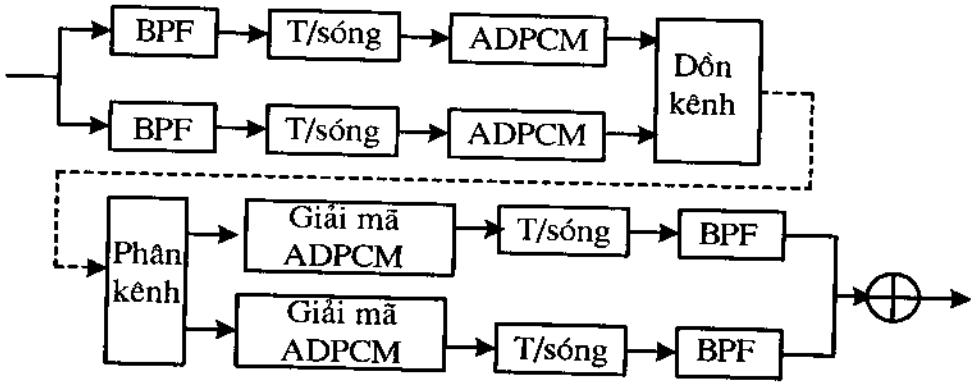
Tất cả các phương pháp mã hoá dựa theo hình dạng tín hiệu vừa nêu ở trên đều đơn thuần sử dụng đặc trưng theo thời gian của tín hiệu. Tuy nhiên, tồn tại cả các xu hướng khác, chúng sử dụng đặc trưng theo tần số của tín hiệu. Sau đây chúng ta nghiên cứu hai phương pháp nén tín hiệu tiếng nói trong miền tần số.

e). Mã hoá băng con (Sub-Band Coding- SBC): tín hiệu đầu vào được chia (được lọc) ra thành một vài băng tần số (băng con- sub-band) và tín hiệu trong mỗi băng con được mã hoá riêng biệt, chẳng hạn, sử dụng kỹ thuật ADPCM. Trên hình 2.19 trình bày sơ đồ đơn giản của kỹ thuật mã hoá băng con, với việc chia thành hai băng con.

Bởi vì mỗi băng tần con có dải thông hẹp hơn (tổng dải thông của các băng con phải bằng dải thông của tín hiệu ban đầu), cho nên tần số lấy mẫu trong mỗi băng con cũng nhỏ hơn. Kết quả, tổng tốc độ bít của tất cả các băng con cũng ít nhất không vượt quá tốc độ bit khi mã hoá tín hiệu ban đầu. Tuy nhiên, ở kỹ thuật này có một vài nhược điểm. Vấn đề ở chỗ là độ nhạy của

thính giác con người đối với tín hiệu và đối với độ méo dạng của chúng là khác nhau ở các đoạn tần số khác nhau.

Độ nhạy của thính giác đạt cực đại trong dải tần 1-1,5 kHz và giảm dần về cả hai phía (phía dải tần cao, cũng như dải tần thấp so với dải này). Như vậy, nếu với dải tần mà độ nhạy của thính giác đạt cực đại, thực hiện lượng tử hoá nhiều mức hơn so với dải tần về hai phía, thì có thể giảm được tốc độ bit. Thực tế khi thực hiện kỹ thuật mã hoá băng con, kết quả nhận được chất lượng tốt tiếng nói với tốc độ 16 - 32 Kbit/sec. Bộ mã thực hiện phức tạp hơn so với kỹ thuật ADPCM



Hình 2.19 Sơ đồ đơn giản mã hoá băng con (2 băng con).

đơn thuần. Tuy nhiên, không thể mong đợi đơn giản hơn để nén hiệu quả tín hiệu tiếng nói.

f). Mã hoá biến đổi thích nghi (Adaptive Transformable Coding- ATC): chia các mẫu của tín hiệu nguồn thành các khung, mỗi khung N_f mẫu. Sau đó số liệu trong mỗi khung được thực hiện biến đổi tuyến tính rời rạc, chẳng hạn, biến đổi cosine rời rạc, để nhận được các hệ số phù hợp với đặc trưng phổ của từng khung và được lượng tử hoá. Kỹ thuật này rất giống kỹ thuật nén JPEG, nhưng được ứng dụng với tín hiệu tiếng nói. Kết quả, nhận được tốc độ bit từ 12- 16 Kbit/sec với chất lượng tiếng nói tương đương các kỹ thuật khác. Chuẩn nén MPEG áp dụng cho tín hiệu audio (phổ của chúng rộng hơn phổ tiếng nói) lấy mỗi khung $N_f = 32$ mẫu, rồi thực hiện DCT như chuẩn JPEG.

Để có các thông tin chi tiết hơn về các mã hoá theo hình dạng tín hiệu tiếng nói, độc giả có thể tham khảo các tài liệu khác, chẳng hạn [9].

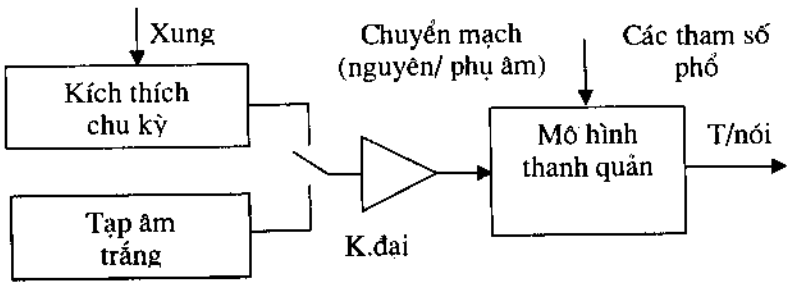
14.3. Các mã dựa trên mô hình nguồn

Khác biệt với các mã dựa theo hình dạng tín hiệu, các mã dựa trên mô hình nguồn, khi đó từ tín hiệu mã hoá lấy ra thông tin về các tham số của mô hình nguồn của tín hiệu này. Đồng thời, *kết quả của mã hoá không phải là mã của tín hiệu mà là mã của các tham số nguồn tín hiệu này.*

Bộ mã nguồn để mã tín hiệu tiếng nói được gọi là Vocoder và làm việc như sau:

Thanh quản được hình dung như là mạch lọc tuyến tính với các tham số thay đổi theo thời gian, được kích thích bởi hoặc nguồn tạp âm trắng (khi hình thành các phụ âm), hoặc bởi chuỗi xung nối tiếp với chu kỳ của âm cơ bản (tone) (khi hình thành các nguyên âm) như hình 2.20.

Mô hình tuyến tính của hệ thống tiếng nói và các tham số của nó có thể tìm được bằng các khả năng khác nhau. Dạng của bộ Vocoder phụ thuộc vào các khả năng tìm kiếm như thế.



Hình 2.20. Mô hình nguồn hình thành tiếng nói.

Các thông tin, mà bộ Vocoder nhận được trong kết quả phân tích tiếng nói và truyền cho bộ giải mã, đó là các tham số của *bộ lọc thanh quản, bảng tra nguyên/phụ âm, công suất của tín hiệu kích thích và chu kỳ của tone cơ bản đối với các nguyên âm.* Các tham số này cần phải được thay đổi trong khoảng 10- 20 msec, để bảo đảm cho tính động của tín hiệu tiếng nói.

Vocoder, khác với bộ mã theo hình dạng tiếng nói, cố gắng hình thành nên tín hiệu, mà tiếng kêu như tiếng nói nguyên bản, không chú ý đến sự khác biệt của hình dạng tín hiệu này với tín hiệu nguồn. Khi đó, tốc độ bit ở đầu ra thông thường không lớn hơn 2,4 Kbit/sec, nghĩa là giảm đi 15 lần so với ADPCM.

Đáng tiếc là chất lượng tiếng nói do Vocoder tạo ra rất xa lý tưởng, mặc dù các âm của nó khá đa dạng, nhưng tuyệt đối không tự nhiên. Ngay cả khi chấp nhận nâng cao tốc độ bit, thì thực tế cũng không nâng cao được chất lượng tiếng nói, bởi vì để mã hoá đã chỉ có thể chọn mô hình đơn giản của thanh quản. Khó khăn đặc biệt về giả thiết rằng tiếng nói chỉ gồm nguyên âm và phụ âm, mà không có các âm trung gian nào khác.

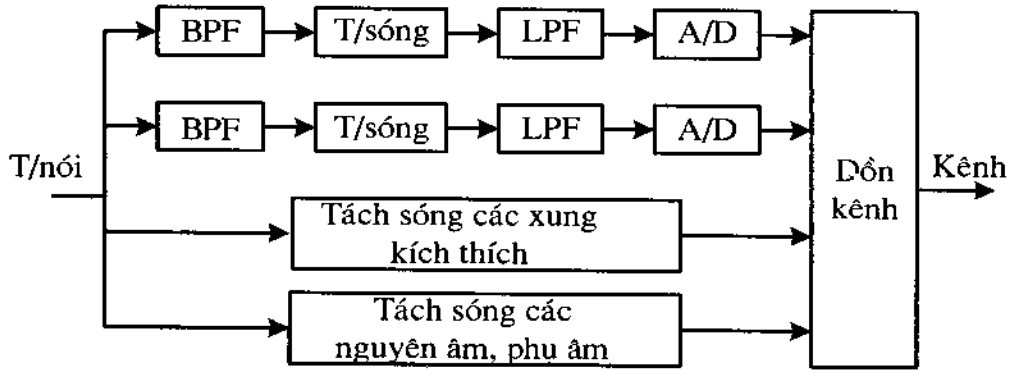
Các ứng dụng cơ bản của Vocoder tìm thấy trong lĩnh vực quân sự. Ở đây, điều chính yếu không phải là chất lượng tiếng nói tự nhiên, mà là mức độ nén cao, hay tốc độ bit thấp, cho phép bảo đảm an toàn tin tức khi truyền.

Chúng ta khảo sát ngắn gọn cơ sở một vài dạng Vocoder đã biết.

a). Các Vocoder kênh

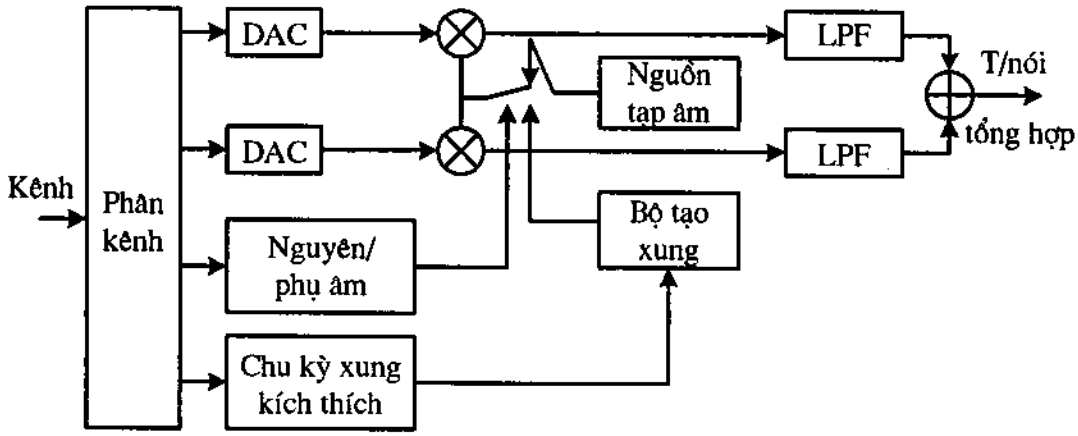
Dạng Vocoder được biết nhiều nhất ngày nay đã được đề cập từ năm 1939. Vocoder này sử dụng đặc tính độ nhạy kém của thính giác con người đối với dịch pha (thời gian) không đáng kể của tín hiệu.

Đối với mỗi đoạn tín hiệu tiếng nói dài 20 - 30 msec nhờ các BPF dải hẹp mà biên độ phổ của nó được xác định. Số bộ lọc càng nhiều, thì phổ càng được đánh giá chính xác, nhưng càng cần nhiều bit hơn để mã hoá nó, tức là càng làm tăng tốc độ bit. Tín hiệu từ đầu ra BPF được tách sóng, rồi đưa qua LPF, sau đó được ADC để biến thành mã nhị phân. Quá trình này được biểu diễn trên hình 2.21.



Hình 2.21. Bộ mã hoá Vocoder kênh.

Như vậy, các tham số biến thiên chậm của thanh quản được xác định và ngoài ra, nhờ bộ tách sóng của âm cơ bản và các nguyên âm mà chu kỳ âm kích thích cơ bản và dấu hiệu của nguyên/phụ âm cũng được xác định.



Hình 2.22. Bộ giải mã Vocoder kênh.

Vocoder kênh có thể thực hiện trong dạng số, cũng như trong dạng tương tự và bảo đảm tiếng nói đủ rõ ràng, khi tốc độ bit ở đầu ra khoảng 2,4 Kbit/sec.

Bộ giải mã (hình 2.22) tiếp nhận các thông tin mà bộ mã hoá đã gia công, thực hiện gia công chúng theo trình tự ngược, tổng hợp tín hiệu tiếng nói ở đầu ra của nó, sao cho giống với tín hiệu tiếng nói ban đầu nhất.

Không nên mong đợi chất lượng tốt của tiếng nói được khôi phục từ Vocoder này. Bộ Vocoder kênh được sử dụng dựa trên các yêu cầu nghiêm khắc, cũng như hệ số nén cao được sử dụng trong thông tin quân sự, thông tin vũ trụ, ...

b). Vocoder cùng hình (homomorphic)

Gia công cùng hình (hay đồng hình) các tín hiệu là một trong các phương pháp gia công phi tuyến, mà có thể được áp dụng hiệu quả đối với các tín hiệu phức tạp, chẳng hạn, tín hiệu tiếng nói.

Với sự tính toán để sử dụng trong Vocoder của mô hình hệ thống thanh quản, tín hiệu tiếng nói- s(t) có thể trình bày như là sự xoắn (tích chập) theo thời gian của đáp ứng xung - h(t) của thanh quản với tín hiệu kích thích đầu vào-x(t):

$$s(t) = x(t) * h(t) = \int_{-\infty}^{+\infty} x(\lambda).h(t-\lambda)d\lambda$$

Trong miền tần số điều này phù hợp là tích hàm truyền đạt -*H(f)* của thanh quản và phổ tín hiệu kích thích đầu vào *X(f)*:

$$S(f) = X(f).H(f)$$

Cuối cùng, nếu lấy logarit của tích này, thì nhận được tổng logarit của phổ tín hiệu kích thích đầu vào và hàm truyền đạt của thanh quản. Bởi vì tai của con người thực tế không nhạy cảm với dịch pha của tín hiệu, nên có thể chỉ sử dụng phổ biên độ:

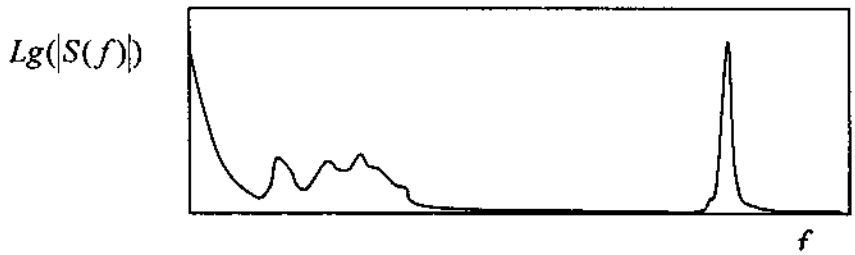
$$Lg(|S(f)|) = Lg(|X(f)|) + Lg(|H(f)|)$$

ở đây: $S(f) = |S(f)|.e^{i\theta(f)}$ - phổ tín hiệu tiếng nói; $X(f) = |X(f)|.e^{i\theta(f)}$ - phổ tín hiệu kích thích đầu vào; $H(f) = |H(f)|.e^{i\theta(f)}$ - hàm truyền đạt của thanh quản.

Nếu bây giờ chúng ta thực hiện biến đổi Fourier ngược đối với $Lg(|S(f)|$, tức là:

$Lg(|s(t)|\ ) = FT^{-1}[Lg(|S(f)|)]$, thì chúng ta nhận được cái gọi là *kép của tín hiệu.*

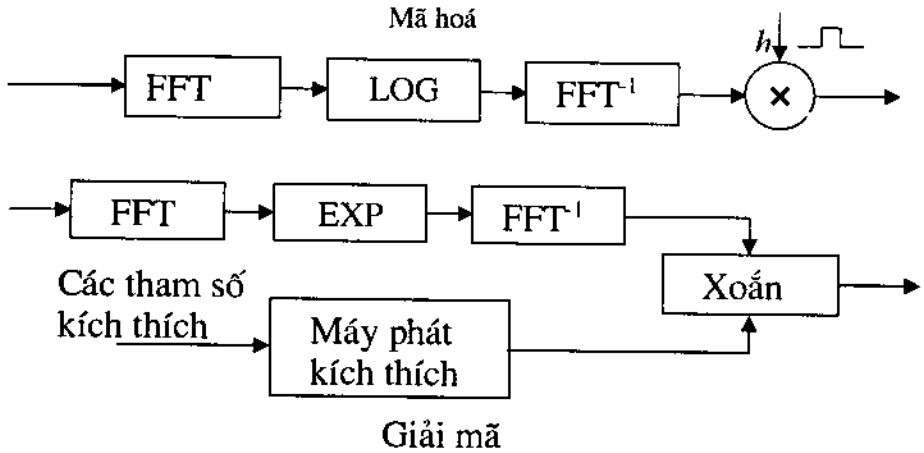
Các tham số bộ lọc thanh quản thay đổi theo thời gian tương đối chậm (phổ của chúng nằm ở vùng tần số thấp - LF), trong khi đó tín hiệu kích thích là hàm biến thiên nhanh theo thời gian (phổ của nó tập trung ở vùng tần số cao - HF). Bởi vậy trong kép của tín hiệu tiếng nói, các thành phần này có thể được tách ra (hình 2.23), và có thể được mã hoá riêng biệt



Hình 2.23. Dạng kép của tín hiệu.

Sơ đồ mã hoá/giải mã Vocoder cùng hình của tín hiệu tiếng nói trình bày trên hình 2.24. Trong đó: FFT- phép biến đổi Fourier rời rạc nhanh thuận; còn FFT^{-1}- phép biến đổi Fourier rời rạc

nhanh ngược; LOG- phép lấy logarit; EXP- phép lấy hàm mũ. Thực tiễn nhận thấy rằng với việc sử dụng nó có thể nhận được tốc độ bit khoảng 4Kbit/sec.



Hình 2.24. Bộ mã hoá/giải mã Vocoder cùng hình.

c). Vocoder dựa theo giọng nói

Như đã nhận xét trước đây, thông tin cơ bản về tín hiệu tiếng nói được chứa đựng trong vị trí và bề rộng phổ của các thành phần *giọng nói* (formant) của nó. Nếu với việc xác định chính xác cao và mã hoá các tham số của *giọng nói* này, kết quả có thể nhận được tốc độ bit rất thấp - dưới 1 Kbit/sec. Nhưng thực hiện điều này rất khó khăn, vì các Vocoder *dựa theo giọng nói* hiện giờ không thấy phổ biến rộng rãi.

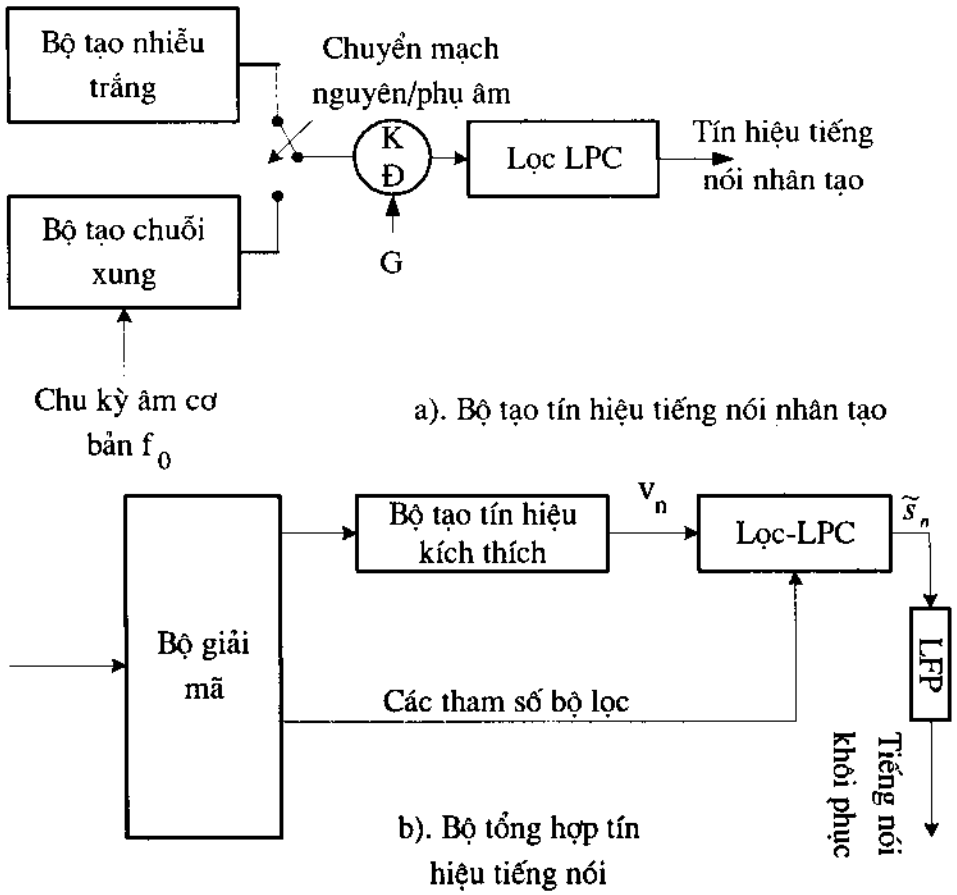
d). Các Vocoder mã hoá dự đoán tuyến tính (LPC)

Các Vocoder dựa trên cơ sở *dự đoán tuyến tính* cũng sử dụng mô hình thanh quản. Điều khác biệt của chúng- đó là phương pháp xác định các tham số mô hình thanh quản. Mã hoá dự đoán tuyến tính- LPC giả thiết thanh quản là bộ lọc toàn cực (rời rạc) với hàm truyền đạt:

$$H(z) = \frac{G}{1 - \sum_{i=1}^{p} a_i z^{-i}} = \frac{1}{A(z)}$$

trong đó mỗi mẫu hiện thời của tín hiệu có thể nhận được theo tổ hợp tuyến tính của một số mẫu trước đó: $s_n = \sum_{i=1}^{p} a_i . s_{n-i}$.

Trong các Vocoder trên cơ sở LPC, tín hiệu tiếng nói được phân ra thành các đoạn, mỗi đoạn dài khoảng 20 msec. Các hệ số dự đoán tuyến tính a_i được xác định trong mỗi đoạn. Công dụng của bộ dự đoán là loại bỏ sự tương quan giữa các mẫu tín hiệu lân cận nhau, còn sự tương quan lâu về thời gian của tín hiệu được biểu hiện rõ rệt hơn nhiều, điều này cho phép xác định chính



Hình 2.25. Sơ đồ khối Vocoder trên cơ sở LPC.

xác tần số của âm cơ bản f_0 và tách được các dấu hiệu của nguyên/phụ âm. *Tần số âm cơ bản f_0, các dấu hiệu của nguyên/phụ âm, hệ số khuếch đại G, các hệ số dự đoán a_i được lượng tử hoá, mã hoá và truyền đến phía thu.* Ở phía thu các tham số này được giải mã. Ở đây các mẫu của tín hiệu kích thích được tổng hợp lại, rồi truyền qua bộ lọc-LP. Đầu ra bộ lọc nhận được đánh giá mẫu tín hiệu tiếng nói $\tilde{s}_n$. Các mẫu này được cho qua mạch lọc thông thấp LPF, để khôi phục lại tiếng nói. Sơ đồ bộ tổng hợp tín hiệu tiếng nói như hình 2.25b.

Ngày nay các Vocoder trên cơ sở LPC được sử dụng rộng rãi nhất, bởi vì tất cả đều sử dụng mô hình bộ lọc thanh quản, chúng làm việc rất tốt. Với chất lượng tiếng nói không tồi, các Vocoder trên cơ sở LPC cho tốc độ bit đến 2,4 Kbit/sec.

14.4. Các phương pháp lai ghép mã hoá tín hiệu tiếng nói

Các phương pháp mã hoá tín hiệu tiếng nói lai ghép, hoặc tổ hợp, chứa đầy trong chúng những chỗ phân cách giữa *các mã dựa theo hình dáng tín hiệu*, trong đó hoàn toàn không đếm xỉa đến tính tự nhiên của tín hiệu tiếng nói, và *các mã dựa trên mô hình nguồn*, trong đó để mã hoá

không phải tín hiệu, mà là các tham số của mô hình nguồn phát sinh tiếng nói. Như đã nhận xét trước đây, các mã hoá dựa theo hình dáng tín hiệu có thể bảo đảm chất lượng tiếng nói tốt, khi tốc độ bit 16 Kbit/sec, nhưng nói chung không thể thấp hơn; trong khi đó các Vocoder bảo đảm gia công tiếng nói với tốc độ bit 2,4 Kbit/sec và thấp hơn, nhưng không thể bảo đảm chất lượng tiếng nói, dù với tốc độ bit bất kỳ.

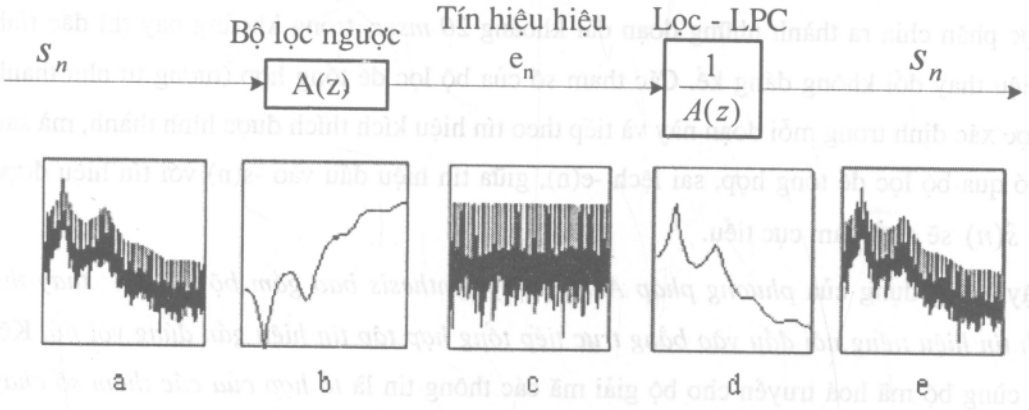
Ngày nay phổ biến nhất là các phương pháp mã hoá lai ghép làm việc trong miền thời gian (tức là với tín hiệu, chứ không phải với phổ, cũng như với bất kỳ các biến đổi tuyến tính nào khác của nó), được dựa trên *sự phân tích tín hiệu qua tổng hợp nó (Analysis-by-Synthesis)* - được gọi là bộ mã- **AbS**. Các bộ mã này, cũng như các Vocoder, sử dụng mô hình thanh quản, nhưng có một vài điểm khác- *việc lựa chọn tín hiệu kích thích phải bảo đảm sự trùng tốt nhất của tín hiệu tổng hợp trên cơ sở của tín hiệu tiếng nói ban đầu.*

Các mã **AbS** đã được đề cập lần đầu tiên cách đây không lâu- vào năm 1982 và trong dạng đầu tiên được gọi là các mã **MPE** (Multi-Pulse Excited- mã hoá được kích thích nhiều xung). Sau này hoàn thiện hơn với tên gọi các mã **RPE** (Regular- Pulse Excited - mã hoá được kích thích xung đều đặn) và các mã **CELP** (Codebook Excited Linear Predictive- dự đoán tuyến tính được kích thích bởi quyển sách mã). Ngày nay tồn tại cả những dạng khác của chúng, nhưng tất cả chúng đều dựa trên ý tưởng chung.

a). Các mã RELP

Để hiểu được dựa trên những cơ sở nào mà các mã AbS có hiệu quả và có chất lượng, trước hết chúng ta hãy nghiên cứu nguyên tắc làm việc của bộ mã **RELP** (Residual Excited Linear Prediction- mã hoá dự đoán tuyến tính được kích thích phần dư).

Nếu tín hiệu tiếng nói có phổ như hình 2.26a cho qua bộ dự đoán tuyến tính có đặc trưng tần số như hình 2.26b, đặc trưng tần số này ngược với đặc trưng phổ của tín hiệu tiếng nói (theo nghĩa tích của hai đặc trưng tần số là hằng số theo tần số), thì tương quan giữa các mẫu của tín hiệu đầu ra (các sai số dự đoán) sẽ giảm đáng kể. Nếu bộ dự đoán đã được thực hiện đủ tốt, thì ở đầu ra dự đoán thực tế sẽ là tạp âm trắng với phổ phẳng đều, như hình 2.26c.



Hình 2.26. Quá trình xử lý tín hiệu của mã RELP

Trong hình 2.26: s_n = các mẫu của tín hiệu tiếng nói, n = 0, 1,..., N-1; 1/A(z) - hàm truyền của bộ lọc dự đoán tuyến tính.

$e_n = s_n - \sum_{i=1}^{p} a_i . s_{n-i}$ - sai lệch giữa mẫu hiện thời với tổ hợp tuyến tính của một vài mẫu trước đó, gọi là mẫu dự đoán, còn e_n gọi là tín hiệu hiệu. Các hệ số a_i được lựa chọn để tối thiểu hoá trung bình bình phương sai lệch này.

Tạp âm trắng (sai lệch của dự đoán) này mang thông tin về tín hiệu tiếng nói được mã hoá, nếu lại cho nó đi qua bộ lọc- LPC (mã hoá dự đoán tuyến tính) với đặc trưng tần số như hình 2.26d (như đặc trưng phổ của tín hiệu tiếng nói đem mã hoá), thì ở đầu ra chúng ta khôi phục chính xác tuyệt đối tín hiệu tiếng nói ban đầu. Bởi vì thông tin về sai lệch dự đoán có phổ phân bố gần như bằng phẳng, từ đó nảy sinh ý tưởng mã hoá và truyền chỉ một phần nhỏ của phổ sai lệch dự đoán E(f), và khôi phục phần còn lại trong giải mã.

Trong các mã RELP các tín hiệu sai lệch dự đoán được truyền qua bộ lọc thông thấp (LPF), với tần số cắt khoảng 1 kHz, tín hiệu ở đầu ra mạch lọc được mã hoá theo dạng, chẳng hạn, DPCM. Khi giải mã các sai lệch dự đoán được khôi phục bằng cách đặt nó vào vùng tần số xa so với dải thông của LPF khi mã hóa.

Bộ mã RELP sẽ làm việc lý tưởng, nếu trong quá trình dự đoán tuyến tính chúng ta nhận được tạp âm trắng. Tuy nhiên, vì tồn tại trong tín hiệu tiếng nói các thành phần giọng nói giả chu kỳ, bộ dự đoán tuyến tính không có thể gạt bỏ sự tương quan dài hạn với chu kỳ của âm cơ bản giọng nói và chúng rõ ràng có mặt trong phổ của tín hiệu sai lệch dự đoán. Nếu bây giờ, chúng ta đưa E(f) qua LPF thì các thành phần giọng nói cao tần sẽ bị gạt bỏ và sau này không thể khôi phục được chúng.

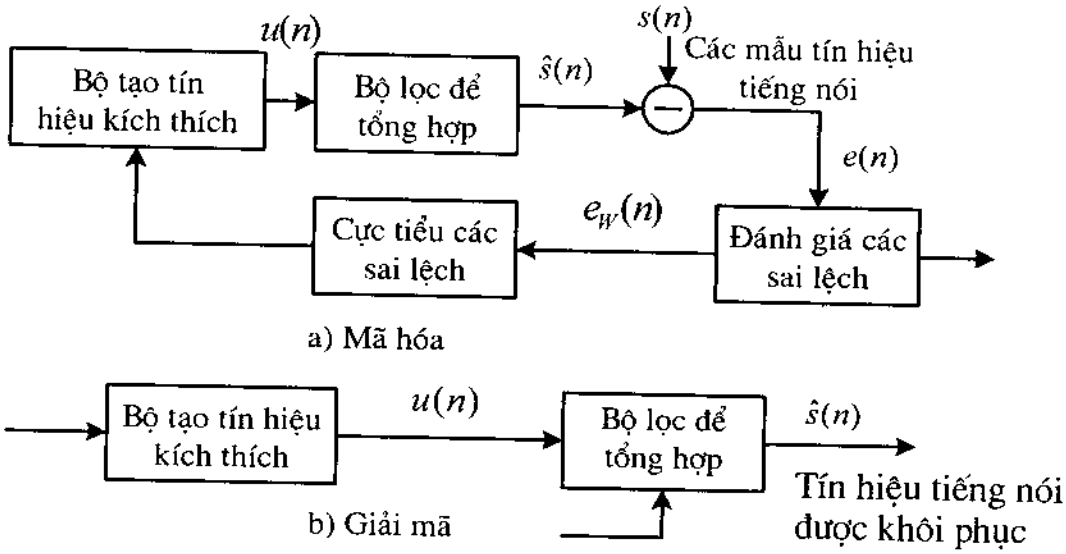
Các bộ mã RELP cho phép nhận được chất lượng tiếng nói khá tốt, khi tốc độ bit 9,6 Kbit/sec. Tuy nhiên, trong mọt chừng mực nào đấy, ở chúng mang các nhược điểm của Vocoder - đó là tiếng nói được khôi phục mang đặc trưng tổng hợp.

b). Nguyên lý chung các mã dựa trên AbS

Các mã AbS làm việc như sau. Tín hiệu đầu vào để mã hoá (đã ở dạng số, trong dạng dòng các mẫu) được phân chia ra thành những đoạn dài khoảng *20 msec*, trong khoảng này thì đặc tính của tín hiệu thay đổi không đáng kể. Các tham số của bộ lọc để tổng hợp (tương tự như thanh quản) được xác định trong mỗi đoạn này và tiếp theo tín hiệu kích thích được hình thành, mà sau đó đưa nó qua bộ lọc để tổng hợp, sai lệch -e(n), giữa tín hiệu đầu vào -s(n) với tín hiệu được tổng hợp $\hat{s}(n)$ sẽ được làm cực tiểu.

Như vậy, công dụng của *phương pháp Analysis-by-Synthesis bao gồm bộ mã với thay thế phân tích tín hiệu tiếng nói đầu vào bằng trực tiếp tổng hợp tập tín hiệu gần đúng với nó.* Kết quả cuối cùng bộ mã hoá truyền cho bộ giải mã các thông tin là *tổ hợp của các tham số chạy*

của mạch lọc để tổng hợp và thông tin của tín hiệu kích thích. Mong muốn rằng các số liệu này là ít nhất. Bộ giải mã theo các tham số này sẽ khôi phục lại tín hiệu đem mã hoá, hơn nữa nó làm việc này cũng giống như bộ mã hoá đã làm trong quá trình AbS. Sự khác biệt giữa các bộ mã AbS khác nhau ở chỗ: *tín hiệu kích thích của bộ lọc để tổng hợp* ***u(n)*** *được hình thành trong mỗi chúng như thế nào.*



Hình 2.27. Sơ đồ mã hoá/giải mã trên cơ sở AbS

Về mặt lý thuyết ở đầu vào của bộ lọc để tổng hợp phải đưa một số lượng vô cùng lớn các tín hiệu kích thích khác nhau, để quan sát thấy tín hiệu nào sẽ nhận được ở đầu ra của nó và so sánh nó với tín hiệu đem mã hoá. Tín hiệu kích thích nào đưa ra được sai lệch đánh giá giữa tín hiệu tiếng nói đầu vào và tín hiệu tiếng nói được tổng hợp là cực tiểu sẽ được chọn làm kết quả đem mã hoá. Có một sơ đồ vòng để xác định tín hiệu kích thích (hình 2.27) và bảo đảm cho bộ mã AbS có chất lượng tiếng nói đem mã hoá cao, trong khi tốc độ bit thấp.

Vấn đề còn ở chỗ là phải thực hiện một khối lượng lớn các toán tử tính toán, để chọn ra được tín hiệu kích thích tốt nhất. Ngày nay, nhờ khả năng tính toán cực nhanh của các bộ xử lý tín hiệu số (DSP), bài toán này hoàn toàn được giải quyết.

c). Các mã MPE (mã hoá được kích thích nhiều xung)

Như đã nói, khi đưa tín hiệu tiếng nói qua bộ lọc để dự đoán, thì sự tương quan giữa các mẫu lân cận nhau được giảm đáng kể. Tuy nhiên, đối với các nguyên âm (hữu thanh) sự có mặt của các thành phần giọng nói dẫn đến làm xuất hiện trong tín hiệu tiếng nói sự giả chu kỳ và sự tương quan dài hạn cao. Sự giả chu kỳ này không bị loại bỏ bởi bộ dự đoán tuyến tính và dẫn đến làm xuất hiện trong *tín hiệu sai lệch dự đoán* ở chỗ kết nối có biên độ cao.

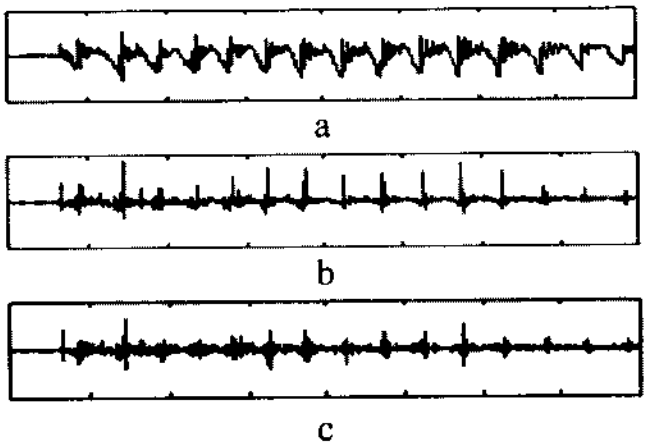
Để loại bỏ tương quan dài hạn, có thể đưa tín hiệu sai lệch dự đoán qua bộ dự đoán tuyến tính thứ hai. Bộ dự đoán tuyến tính này cần phải loại bỏ tương quan không phải giữa các mẫu lân cận

của tín hiệu tiếng nói, mà là giữa các chu kỳ lân cận của sai lệch dự đoán. Điều này đạt được bằng qui định giữ chậm thời gian trong bộ dự đoán theo trị số chu kỳ âm cơ bản của tín hiệu tiếng nói với trị số:

$$p(z) = 1 - \sum_i \beta_i . z^{-(M+i)}$$

ở đây M- chu kỳ của âm cơ bản.

Trong hình 2.28 biểu diễn: a). tín hiệu tiếng nói ban đầu, b). tín hiệu sai lệch của bộ dự đoán tuyến tính ngắn hạn (được mở rộng ra 3 lần), c). tín hiệu ở đầu ra của bộ dự đoán hai tầng (ngắn hạn + dài hạn) (được mở rộng ra 10 lần).



Hình 2.28. Các tín hiệu sai lệch dự đoán.

Nếu bây giờ lấy kết quả tín hiệu sai lệch dự đoán làm tín hiệu kích thích cho bộ lọc dự đoán ngắn hạn và dài hạn xâu chuỗi hai tầng, thì ở đầu ra của nó chúng ta nhận được tín hiệu tiếng nói ban đầu không bị méo. Như vậy, đã có thể mã hoá và truyền theo kênh tín hiệu sai lệch dự đoán nhận được và điều này cũng bảo đảm tính kinh tế vì đã giảm nhỏ biên độ của nó so với tín hiệu tiếng nói ban đầu. Tuy nhiên, dạng tín hiệu trong hình 2.28c) còn khá phức tạp, để mã hoá vẫn đòi hỏi phải sử dụng nhiều bit.

Trong các mã MPE, sử dụng làm tín hiệu kích thích u(n) không phải là tín hiệu sai lệch dự đoán (hình 2.28c), mà đơn giản là *chuỗi nối tiếp ngắn từ bốn đến sáu xung*. Vị trí thời gian và biên độ của chuỗi xung được xác định trong quá trình của thủ tục *phân tích qua tổng hợp* (AbS), cho đến khi nhận được sai lệch cực tiểu giữa tín hiệu tiếng nói ban đầu và tín hiệu tiếng nói tổng hợp. Các tham số của các xung kích thích, để cực tiểu hoá sai lệch, được chọn liên tiếp nhau, bắt đầu từ xung thứ nhất, rồi đến xung thứ hai, và v.v...

Trong thực tiễn qui định vị trí của các xung(sai lệch ví trí $\Delta t_n = |t_n - \hat{t}_n|$) phải đủ chính xác vào khoảng 1μsec, độ chính xác biên độ (sai lệch biên độ $\Delta u_n = |s_n - \hat{s}_n|$) đến 5% (tức 1/20 của biên độ), điều này sẽ bảo đảm tốt chất lượng tiếng nói tổng hợp khi tốc độ bit khoảng 10

Kbit/sec. Đối với mỗi đoạn tín hiệu tiếng nói dài 20 msec sẽ sử dụng 6 xung kích thích, ví trí mỗi xung với độ chính xác 1 msec =1/20 độ dài của đoạn = 5 bit/xung; biên độ xung với độ chính xác đến 5% = 5 bit/ xung; kết quả nhận được tốc độ mã cực tiểu của tín hiệu kích thích = 6 × 10 = 60 bit/20 msec. Ngoài ra, cần phải bổ sung vào mã các tham số của bộ lọc dự đoán ngắn hạn và dài hạn đối với mỗi đoạn đã cho, chẳng hạn 80 - 100 bit/20 msec, kết quả nhận được tốc độ bit 160bit/20 msec = 8Kbit/sec.

d). Các mã RPE (mã hoá được kích thích các xung đều đặn)

Cũng như các mã MPE, các mã RPE (Coding Excited Pulse Regular) sử dụng bộ cố định các xung ngắn làm tín hiệu kích thích u(n). Tuy nhiên, trong bộ mã này các xung được phân bố đều đặn, giữa xung này và xung kia, trên những khoảng cách như nhau và chỉ cần thiết xác định vị trí của xung đầu tiên va biên độ của tất cả các xung. Như vậy, chỉ cần truyền cho bộ giải mã thông tin tối thiểu về vị trí của xung, hơn nữa, trong tín hiệu kích thích có thể bao gồm số lượng lớn các xung, từ đó nhận được sự gần đúng tốt nhất giữa tín hiệu tổng hợp và tín hiệu ban đầu. Chẳng hạn, nếu tốc độ bit 10Kbit/sec, thì bộ mã MPE sử dụng tín hiệu 4 xung kích thích, trong khi đó bộ mã RPE sử dụng tín hiệu 10 xung kích thích, đồng thời nhận được chất lượng tiếng nói cao hơn.

Phương pháp kích thích xung đều đặn RPE, ngày này được sử dụng rộng rãi, chẳng hạn, trong hệ thống thông tin **GSM.**

e). Các mã CELP (Bộ mã được kích thích dự đoán tuyến tính)

Các phương pháp mã hoá tín hiệu tiếng nói MPE và RPE bảo đảm tốt chất lượng tiếng nói với tốc độ bit khoảng 10Kbit/sec và cao hơn, nhưng bắt đầu méo tín hiệu mạnh khi tốc độ bit thấp hơn. Vấn đề ở chỗ để miêu tả được các tham số cần thiết của tín hiệu kích thích như vị trí thời gian, biên độ các xung với độ chính xác yêu cầu cần một số bit lớn.

Liên quan đến điều này đã đưa ra phương pháp để sử dụng làm tín hiệu kích thích không phải là chuỗi xung nối tiếp với bộ các tham số đã cho, *mà là thư viện (quyển sách mã) dạng được chuẩn bị đặc biệt và miêu tả sẵn lưu trữ trong thiết bị nhớ làm tín hiệu kích thích với các dạng khác nhau- Codebook Excited Linear Prediction* (**CELP**). Thuật toán CELP được xây dựng bởi M.K.Schroeder và B.S.Atal vào năm 1984, dựa trên ba ý tưởng sau:

1) Sử dụng mô hình dự báo tuyến tính (LP) làm mô hình của thanh quản;
2) Sử dụng các quyển sách mã thích nghi (adaptive) và cố định (fixed) như là bộ kích thích xung đầu vào của mô hình LP;
3) Thực hiện vòng phản hồi kín để đánh giá và thu hẹp sai lệch trong xử lý.

α). Giải mã của thuật toán CELP

Trước khi nghiên cứu mã hóa thuật toán CELP, chúng ta hãy bắt đầu bằng việc nghiên cứu giải mã thuật toán CELP. Trên hình 2.29 miêu tả sơ đồ khối bộ giải mã CELP tổng quát.

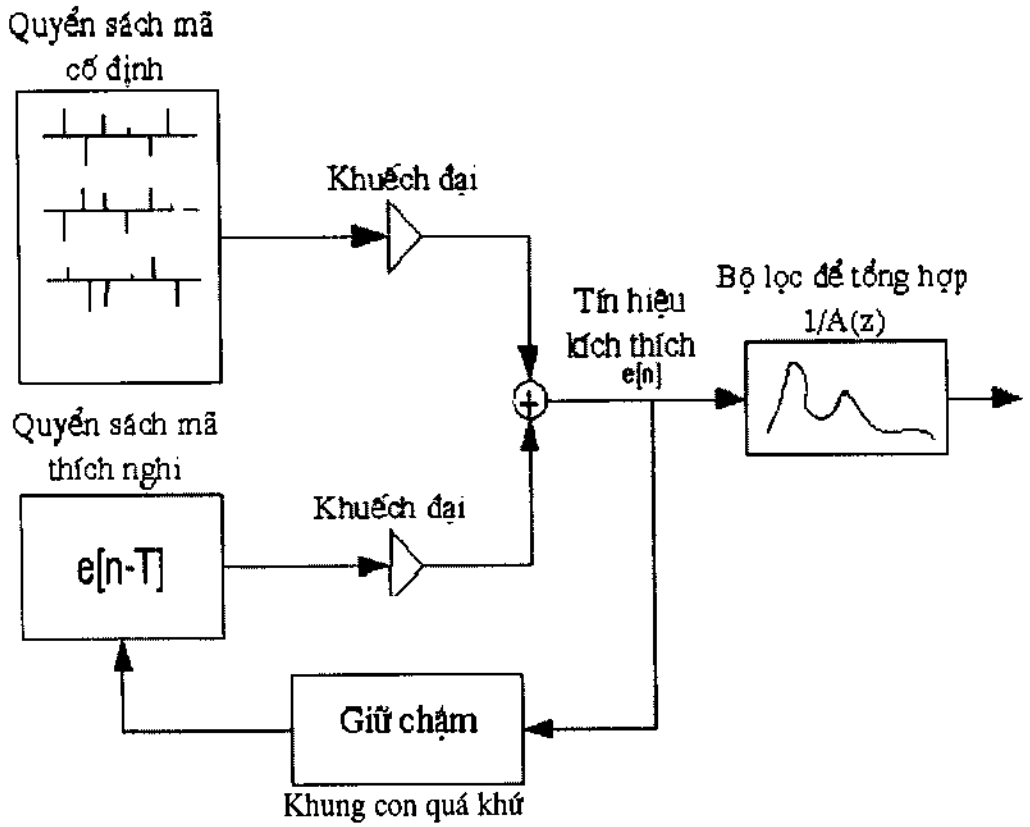
Tín hiệu kích thích là tổng của hai tín hiệu được sinh ra từ quyển sách mã thích nghi và quyển sách mã cố định:

$$e[n] = e_a[n] + e_f[n]$$

ở đây $e_a[n]$ là thành phần tín hiệu của quyển sách mã thích nghi, còn $e_f[n]$ là thành phần tín hiệu của quyển sách mã cố định.

Bộ lọc tổng hợp là bộ lọc toàn cực có hàm truyền dạng 1/A(z), trong đó A(z) là hàm truyền của bộ lọc dự đoán tuyến tính xây dựng theo thuật toán *Levinson-Durbin*. Nó khá phù hợp với mô hình thanh quản và dễ tính toán.

Các mã CELP chọn mỗi khung dài 20 msec, bao gồm 160 mẫu (mỗi mẫu bao gồm 10 xung). Mỗi khung được chia thành 4 khung con, bao gồm 40 mẫu và được mã hóa liên tiếp. Trong hầu hết các mô hình chỉ có các tham số bộ lọc để tổng hợp và hệ số khuếch đại là được mã hóa trong khung chính, các tham số còn lại được mã hóa trong các khung con.



Hình 2.29. Sơ đồ bộ giải mã CELP tổng quát.

Một phương án của CELP sử dụng dự đoán ba bước. Vì thế thành phần tín hiệu đầu ra của quyển sách mã thích nghi được xác định:

$$e_a[n] = g_0 e[n-T-1] + g_1 e[n-T] + g_2 e[n-T+1]$$

trong đó: g_0, g_1 và g_2 là các hệ số khuếch đại bước; T- thời gian giữ chậm.

β). Mã hóa của thuật toán CELP

Nguyên lý chính của thuật toán CELP được gọi là *phân tích bằng tổng hợp (AbS)* có nghĩa là quá trình mã hóa được hình thành như quá trình *phân tích* để xác định các tham số tối ưu rồi truyền cho quá trình giải mã.

Để bảo đảm quá trình mã hóa theo thời gian thực, thuật toán đã sử dụng các nguồn tính toán hữu hạn. Một phương án của CELP sử dụng 4 bước sau:

1) Phép phân tích dự đoán tuyến tính được sử dụng để xác định các tham số bộ lọc tổng hợp;

2) Hệ số khuếch đại và đầu vào của quyển sách mã thích nghi được xác định tối ưu nhờ thủ tục AbS;

3) Hệ số khuếch đại của quyển sách mã cố định được xác định theo kiểu mạch vòng hở căn cứ vào biên độ của tín hiệu kích thích;

4) Đầu vào của quyển sách mã cố định được xác định tối ưu nhờ thủ tục AbS.

Sai lệch giữa tín hiệu đầu ra bộ lọc tổng hợp với mẫu hiện thời của tín hiệu tiếng nói được định nghĩa bằng *hàm trọng số*:

$$W(z) = \frac{A(z/\gamma_1)}{A(z/\gamma_2)}$$

ở đây: A(z)- hàm truyền của bộ lọc dự đoán tuyến tính; còn γ_1 và γ_2 tham số điều khiển hình dạng của bộ lọc bằng cách dịch chuyển điểm cực và điểm không về tâm vòng tròn đơn vị của biến đổi z. Vòng điều khiển kín sẽ đưa hàm trọng số về cực tiểu.

Từ 160 mẫu được chuẩn bị sẵn trong vòng 20 msec đem so sánh với 16 mẫu tín hiệu tiếng nói (tín hiệu tiếng nói được lấy mẫu 10 msec), để phân tích tìm ra các tham số tối ưu dưới đây.

Khi đó kết quả đem mã hoá không phải là các tham số của các xung kích thích, mà là:

- Chỉ số các mẫu tín hiệu kích thích của quyển sách mã thích nghi và cố định;
- Các hệ số khuếch đại;
- Các hệ số của bộ lọc dự đoán tuyến tính.

Đối với CELP các thủ tục tính toán đòi hỏi một khối lượng rất lớn các phép toán, bởi vậy để CELP trở thành hiện thực chỉ trong những năm gần đây, nhờ các bộ xử lý tín hiệu số chuyên dụng (DSP) với tốc độ tính toán khoảng 300 triệu lệnh/sec (tần số xung clock hơn 1,2GHz) và hơn.

Mã hoá tín hiệu tiếng nói trên cơ sở của thuật toán CELP được sử dụng trong hệ thống truyền tin hiện đại đạt tốc độ bit từ 16 đến 4,8 Kbit/sec. Khi tốc độ bit 16 Kbit/sec CELP cho chất lượng tiếng nói như PCM 64 Kbit/sec, còn với tốc độ bit 4,8 Kbit/sec như RPE - GSM 13 Kbit/sec.

Ngày nay, tồn tại các phiên bản cải tiến, nâng cấp trên cơ sở CELP như EVRC (Enhanced Variable Rate Codec - Bộ mã hoá tốc độ thay đổi nâng cao), ARM (Adaptive Multi Rate - Bộ mã hoá nhiều tốc độ thích nghi), ACELP (Algebraic CELP - Quyển sách mã được kích thích dự báo tuyến tính theo đại số). Các bộ mã này được sử dụng hiệu quả trong thông tin di động thế hệ 3. Các mã trên cơ sở ACELP cũng được sử dụng trong công nghệ *VoIP(Voice over Internet Protocol)* – phương pháp truyền tín hiệu tiếng nói theo *gói*.

Sau đây, chúng ta dẫn ra bảng đánh giá chất lượng tiếng nói, (bảng 19), khi mã hóa/giải mã theo các phương pháp khác nhau, theo nguyên tắc điểm trung bình của số người được hỏi ý kiến. Điểm chất lượng cho theo thang 5. Năm điểm là chất lượng lý tưởng, một điểm là chất lượng cực tồi.

LP-CELP: Low delay- CELP - phương pháp CELP độ trễ thấp ;

CS-ACELP: Conjugate Structure – ACELP - phương pháp ACELP cấu trúc ghép đôi;

MP-MLQ: multipulse-multlevel quatization – lượng tử hóa đa xung - đa mức.

Bảng 19. Điểm đánh giá chất lượng tiếng nói của ITU-T CODEC

Phương pháp nén	Tốc độ bit R (Kbit/sec)	Kích thước mẫu Ts (msec)	Điểm trung bình
G.711 PCM	64	0,125	4,1
G.726 ADPCM	32	0,125	3,85
G.728 LD-CELP	15	0,625	3,61
G.729 CS-ACELP	8	10	3,92
G.729a CS-ACELP	8	10	3,70
G.723.1 MP-MLQ	6,3	30	3,5
G.723.1 ACELP	5,3	30	3,61

CHƯƠNG 3

CÁC MÃ CHỐNG NHIỄU

(Các mã kênh)

1. KHÁI NIỆM CHUNG

Trong chương 2, chúng ta đã nghiên cứu các *mã kinh tế* (cũng gọi là các *mã hoá nguồn*), mục đích của chúng là biến đổi các tin tức nguồn để tạo ra các bản mã có độ dư thừa tối thiểu (hay ở dạng nén chặt) được sử dụng khi lưu giữ, cũng như khi truyền tin. Trong chương này, chúng ta sẽ nghiên cứu các *mã chống nhiễu* (hay còn gọi là các *mã kênh*).

Như đã chứng minh trong chương 1, khi truyền tin theo *kênh liên lạc có nhiễu*, thì *có một phần lượng thông tin* sẽ *bị mất mát* trên kênh, vì thế trong số liệu thu sẽ *phát sinh lỗi*. Nếu số lượng lỗi như thế có giá trị nhỏ, hoặc chúng phát sinh đủ thưa thớt, thì thông tin có thể được sử dụng. Tuy nhiên, khi số lượng lỗi lớn, thông tin nhận được không đủ độ tin cậy để sử dụng. Để *giảm nhỏ số lượng lỗi* phát sinh khi truyền thông tin qua kênh có nhiễu, người ta sử dụng mã chống nhiễu (hay còn gọi mã kênh). Mã chống nhiễu phải có hai chức năng: *phát hiện lỗi* và mạnh hơn là *sửa lỗi* khi giải mã.

Tồn tại hai kỹ thuật sửa lỗi chính sau đây:

- Yêu cầu lặp lại tự động (Automatic Repeat Request - ARQ);
- Sửa lỗi phía trước (Forward Error Correction - FEC).

Chúng ta sẽ nghiên cứu các kỹ thuật này ở các mục sau.

Khả năng sử dụng mã hoá để giảm nhỏ số lượng lỗi khi truyền tin qua kênh có nhiễu đã được chứng minh bằng lý thuyết bởi C.Shannon. Vào năm 1948 trong bài báo "Lý thuyết toán học của truyền tin", ông đã khẳng định *nếu tốc độ sản sinh tin (năng suất) của nguồn không vượt khỏi khả năng thông qua của kênh truyền, thì bằng mã hoá/giải mã phù hợp có thể giảm xác suất lỗi trên kênh đến không.*

Tuy nhiên, phải thấy rằng thực tiễn hạn chế tốc độ truyền tin không phải là khả năng thông qua của kênh truyền, mà chính là độ phức tạp của sơ đồ mã hoá và giải mã. Vì vậy, mọi cố gắng trong nghiên cứu, chế tạo trong hai mươi năm trở lại đây là tìm kiếm các bộ mã hiệu quả có khả năng thực hiện trong thực tiễn các sơ đồ mã hoá và giải mã, mà chúng theo các đặc tính của mình tiến gần đến dự đoán lý thuyết.

2. CÁC NGUYÊN LÝ CƠ BẢN XÂY DỰNG MÃ CHỐNG NHIỄU. PHÂN LOẠI MÃ CHỐNG NHIỄU

Mã hoá với chức năng phát hiện lỗi và sửa lỗi là *phương pháp gia công tin tức để nâng cao độ tin cậy của việc truyền tin theo kênh số*. Mặc dù các sơ đồ mã hoá rất khác nhau và dựa trên các lý thuyết toán học khác nhau, nhưng tất cả chúng đều dựa trên *hai nguyên lý chung*:

Thứ nhất - *sử dụng độ dư thừa*. Chuỗi ký hiệu mã hoá luôn luôn chứa đựng các ký hiệu bổ sung, hay dư thừa, tức là *số lượng các ký hiệu trong chuỗi mã V luôn luôn lớn hơn số lượng ký hiệu mang thông tin chứa trong nó*.

Thứ hai - *đặc tính trung bình*, có nghĩa là các ký hiệu dư thừa không phát sinh ngẫu nhiên, mà phụ thuộc vào một vài ký hiệu mang thông tin. Như vậy, thông tin chứa đựng trong chuỗi mã **V** được phân phối lại cả trong các ký hiệu dư thừa.

Hiện tồn tại hai lớp mã - *mã khối và mã xoắn (liên tục)*. Sự phân biệt giữa chúng thường căn cứ vào có hay không có *bộ nhớ* trong sơ đồ mã hoá.

Mã khối được hình thành như sau: từ chuỗi bit **X** của tin tức (ở dạng nhị phân) có độ dài bất kỳ (nhưng hữu hạn), được phân chia ra thành các khối, mỗi khối bao gồm k bit (ký hiệu) thông tin; ở khối cuối cùng, nếu không đủ k bit thông tin, nó sẽ được bổ sung thêm một số ký hiệu dạng 1 0 0..., để đủ trở thành khối có k ký hiệu, sau khi giải mã các ký hiệu nay bị loại bỏ đi.

Quá trình mã hoá sẽ biến đổi khối có k bit thông tin thành ra khối có n ký hiệu mã, được gọi là một *từ mã* và mã khối được ký hiệu M(n, k). Trong mỗi khối số ký hiệu dư thừa ký hiệu là r và được xác định $r = n - k$, hoặc là $n = k + r$. *Chức năng của các dấu dư thừa là để bảo đảm khả năng phát hiện lỗi và mạnh hơn để sửa lỗi* trong quá trình giải mã. Rõ ràng, số lượng các ký hiệu dư thừa r càng nhiều thì khả năng phát hiện lỗi và sửa lỗi càng mạnh, tuy nhiên hiệu quả truyền tin của khối càng kém. Vì vậy, một xu hướng quan trọng của mã chống nhiễu là tìm kiếm các bộ mã có khả năng sửa được nhiều lỗi nhất với ít ký hiệu dư thừa nhất, được gọi là bộ mã hiệu quả.

Như vậy, từ k bit thông tin của mỗi khối hình thành nên 2^k khả năng, hay 2^k từ mã ở đầu ra của bộ mã hoá, mỗi từ mã dài $n = k + r$ ký hiệu.

Thuật ngữ "*không có nhớ*" chỉ ra rằng trong *mỗi khối dài n ký hiệu mã, thì các ký hiệu dư thừa r của khối chỉ phụ thuộc vào chính k bit thông tin trong khối mã đó, mà không phụ thuộc vào các bit thông tin của các khối mã nào khác*.

Tuỳ theo thuật toán mã hoá để phát sinh các ký hiệu dư thừa r từ k bit thông tin, mà chúng ta có các loại mã khối khác nhau. Chúng ta có thể kể ra các loại mã khối theo trình tự từ đơn giản đến phức tạp như sau:

- Mã khối tuyến tính;
- Mã Cyclic (CRC);
- Mã Bose- Chaudhuri- Hocquenghem (BCH);
- Mã Reed- Solomon (RS).

Khi phân tích mã khối thường cần phải sử dụng một vài định nghĩa sau:

- *Trọng số Hamming* của một từ mã nhị phân α_i, thường ký hiệu $w(\alpha_i)$, là số lượng ký hiệu 1 có trong từ mã đó. Chẳng hạn, chúng ta có từ mã α_i = 1000101, trọng số Hamming của từ mã được xác định $w(\alpha_i) = 3$.

- *Khoảng cách Hamming* giữa hai từ mã α_i, α_k (có độ dài đều), thường ký hiệu $d(\alpha_i,\alpha_k)$, là số lượng vị trí tương ứng mà tại đó chúng có giá trị khác nhau. Chẳng hạn, α_i =1010011, α_k =1000101, khoảng cách Hamming giữa hai từ mã được xác định $d(\alpha_i,\alpha_k) = 3$. Từ đây, chúng ta có $d(\alpha_i,\alpha_k) = w(\alpha_i \oplus \alpha_k)$, với ký hiệu $\oplus$ chỉ phép cộng theo modulo 2 (mod2).

- *Khoảng cách Hamming cực tiểu của bộ mã* (chỉ áp dụng cho bộ mã đều), thường ký hiệu d_{min}, hoặc d_0, được xác định $d_0 = \min d(\alpha_i,\alpha_k)$ với mọi i,k. Tham số d_0 của bộ mã quyết định khả năng phát hiện lỗi và sửa lỗi:

+ *Số lỗi có thể phát hiện được khi giải mã*, ký hiệu là e và được xác định $e = d_0 - 1$. Một bộ mã có $d_0 = 1$, không có khả năng phát hiện bất kỳ lỗi nào, e = 0, được gọi là *bộ mã đầy*, hay bộ mã không có dư thừa, $g_C = 0$. Một bộ mã muốn phát hiện được một lỗi, e = 1, cần $d_0 = 2$. Các bộ mã có $d_0 \geq 2$ được gọi là *bộ mã vơi*, hay bộ mã có dư thừa $g_C > 0$ ($g_C = 1 - \frac{H(X)}{\bar{n}} = 1 - \frac{\log_2 2^k}{n} = 1 - \frac{k}{k+r} > 0$, nếu r = 0 thì $g_C = 0$).

+ *Số lỗi có thể sửa được khi giải mã*, ký hiệu là t và được xác định $t \leq \left[\frac{d_0 - 1}{2}\right]$, trong đó ký hiệu [x] để chỉ phép lấy phần nguyên của x. Từ công thức này xác định được một bộ mã muốn sửa được t lỗi cần $d_0 \geq 2t + 1$. Chẳng hạn, bộ mã muốn sửa được 1 lỗi, t = 1, cần $d_0 \geq 3$ (trong trường hợp này có thể phát hiện được 2 lỗi).

Trong các loại mã khối có thể là mã khối trong trường nhị phân **GF(2)**, cũng có thể là mã khối trong trường **GF(2^m)**.

Mã xoắn (hay mã liên tục) làm việc với chuỗi các bit thông tin không được phân ra thành khối, hay nói một cách khác quá trình mã hoá và quá trình giải mã diễn ra liên tục theo thời gian từ khi bắt đầu có tin tức cho đến khi kết thúc tin tức. Trong chuỗi các ký hiệu mã đầu ra bộ mã hoá, nói chung không phân biệt được đâu là các bit thông tin, đâu là các ký hiệu dư thừa (với các mã xoắn hệ thống có thể phân biệt được đâu là bit thông tin, đâu là dấu dư thừa), vì vậy mã xoắn còn có tên gọi là **mã chập**. Quá trình mã hoá của mã xoắn có thể cho rằng bao gồm nhiều chu kỳ mã hoá. Ở mỗi chu kỳ mã hoá từ k bit thông tin đầu vào hình thành nên n ký hiệu mã đầu ra, hơn nữa các ký hiệu mã đầu ra không chỉ phụ thuộc vào k bit thông tin đầu vào, mà còn phụ thuộc vào m bit thông tin ở các thời điểm trước đó, vì trong bộ mã hoá có sử dụng m ô nhớ (ô ghi dịch). Tuy nhiên, phải nhớ rằng các bit thông tin đi vào, cũng như các ký hiệu mã đi ra là

liên tục, không tồn tại quá trình phân thành khối riêng biệt, vì vậy, có tên **mã liên tục**. Mã xoắn là loại mã có nhớ. Mã xoắn thường được ký hiệu X(n, k, m).

Mã xoắn có thể phân thành hai lớp: mã xoắn hệ thống và mã xoắn không hệ thống. Mã xoắn hệ thống điển hình nhất được sử dụng trong thực tiễn là mã Turbo (thực chất Turbo là mã khối được xây dựng từ hai mã xoắn hệ thống), mã TCM (Trellis Coded Modulation- điều chế có mã hoá giàn mắt cáo).

Mã khối được sử dụng trong thực tiễn kỹ thuật, khi mà số liệu một cách tự nhiên có thể phân chia được thành các khối. Còn mã xoắn được sử dụng trong thực tiễn kỹ thuật, khi mà số liệu một cách tự nhiên được truyền ở dạng liên tục sẽ thuận tiện hơn. Hơn nữa, với cùng độ dư thừa, mã xoắn có khả năng sửa lỗi tốt hơn.

3. CÁC MÃ KHỐI TUYẾN TÍNH

Với mã khối, mỗi khối có k bit thông tin, vì vậy hình thành nên 2^k từ mã, mỗi từ mã dài n ký hiệu mã. Các ký hiệu dư thừa r phụ thuộc vào k bit thông tin sao cho đơn giản nhất, để có khả năng thực hiện trong thực tiễn kỹ thuật. Mã khối tuyến tính là loại mã khối đơn giản nhất trong các mã khối. Tại sao có tên là *mã khối tuyến tính*?

Mỗi khối dài n ký hiệu, bao gồm 2^k từ mã, được gọi là mã khối tuyến tính M(n,k) với điều kiện là tất cả 2^k từ mã hình thành nên không gian con bậc k của không gian véc tơ bậc n trong trường nhị phân **GF(2)**.

Nói một cách đơn giản là *mã nhị phân được gọi là tuyến tính, nếu tổng theo modul2 (mod2) của hai từ mã bất kỳ cũng là từ mã của bộ mã đó.*

Chúng ta định nghĩa một vài khái niệm cơ bản về trường nhị phân **GF(2)**: *Trường* là tập hợp của các đối tượng toán học, chúng có thể cộng, trừ, nhân và chia.

Trường đơn giản nhất là trường nhị phân, vì trường chỉ bao gồm có hai phần tử là 0 và 1. Các toán tử cộng và nhân với các phần tử này, được định nghĩa như sau:

$$0 \oplus 0 = 0 \qquad 0 . 0 = 0$$

$$0 \oplus 1 = 1 \qquad 0 . 1 = 0$$

$$1 \oplus 0 = 1 \qquad 1 . 0 = 0$$

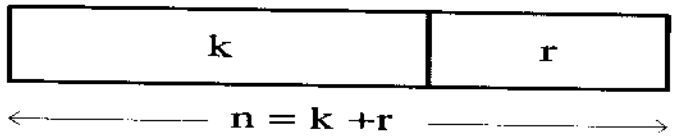
$$1 \oplus 1 = 0 \qquad 1 . 1 = 1$$

Các phép cộng và nhân được định nghĩa như thế này được gọi là *phép cộng* và *phép nhân theo modul2* **(mod2)**.

Từ các định nghĩa này, chúng ta thấy rằng $1 \oplus 1 = 0$, dẫn đến $-1 = 1$ và $1 \oplus 1 = 1 - 1$, tức là trong trường nhị phân phép cộng và phép trừ là như nhau, đều là phép cộng mod2; với phép nhân ta có $1 . 1 = 1$ suy ra $1 : 1 = 1$.

Tập hợp từ hai phần tử 0 và 1, với phép cộng và phép nhân theo mod2, được gọi là trường nhị phân và ký hiệu là **GF(2)**. Đối với trường **GF(2)** có thể áp dụng mọi toán tử đại số tuyến tính, trong đó có toán tử ma trận.

Mọi mã khối tuyến tính đều là *mã hệ thống*. Mã khối hệ thống là mã khối mà với mỗi khối bao gồm k bit thông tin và r ký hiệu dư thừa, tạo thành hai khối con tách biệt nhau, như hình 3.1.



Hình 3.1. Một khối mã (từ mã) của mã khối hệ thống.

Sau đây chúng ta nghiên cứu các mã khối tuyến tính hệ thống đơn giản nhất.

3.1. Mã dựa trên kiểm tra chẵn (hoặc lẻ)

Mã khối tuyến tính đơn giản nhất là mã dựa trên kiểm tra chẵn (hoặc lẻ), có dạng M(n, n–1), tức là chỉ có 1 dấu dư, chính là dấu kiểm tra chẵn (hoặc lẻ). Sau đây, chúng ta chỉ nghiên cứu các mã dựa trên sự kiểm tra chẵn, chẳng hạn, mã khối M(4, 3), có thể viết dưới dạng *ma trận dòng* như sau:

$$\mathbf{V} = [m_0\ m_1\ m_2\ m_0 \oplus m_1 \oplus m_2] \qquad (3.1)$$

trong đó m_i- các bit thông tin, nhận các giá trị 0 hoặc 1, ký hiệu thứ 4 là ký hiệu kiểm tra chẵn.

Chúng ta hãy giải thích ý tưởng về kiểm tra chẵn. Giả thử, chúng ta có ma trận dòng của các bit thông tin dạng:

$$\mathbf{I} = [101]$$

Chú ý trong công thức trên số lượng các ký hiệu 1 là số chẵn - 2,

Khi đó, theo công thức (3.1) ký hiệu thứ 4 được xác định:

$$m_3 = m_0 \oplus m_1 \oplus m_2 = 1 \oplus 0 \oplus 1 = 0$$

Kết quả chúng ta nhận được:

$$\mathbf{V} = [m_0\ m_1\ m_2\ m_3] = [1010] \qquad (3.2)$$

Chú ý trong công thức (3.2) số lượng các ký hiệu 1 cũng là số chẵn - 2.

Một thí dụ khác, giả thử, chúng ta có ma trận dòng của các bit thông tin dạng:

$$\mathbf{I} = [100]$$

Khi đó từ mã **V** được xác định:

$$\mathbf{V} = [m_0\ m_1\ m_2\ m_3] = [1001]$$

Chú ý trong công thức trên số lượng các ký hiệu 1 cũng là số chẵn - 2.

Dễ dàng nhận ra rằng khi trong **I** số lượng ký hiệu 1 là *bất kỳ* (chẵn, hoặc lẻ) thì trong **V** số lượng ký hiệu 1 *luôn luôn là số chẵn*. Vì thế ký hiệu thứ 4 được gọi là ký hiệu *kiểm tra chẵn*.

Từ kết quả (3.2) và công thức (3.1), căn cứ theo qui tắc nhân ma trận, chúng ta có thể xây dựng công thức tổng quát hơn như sau:

$$\boldsymbol{V} = \boldsymbol{I} \times \boldsymbol{G}_{(4,3)} \tag{3.3}$$

Trong đó:

$$G_{(4,3)} = \begin{bmatrix} 1001 \\ 0101 \\ 0011 \end{bmatrix} \tag{3.4}$$

$\boldsymbol{G}_{(4,3)}$- được gọi là *ma trận sinh*

Hơn nữa, ma trận sinh $\boldsymbol{G}_{(4,3)}$ có thể tách thành 2 ma trận, ma trận thứ nhất là ma trận đơn vị, ký hiệu là $\boldsymbol{E}_{(3,3)}$, có dạng:

$$E_{(3,3)} = \begin{bmatrix} 100 \\ 010 \\ 001 \end{bmatrix} \tag{3.5}$$

$\boldsymbol{E}_{(3,3)}$ - ma trận đơn vị bậc 3×3

Ma trận thứ 2, ký hiệu $\boldsymbol{R}_{(1,3)}$, là ma trận cột, với 1 cột và 3 dòng, tất cả đều là ký hiệu 1.

$$R_{(1,3)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \text{- ma trận bậc } 1\times 3 \tag{3.6}$$

Chúng ta có thể biểu diễn lại ma trận $\boldsymbol{G}_{(4,3)}$ theo dạng:

$$G_{(4,3)} = \left[E_{(3,3)}; R_{(1,3)}\right] \tag{3.7}$$

Độc giả hãy chú ý đến dạng biểu diễn (3.7), đặc biệt các chỉ số của hai ma trận thành phần. Rõ ràng $\boldsymbol{E}_{(3,3)} = \boldsymbol{E}_{(k,k)}$, vì k =3 là số lượng các bit thông tin; còn $\boldsymbol{R}_{(1,3)} = \boldsymbol{R}_{(r,k)}$, với r = 1 là số dấu kiểm tra. Chúng ta viết lại công thức (3.7) cho tổng quát hơn như sau:

$$G_{(n,k)} = \left[E_{(k,k)}; R_{(r,k)}\right] \tag{3.7a}$$

Dạng ma trận sinh theo công thức (3.7a) được sử dụng cho mọi mã khối tuyến tính.

Thuật toán xác định theo công thức (3.3), gọi là *thuật toán mã hoá.*

Khi truyền theo kênh liên lạc có nhiễu, tại điểm thu, chuỗi ký hiệu mã có thể bị mắc lỗi, tức là khác với chuỗi ký hiệu mã đã gửi đi từ phía phát (ký hiệu 0 → 1, và ngược lại).

Nếu các lỗi của các ký hiệu mã xuất hiện với xác suất như nhau và không phụ thuộc lẫn nhau, thì xác suất để xuất hiện một lỗi trong chuỗi n ký hiệu mã, được xác định:

$$P_1 = n.P_e.(1 - P_e)^{n-1} \tag{3.8}$$

(nghĩa là khi một ký hiệu mã bị lỗi, thì (n – 1) ký hiệu mã còn lại không bị mắc lỗi).

Xác suất để xảy ra hai lỗi được xác định bằng số lượng của tổ hợp các khả năng hai lỗi (hai lỗi bất kỳ nào đó trong chuỗi n ký hiệu mã, còn (n–2) các ký hiệu mã còn lại không mắc lỗi):

$$P_2 = C_n^2 . P_e . (1 - P_e)^{n-2} \tag{3.9}$$

và tương tự khi xác định xác suất để xuất hiện nhiều lỗi hơn.

Nếu cho rằng xác suất xuất hiện lỗi P_e trong các ký hiệu mã đủ bé (tức là $P_e << 1$), thì xác suất để xuất hiện m lỗi trong chuỗi n ký hiệu mã được xác định gần đúng:

$$P_m \approx P_e^m \tag{3.10}$$

Chúng ta hãy quay lại với mã kiểm tra chẵn, với dạng cụ thể M(4,3). Khi truyền theo kênh liên lạc có nhiễu, các từ mã dạng (3.2), rõ ràng chúng có khả năng phát hiện ra một lỗi (*chỉ một lỗi thôi*) khi giải mã theo *thuật toán kiểm tra chẵn*. Thực vậy, đem cộng theo mod2 các ký hiệu mã trong từ mã, tức là:

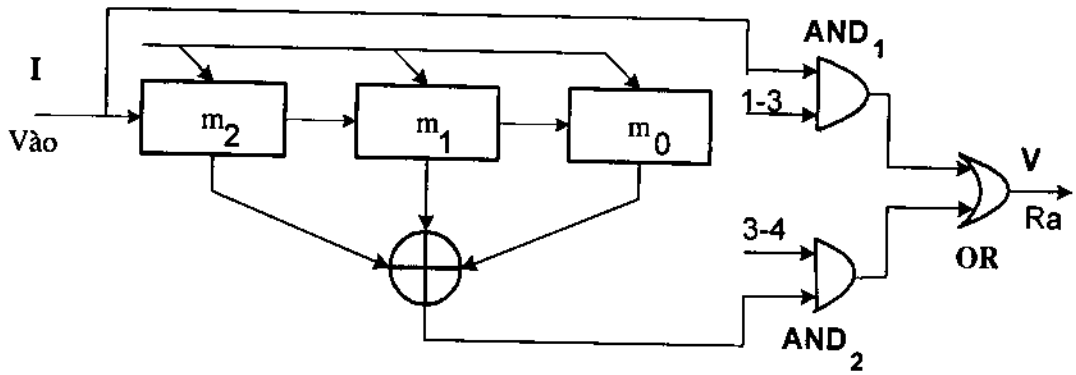
$$\mathbf{S} = m_0 \oplus m_1 \oplus m_2 \oplus m_3 \tag{3.11}$$

Trong đó **S** được gọi là véctơ Syndrome (véctơ hội chứng, trong trường hợp này **S** có một giá trị, vì chỉ có một dấu dư). Nếu **S** = 0, từ mã không mắc lỗi (với điều kiện từ mã thu chỉ có một lỗi). Điều này là rõ ràng, vì số lượng các ký hiệu 1 là số chẵn. Nếu ngược lại, **S** = 1, từ mã mắc một lỗi. Vì khi **S** = 1, chứng tỏ số ký hiệu 1 trong từ mã không còn là số chẵn nữa. Hơn nữa, thuật toán này chỉ phát hiện được một lỗi. Để sửa lỗi, người ta sử dụng *kỹ thuật ARQ*, tức là yêu cầu phía phát gửi lặp lại từ mã bị lỗi, cho đến khi thu đúng.

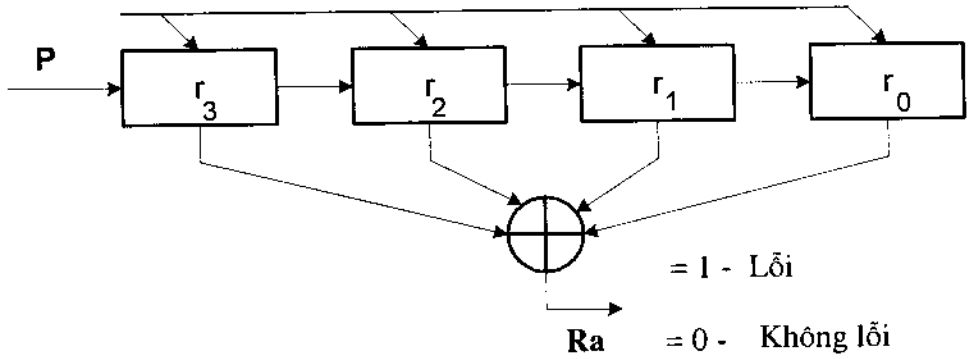
Mạch điện mã hoá và giải mã cho thuật toán kiểm tra chẵn được biểu diễn như hình 3.2.

Mạch điện mã hoá bao gồm:

- Ba ô ghi dịch, để ghi 3 bit thông tin, theo trình tự m_0 dịch vào đầu tiên, còn m_2 dịch vào cuối cùng;
- Bộ cộng mod2 (XOR), để cộng 3 bit thông tin tạo ra ký hiệu kiểm tra;
- Cổng AND_1, mở từ nhịp 1 ÷ 3, để đưa 3 bit thông tin ra ngoài. Cổng AND_2, mở vào nhịp 4, để đưa ký hiệu kiểm tra ra ngoài. Phần tử OR để đưa các bit thông tin và ký hiệu kiểm tra ra ngoài, hình thành nên từ mã **V**;



a). Mạch điện mã hoá



b). Mạch giải mã

Hình 3.2. Mạch điện mã hoá (a), mạch điện giải mã (b) của mã dựa trên kiểm tra chẵn.

Mạch điện giải mã bao gồm:

- Bốn ô ghi dịch, để ghi 4 ký hiệu mã thu nhận được, theo trình tự ký hiệu r_0 dịch vào đầu tiên, còn r_3 dịch vào cuối cùng;
- Bộ cộng mod2 (XOR), để cộng 4 ký hiệu mã (hay để tính véctơ **S**), thực hiện phép kiểm tra chẵn. Nếu **S** = 0, từ mã thu được không có lỗi; ngược lại, nếu **S** = 1, từ mã thu được có lỗi.

Chúng ta cũng nhận xét thấy rằng nếu đem cộng 2 từ mã của cùng bộ mã, thì từ mã thu được cũng là từ mã thuộc bộ mã đang xét. Thực vậy, chẳng hạn với $M_{(4,3)}$, chúng ta có hai từ mã:

$$\mathbf{a} = [a_0, a_1, a_2, a_0 \oplus a_1 \oplus a_2] \text{ và } \mathbf{b} = [b_0, b_1, b_2, b_0 \oplus b_1 \oplus b_2]$$

Chúng ta nhận được:

$$\mathbf{c} = \mathbf{a} \oplus \mathbf{b} = [a_0 \oplus b_0, \oplus a_1 \oplus b_1, \oplus a_2 \oplus b_2, a_0 \oplus a_1 \oplus a_2 \oplus b_0 \oplus b_1 \oplus b_2]$$
$$= [c_0, c_1, c_2, c_3]$$

Ký hiệu mã c_3 cũng đóng vai trò ký hiệu *kiểm tra chẵn* đối với các ký hiệu c_0, c_1, c_2. Do đó từ mã **c** cũng thuộc bộ mã $M_{(4,3)}$ đang xét. Tính chất này, như đã biết, gọi là *tính chất tuyến tính* của bộ mã.

Các mã dựa trên kiểm tra chẵn rất đơn giản và cũng rất hiệu quả. Vì vậy, chúng được sử dụng rộng rãi trong thực tiễn kỹ thuật.

Sau đây, chúng ta sẽ nghiên cứu các mã khối tuyến tính ở dạng chung nhất M(n, k), với số dấu dư thừa r = n – k là bất kỳ.

3.2. Các mã khối tuyến tính dạng M(n, k)

Trong mục trên, chúng ta đã nghiên cứu dạng mã khối tuyến tính hệ thống đơn giản nhất - mã dựa trên kiểm tra chẵn. Với dạng mã này, khi mã hoá, chỉ phát sinh một ký hiệu dư (r = 1) và có khả năng phát hiện một lỗi, khi giải mã. Để sửa lỗi người ta sử dụng phương pháp ARQ (yêu cầu

phát lặp). Cũng trong mục trên, chúng ta cũng đề cập tới khái niệm ma trận sinh $G_{(n,k)}$, cũng như thuật toán mã hoá.

Ở mục này, chúng ta sẽ nghiên cứu mã khối tuyến tính hệ thống dạng M(n, k), với số ký hiệu dư r bất kỳ. Các mã chống nhiễu với số dấu dư r ≥ 2, sẽ sửa lỗi theo phương pháp ***FEC*** (sửa lỗi phía trước)

a). Ma trận sinh $G_{(n,k)}$

Mã khối tuyến tính hệ thống M(n, k) hoàn toàn được xác định nhờ ma trận sinh $G_{(n,k)}$ bậc (n × k) với các phần tử nhị phân. Khi đó mỗi từ mã được xác định bằng tổ hợp tuyến tính của các phần tử ma trận dòng các bit thông tin với các phần tử cột tương ứng của ma trận sinh. Từ công thức suy rộng (3.7a), chúng ta xác định được ma trận sinh $G_{(n,k)}$ có dạng:

$$G_{(n,k)} = \left[E_{(k,k)}; R_{(r,k)}\right] \tag{3.12}$$

Trong đó $E_{(k,k)}$ - ma trận đơn vị bậc (k× k), còn $R_{(r,k)}$ - ma trận bậc (r× k). Khi xây dựng ma trận $R_{(r,k)}$ phải chú đến các đặc điểm sau:

- Số cột của ma trận $R_{(r,k)}$ bằng số ký hiệu dư (ký hiệu kiểm tra) - r;
- Các ký hiệu dư r_j là tổ hợp tuyến tính của các bit thông tin m_i và phải bảo đảm điều kiện:

+ Các tổ hợp khác nhau, phải khác nhau, ít nhất một bit thông tin;

+ Một bit thông tin bất kỳ, phải tham gia ít nhất vào một tổ hợp tuyến tính nào đó.

- Khoảng cách Hamming cực tiểu của bộ mã - d_0 được xác định bằng:

d_0= 1 + (số ký hiệu 1 trong hàng nào đó có ít ký hiệu 1 nhất).

Tức là muốn d_0 của bộ mã cực đại cần số ký hiệu 0 trong hàng bất kỳ là ít nhất có thể.

Sau đây, căn cứ vào các điều kiện trên, chúng ta xây dựng các ma trận sinh của các mã khối tuyến tính đơn giản M(6,3), M(7,4), M(8,4):

$$G_{(6,3)} = \begin{bmatrix} 100101 \\ 010110 \\ 001011 \end{bmatrix}; \quad G_{(7,4)} = \begin{bmatrix} 1000110 \\ 0100011 \\ 0010111 \\ 0001101 \end{bmatrix}; \quad G_{(8,4)} = \begin{bmatrix} 10001101 \\ 01000111 \\ 00101110 \\ 00011011 \end{bmatrix} \tag{3.13}$$

Chú ý rằng trong các ma trận trên, ma trận $R_{(r,k)}$ có nhiều dạng khác nhau (bằng cách hoán vị hoặc dòng, hoặc cột).

d_0 của $G_{(6,3)}$ và $G_{(7,4)}$ = 1 + 2 = 3; bộ mã này có thể phát hiện được 2 lỗi và sửa được 1 lỗi.

d_0 của $G_{(8,4)}$ = 1 + 3 = 4; bộ mã này có thể phát hiện được 3 lỗi và sửa được 1 lỗi.

Các mã có tham số với $m \geq 3$:

$$n = 2^m - 1$$

$$k = n - m$$

$$r = m$$

được gọi là *mã Hamming*. Mã này chỉ sửa được 1 lỗi.

b). Thuật toán mã hoá

Thuật toán mã hoá được xác định bằng công thức (3.3) dạng suy rộng:

$$\mathbf{V} = \mathbf{I} \times \mathbf{G}_{(n,k)} \tag{3.14}$$

Chúng ta sử dụng công thức này cho mã M(7, 4) với ma trận sinh dạng (3.13).

Chẳng hạn, các bít thông tin có dạng:

$$\mathbf{I} = (m_0, m_1, m_2, m_3)$$

Còn từ mã V được xác định dưới dạng:

$$\mathbf{V} = (v_0, v_1, v_2, v_3, v_4, v_5, v_6)$$

Theo công thức của thuật toán mã hoá (3.14), chúng ta có thể viết:

$$(v_0, v_1, v_2, v_3, v_4, v_5, v_6) = (m_0, m_1, m_2, m_3) \times \begin{bmatrix} 1000110 \\ 0100011 \\ 0010111 \\ 0001101 \end{bmatrix}$$

Từ đây, theo qui tắc nhân ma trận, chúng ta xác định được:

$$\begin{aligned} &v_0 = m_0, v_1 = m_1, v_2 = m_2, v_3 = m_3 , \\ &v_4 = m_0 \oplus m_2 \oplus m_3 , \\ &v_5 = m_0 \oplus m_1 \oplus m_2 , \\ &v_6 = m_1 \oplus m_2 \oplus m_3 , \end{aligned} \tag{3.15}$$

Chẳng hạn, các bit thông tin có dạng **I** = (1000), khi đó từ mã **V** được xác định:

$$v_0 = 1, v_1 = 0, v_2 = 0, v_3 = 0 ,$$

$$v_4 = 1 \oplus 0 \oplus 0 = 1 ,$$

$$v_5 = 1 \oplus 0 \oplus 0 = 1 ,$$

$$v_6 = 0 \oplus 0 \oplus 0 = 0 ,$$

Do đó **V** = (1000110).

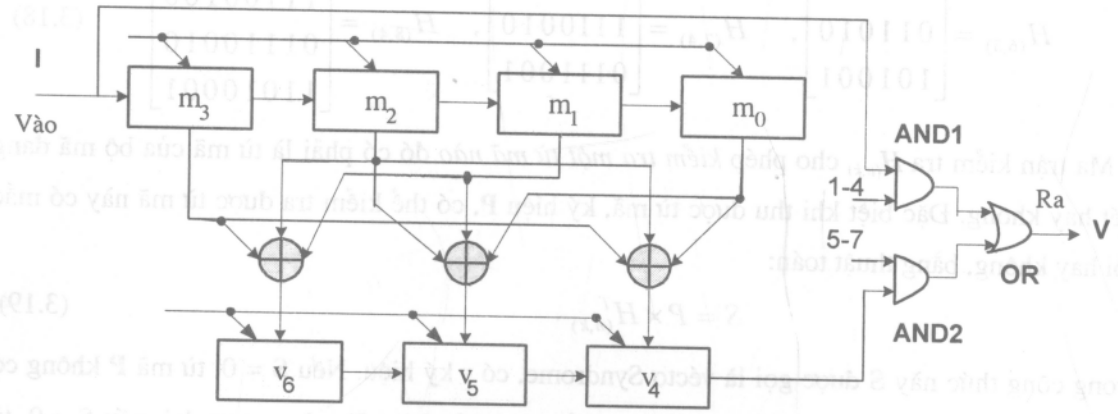
Một thí dụ khác **I** = (1100), một cách tương tự, chúng ta xác định được:

V = (1100101).

Khi k = 4, có tất cả $2^4 = 16$ từ mã, chúng ta có thể xác định chúng tương tự như trên.

c). Mạch điện mã hoá

Mạch điện mã hoá được xây dựng căn cứ vào thuật toán mã hoá. Chúng ta xây dựng mạch điện mã hoá cho một trường hợp cụ thể M(7,4) với ma trận sinh dạng (3.13). Khi xây dựng mạch điện mã hoá, trong trường hợp này, chúng ta căn cứ vào kết quả mã hoá (3.15). Mạch điện có sơ đồ như hình 3.3.



Hình 3.3. Mạch điện mã hoá của mã Hamming M(7, 4).

Mạch điện mã hoá bao gồm các phần tử:

- 4 ô ghi dịch (phía trên), để ghi 4 bit thông tin. Các bit thông tin được dịch vào theo trình tự: bit m_0 dịch vào đầu tiên, còn bit m_3 dịch vào sau cùng;
- 3 bộ cộng mod2 (XOR), để tính 3 ký hiệu dư;
- 3 ô ghi dịch (vẽ dưới), để ghi 3 ký hiệu dư;
- Cổng AND1 mở từ nhịp 1-4 để đưa 4 bít thông tin ra ngoài, hình thành nên 4 bit thông tin của từ mã **V**;
- Cổng AND2 mở từ nhịp 5-7 để đưa 3 ký hiệu dư ra ngoài, hình thành nên 3 ký hiệu kiểm tra của từ mã **V**;
- Phần tử OR làm việc trong 7 nhịp, để kết thúc một quá trình mã hoá, hình thành nên từ mã **V** với 7 ký hiệu.

d). Ma trận kiểm tra $H_{(n,k)}$

Ma trận kiểm tra $H_{(n,k)}$ có đặc tính sau: *tích của mọi từ mã **V** với ma trận kiểm tra $H_{(n,k)}$ chuyển vị, đều cho kết quả bằng 0, tức là:*

$$V \times H^T_{(n,k)} \equiv 0 \qquad (3.16)$$

trong công thức (3.16) ký hiệu T (translation) chỉ phép chuyển vị ma trận (phép thay thế hàng bởi cột).

Với đặc tính (3.16), ma trận kiểm tra $H_{(n,k)}$ được xác định theo ma trận sinh $G_{(n,k)}$ với dạng biểu diễn (3.12) như sau:

$$H_{(n,k)} = \left[R^T_{(r,k)}, E_{(r,r)}\right] \tag{3.17}$$

Chúng ta dẫn ra dưới đây ba ma trận kiểm tra tương ứng với ba ma trận sinh ở công thức (3.13):

$$H_{(6,3)} = \begin{bmatrix} 110100 \\ 011010 \\ 101001 \end{bmatrix}, \quad H_{(7,4)} = \begin{bmatrix} 1011100 \\ 1110010 \\ 0111001 \end{bmatrix}, \quad H_{(8,4)} = \begin{bmatrix} 10111000 \\ 11100100 \\ 01110010 \\ 11010001 \end{bmatrix} \tag{3.18}$$

Ma trận kiểm tra $H_{(n,k)}$ cho phép *kiểm tra một từ mã nào* đó có phải là từ mã của bộ mã đang xét hay không. Đặc biệt khi thu được từ mã, ký hiệu **P**, có thể kiểm tra được từ mã này có mắc lỗi hay không, bằng thuật toán:

$$S = P \times H^T_{(n,k)} \tag{3.19}$$

trong công thức này **S** được gọi là véctơ Syndrome, có r ký hiệu. Nếu **S** = 0, từ mã **P** không có lỗi (với giả thiết số lỗi nằm trong khả năng kiểm soát của bộ mã), còn ngược lại, nếu **S** ≠ 0, từ mã **P** có lỗi. Công thức (3.19) được gọi là *thuật toán giải mã phát hiện sai*. Để *sửa lỗi cần các phân tích bổ sung*, chúng ta sẽ xét sau này.

e). Thuật toán giải mã

Thuật toán giải mã được xác định theo công thức (3.19). Chúng ta hãy sử dụng thuật toán này cho mã M(7, 4) với ma trận sinh dạng (3.13) và ma trận kiểm tra dạng (3.18). Giả thử phía phát gửi đi từ mã **V** = (1000110). Phía thu nhận được từ mã **P**. Chúng ta xét hai trường hợp:

- Quá trình truyền không phát sinh lỗi, khi đó **P** ≡ **V**. Hãy chứng tỏ trong trường hợp này **S = 0** (với giả thiết số lỗi e ≤ 2).
- Quá trình truyền phát sinh lỗi, khi đó **P** ≠ **V**, chẳng hạn, **P** = (11*00110), ở đây dấu * chỉ ký hiệu mắc lỗi. Hãy chứng tỏ trong trường hợp này **S** ≠ **0**.

Để giải các bài toán trên, chúng ta ký hiệu **P** = (r_0, r_1, r_2, r_3, r_4, r_5, r_6), còn **S** = (s_0, s_1, s_2), theo công thức (3.19), chúng ta có:

$$S = P \times H^T_{(7,4)} = \left[r_0 r_1 r_2 r_3 r_4 r_5 r_6\right] \times \begin{bmatrix} 110 \\ 011 \\ 111 \\ 101 \\ 100 \\ 010 \\ 001 \end{bmatrix} = \left[s_0 s_1 s_2\right] \tag{3.19a}$$

(*Chú ý*: ma trận kiểm tra chuyển vị $H^T_{(n,k)}$ có thể biểu diễn theo các ma trận thành phần của ma trận sinh $G_{(n,k)}$ dưới dạng cột như sau:

$$H^T_{(n,k)} = \begin{bmatrix} R_{(r,k)} \\ E_{(r,r)} \end{bmatrix} \tag{3.17a}$$

cách biểu diễn này rất thuận tiện trong tính toán).

Chúng ta quay trở lại công thức (3.19a). Kết quả của phép nhân ma trận, chúng ta nhận được:

$$\begin{aligned} s_0 &= r_0 \oplus r_2 \oplus r_3 \oplus r_4 \\ s_1 &= r_0 \oplus r_1 \oplus r_2 \oplus r_5 \\ s_2 &= r_1 \oplus r_2 \oplus r_3 \oplus r_6 \end{aligned} \tag{3.20}$$

Các công thức (3.20) được gọi là các *tổng kiểm tra*, để xác định các phần tử của véctơ **S**.

Thay các giá trị cụ thể của hai trường hợp trên vào (3.20), chúng ta nhận được:

- Trường hợp 1:

$$\begin{aligned} s_0 &= 1 \oplus 0 \oplus 0 \oplus 1 = 0, \\ s_1 &= 1 \oplus 0 \oplus 0 \oplus 1 = 0, \\ s_2 &= 0 \oplus 0 \oplus 0 \oplus 0 = 0, \end{aligned}$$

Rõ ràng **S = 0**, dẫn đến kết luận **P** không mắc lỗi, như giả thiết.

- Trường hợp 2:

$$\begin{aligned} s_0 &= 1 \oplus 0 \oplus 0 \oplus 1 = 0, \\ s_1 &= 1 \oplus 1 \oplus 0 \oplus 1 = 1, \\ s_2 &= 1 \oplus 0 \oplus 0 \oplus 0 = 1, \end{aligned}$$

Trong trường hợp này **S** = (011) ≠ **0**, dẫn đến kết luận **P** mắc lỗi, như giả thiết.

Chúng ta hãy *phân tích để tìm lỗi* ở ký hiệu nào ? Xét kết quả của trường hợp 2 ở trên:

♣ Vì $s_0 = 0$, hơn nữa chúng ta giả thiết **P** chỉ mắc 1 lỗi, cho nên khẳng định các ký hiệu tham gia vào tổng này không mắc lỗi; tức là r_0, r_2, r_3, r_4 không mắc lỗi.

♣ Vì $s_1 = 1$, hơn nữa chúng ta giả thiết **P** chỉ mắc 1 lỗi, cho nên khẳng định các ký hiệu tham gia vào tổng này mắc 1 lỗi. Kết hợp với kết luận trên, dẫn đến hoặc r_1, hoặc r_5 bị mắc lỗi.

♣ Vì $s_2 = 1$, hơn nữa chúng ta giả thiết **P** chỉ mắc 1 lỗi, cho nên khẳng định các ký hiệu tham gia vào tổng này mắc 1 lỗi. Kết hợp với kết luận trên, dẫn đến hoặc r_1, hoặc r_6 bị mắc lỗi.

Kết hợp hai kết luận cuối, với giả thiết **P** chỉ mắc 1 lỗi, dẫn đến kết luận *phần tử mắc lỗi là r_1*, điều này hoàn toàn phù hợp với giả thiết từ đầu.

Khi thu có lỗi người ta thường sử dụng ký hiệu:

$$\mathbf{P} = \mathbf{V} + \mathbf{e} \tag{3.21}$$

Trong đó **e** gọi là véctơ lỗi. Trong thí dụ trên **e** = (0100000).

Bảng 1. Mối quan hệ giữa S và e

Véctơ lỗi **e**	Véctơ Syndrome S= $s_0s_1s_2$	Véctơ **S** trong hệ 10
1000000	110	3
0100000	011	6
0010000	111	7
0001000	101	5
0000100	100	1
0000010	010	2
0000001	001	4
0000000	000	0

Bằng cách phân tích tương tự cho các vị trí lỗi khác, chúng ta được kết quả trình bày trong bảng 1, về mối quan hệ giữa véctơ **S** và véctơ lỗi **e** (*Chú ý:* ở đây cũng như ở mọi nơi khác, chúng ta luôn viết theo *trình tự trực tiếp*, tức là theo chiều từ trái sang phải các chỉ số tăng dần, bắt đầu từ chỉ số 0, kết thúc với chỉ số 6).

Độc giả hãy tự tìm, để khẳng định, trong trường hợp véctơ **P** mắc hai lỗi, từ các tổng kiểm tra (3.20), có tổ hợp nào để **S = 0** không? Rõ ràng là không. Điều này khẳng định mã M(7, 4) với $d_0 = 3$, luôn luôn phát hiện được 2 lỗi, nhưng chỉ sửa được 1 lỗi. Trong trường hợp số lỗi nhiều hơn hoặc bằng 3, khi giải mã sẽ phát hiện nhầm. Chẳng hạn, tổ hợp r_0, r_2, r_6, mắc lỗi (3 lỗi), thì véctơ **S** vẫn bằng 0, tức là đã phát hiện nhầm.

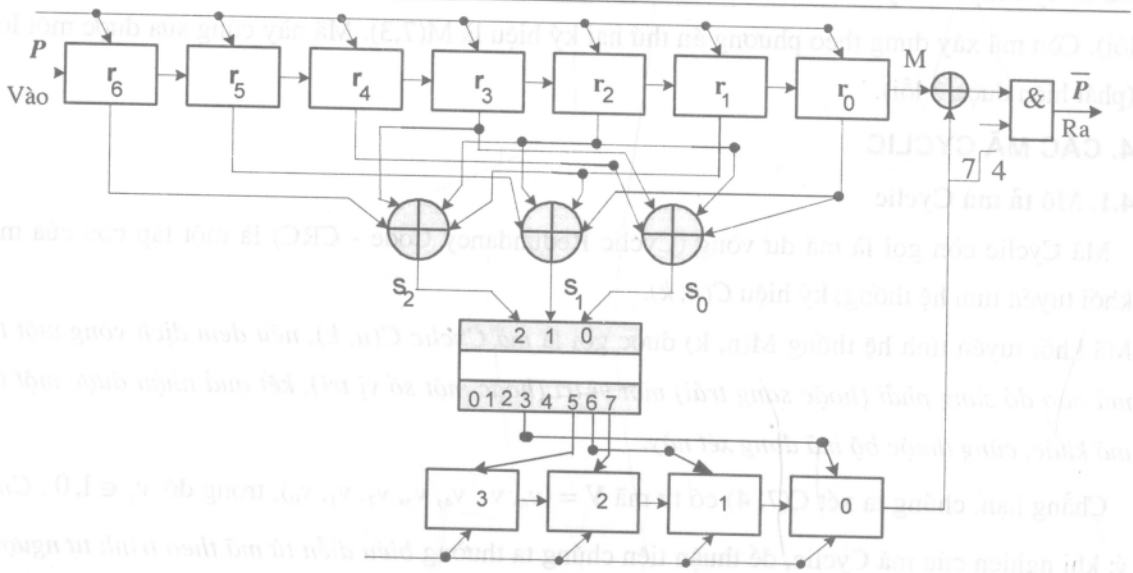
f). Mạch điện giải mã

Mạch điện giải mã được xây dựng dựa trên thuật toán giải mã như đã trình bày ở mục trên. Chúng ta xây dựng mạch điện giải mã cho trường hợp cụ thể mã M(7,4) với ma trận sinh đã xét ở trên.

Có *một vài phương án* xây dựng mạch điện giải mã. Chúng ta sẽ trình bày một phương án tương ứng với các phân tích ở mục trên, như hình 3.4.

Mạch điện giải mã bao gồm các phần tử:

- Mạch tính Syndrome, gồm 3 bộ cộng mod2, để tính (s_0, s_1, s_2). Các đầu vào của các XOR được vẽ căn cứ vào công thức (3.20);
- Mạch chuyển đổi (giải mã) véctơ **S** từ mã hệ 2 thành mã hệ 10, và đưa ra mức logic cao -"1" ở đầu ra tương ứng với con số của hệ 10; các đầu ra còn lại có mức logic 0. Mạch điện này được gọi là mạch tạo ký hiệu sửa sai;
- Mạch điện chứa dấu sửa sai (được đánh số từ 0 - 3). Phần tử r_i nào mắc lỗi, ô ghi dịch có ghi số tương ứng được nạp mức logic 1, các ô còn lại nạp mức logic 0;



Hình 3.4. Mạch điện giải mã của mã Hamming M(7, 4).

- Mạch sửa lỗi là XOR, ký hiệu M. Ký hiệu thu r_i nào mắc lỗi, sẽ được sửa nhờ phép cộng mod2: $r_i^* \oplus 1 = \overline{r}_i$ (ở đây ký hiệu $\overline{r}_i$ để chỉ đã được sửa), ở nhịp tương ứng. Các ký hiệu r_j còn lại (không mắc lỗi), không bị ảnh hưởng bởi phép cộng mod2 này: $r_j \oplus 0 \equiv r_j$.

Mạch điện giải mã cần làm việc trong 11 nhịp, trong đó 7 nhịp đầu để đưa các ký hiệu thu vào thanh ghi dịch đệm, tính Syndrome, chuyển mã hệ 2 sang mã hệ 10. 4 nhịp tiếp theo để sửa lỗi. Ở đầu ra là véctơ $\overline{P}$ đã sửa lỗi.

g). Giải mã theo thuật toán ngưỡng (hay đa số)

Các mã khối tuyến tính có thể giải mã theo thuật toán ngưỡng (hay đa số).

h). Các mã đổi lẫn.

Ma trận sinh $G_{(n,k)}$ và $H_{(n,k)}$ có tính chất đổi lẫn. Thực vậy, từ các quan hệ:

$$V \times H^T_{(n,k)} \equiv 0 \quad \text{và} \quad V = I \times G_{(n,k)}$$

Chúng ta dễ dàng dẫn ra các quan hệ:

$$G_{(n,k)} \times H^T_{(n,k)} \equiv 0 \quad \text{và} \quad H_{(n,k)} \times G^T_{(n,k)} \equiv 0 \qquad (3.22)$$

Các ma trận $G_{(n,k)}$ và $H_{(n,k)}$ chuyển vị là các ma trận trực giao với nhau; đồng thời các ma trận $H_{(n,k)}$ và $G_{(n,k)}$ chuyển vị cũng là các ma trận trực giao với nhau.

Từ các công thức (3.22), chúng ta thấy rằng các ma trận $G_{(n,k)}$ và $H_{(n,k)}$ có tính chất đổi lẫn. Trong trường hợp thứ nhất $G_{(n,k)}$ là ma trận sinh, còn $H_{(n,k)}$ là ma trận kiểm tra. Trong trường hợp thứ 2 $H_{(n,k)}$ là ma trận sinh, còn $G_{(n,k)}$ là ma trận kiểm tra. Các mã xây dựng theo hai phương án này được gọi là các *mã đổi lẫn*. Chẳng hạn, chúng ta khảo sát trường hợp (n,k) = (7,4). Mã xây dựng theo phương án thứ nhất ký hiệu là M(7,4). Mã này sửa được một lỗi (phát hiện được 2

lỗi). Còn mã xây dựng theo phương án thứ hai ký hiệu là M(7,3). Mã này cũng sửa được một lỗi (phát hiện được 3 lỗi).

4. CÁC MÃ CYCLIC

4.1. Mô tả mã Cyclic

Mã Cyclic còn gọi là mã dư vòng (Cyclic Redundancy Code - CRC) là một tập con của mã khối tuyến tính hệ thống, ký hiệu *C(n, k)*.

Mã khối tuyến tính hệ thống M(n, k) được gọi là *mã Cyclic C(n, k), nếu đem dịch vòng một từ mã nào đó sang phải (hoặc sang trái) một vị trí (hoặc một số vị trí), kết quả nhận được một từ mã khác, cũng thuộc bộ mã đang xét này.*

Chẳng hạn, chúng ta xét C(7, 4) có từ mã **V** = (v_6, v_5, v_4, v_3, v_2, v_1, v_0), trong đó $v_i \in 1,0$. *Chú ý*: khi nghiên cứu mã Cyclic, để thuận tiện chúng ta thường *biểu diễn từ mã theo trình tự ngược*.

Đem dịch vòng từ mã này *sang phải một vị trí*, thì kết quả ta có từ mã mới $\mathbf{V}_1 = V^{+1}$ = (v_5, v_4, v_3, v_2, v_1, v_0 ,v_6); hoặc đem dịch vòng từ mã này *sang trái một vị trí*, thì kết quả ta có từ mã mới $\mathbf{V}_2 = V^{-1}$ = (v_0,v_6, v_5, v_4, v_3, v_2, v_1), thì các từ mã $\mathbf{V}_1$, $\mathbf{V}_2$ cũng thuộc bộ mã đang xét.

Các mã Cyclic được ứng dụng rộng rãi trong thực tiễn, vì hai nguyên nhân:

- Thứ nhất, toán tử mã hoá và phép tính Syndrome được thực hiện rất đơn giản với việc sử dụng các thanh ghi dịch;
- Thứ hai, các mã này có cấu trúc đại số, vì vậy có thể tìm được các khả năng giải mã chúng đơn giản và hiệu quả.

Khi nghiên cứu mã Cyclic, các từ mã được biểu diễn dưới dạng *đa thức bậc (n-1)* theo các hệ số của chúng như sau:

$$V(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + \ldots + v_2x^2 + v_1x^1 + v_0 \qquad (3.23)$$

trong đó v_i là các phần tử của từ mã **V**, nhận các giá trị 1 hoặc 0; còn x^i là biến hình thức để chỉ vị trí của phần tử v_i tương ứng. Khi phần tử v_i nào đó bằng 0, thường không viết biến x^i tương ứng; chỉ các phần tử $v_i \neq 0$, sẽ viết biến x^i tương ứng.

Chẳng hạn, xét mã C(7, 4), từ mã $\mathbf{V}_1$ = (1000101), có đa thức mã tương ứng $V_1(x) = x^6 + x^2 + 1$. Một từ mã khác $\mathbf{V}_2$ = (1100010), có đa thức mã tương ứng $V_2(x) = x^6 + x^5 + x$. Một từ mã khác $\mathbf{V}_3$ = (0001011), có đa thức mã tương ứng $V_3(x) = x^3 + x + 1$. (3.24). Hơn nữa, từ mã $\mathbf{V}_3$ là kết quả dịch vòng phải một vị trí của từ mã $\mathbf{V}_1$, còn từ mã $\mathbf{V}_2$ là kết quả dịch vòng trái một vị trí của từ mã $\mathbf{V}_1$.

Chúng ta xác định các phép toán đối với đa thức trong trường nhị phân **GF(2)**:

a). Phép cộng (phép trừ): tổng (hoặc hiệu) của hai đa thức *f(x)* và *h(x)* trong trường nhị phân **GF(2)** là một đa thức cũng trong trường nhị phân ấy và được xác định như sau:

$$f(x) \oplus h(x) = \sum_{i=0}^{n-1} (f_i \oplus h_i).x^i \tag{3.25}$$

Nói một cách khác, tổng (hoặc hiệu) của hai đa thức *f(x)* và *h(x)* trong trường nhị phân **GF(2)** được thực hiện theo **mod2**, và phép cộng và phép trừ là như nhau.

Thí dụ:

$$\begin{array}{rrrrl} x^3 + & x^2 + & 0x + & 1 & \\ & & x + & 1 & \\ \hline x^3 + & x^2 + & x + & 0 & = x^3 + x^2 + x \end{array}$$

Hoặc

$$\begin{array}{rrrrl} x^3 + & x^2 + & 0x + & 1 & \\ & x^2 + & x + & 1 & \\ \hline x^3 + & 0x^2 + & x + & 0 & = x^3 + x \end{array}$$

b). Phép nhân: tích của 2 đa thức *f(x)* bậc (n-1) và đa thức *h(x)* bậc (m-1) trong trường nhị phân **GF(2)** là một đa thức cũng trong trường nhị phân ấy và được xác định như sau:

$$f(x).h(x) = \sum_{j=0}^{m-1} \left(\sum_{i=0}^{n-1} (f_i.h_j) \right) .x^{(i+j)} \tag{3.26}$$

nghĩa là tích nhận được theo qui tắc nhân thông thường, tuy nhiên, các hệ số nhận được cùng bậc của x được cộng theo mod2.

Thí dụ:

$$\begin{array}{rrrrl} & x^3 + & x^2 + & 0x + & 1 \\ & & & x + & 1 \\ \hline & x^3 + & x^2 + & 0x + & 1 \\ x^4 + & x^3 + & 0x^2 + & x & \\ \hline x^4 + & 0x^3 + & x^2 + & x + & 1 = x^4 + x^2 + x + 1 \end{array}$$

Hoặc

$$\begin{array}{rrrrl} & x^3 + & x^2 + & 0x + & 1 \\ & & x^2 + & x & \\ \hline & x^4 + & x^3 + & 0x^2 + & x \\ x^5 + & x^4 + & 0x^3 + & x^2 & \\ \hline x^5 + & 0x^4 + & x^3 + & x^2 + & x = x^5 + x^3 + x^2 + x \end{array}$$

c). Phép chia: phép chia được hình thành như sau: đối với mỗi cặp đa thức *f(x)* và *h(x)*, với *h(x)* ≠ 0, tồn tại duy nhất một cặp đa thức *q(x)*- phần nguyên và *r(x)*- phần dư, sao cho:

$$f(x) = q(x) \times h(x) + r(x) \tag{3.27}$$

ở đây bậc của đa thức dư $r(x)$ phải nhỏ hơn bậc của đa thức chia $h(x)$.

Nói cách khác, phép chia đa thức được diễn ra theo qui tắc chia số hạng có bậc cao nhất của *đa thức bị chia* cho số hạng cao nhất của *đa thức chia*. Để tìm đa thức dư của mỗi lần chia, phép trừ được diễn ra theo mod2. Các lần chia tiếp theo được diễn ra tương tự như trên, cho đến khi đa thức dư có bậc nhỏ hơn đa thức chia, thì phép chia kết thúc. Còn *đa thức dư của lần chia cuối cùng* được gọi là *đa thức dư* của phép chia, ký hiệu $r(x)$.

Thí dụ: $x^4 + 0x^3 + x^2 + x + 1 \;\Big|\; \dfrac{x+1}{x^3 + x^2 + 1}$ đa thức $x^3 + x^2 + 1 \leftarrow q(x)$.

$$\begin{array}{l}
\underline{x^4 + x^3} \\
\quad x^3 + x^2 + x + 1 \\
\quad \underline{x^3 + x^2} \\
\qquad\quad x + 1 \\
\qquad\quad \underline{x + 1} \\
\qquad\qquad 0 \leftarrow r(x).
\end{array}$$

Mã Cyclic C(n,k) ngoài đặc tính *dịch vòng*, còn có các đặc tính khác như sau:

1. *Trong mã Cyclic C(n,k), mỗi đa thức mã V(x) khác không cần phải có bậclớn hơn (n–k), nhưng không lớn hơn (n–1).*
2. *Tồn tại chỉ một đa thức mã g(x) có bậc (n-k) dạng:*

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \ldots + g_2x^2 + g_1x + 1$$

$$= g_rx^r + g_{r-1}x^{r-1} + \ldots + g_2x^2 + g_1x + g_0 \tag{3.28}$$

được gọi là đa thức sinh.

Các hệ số g_i nhận giá trị 1 hoặc 0; còn hệ số g_0, $g_r \equiv 1$.

3. *Mỗi đa thức mã V(x) đều là bội số của g(x), tức là:*

$$V(x) = q(x).g(x) \tag{3.29}$$

4. *Đa thức sinh g(x) là thừa số chia hết (không dư) của đa thức $(x^n + 1)$, tức là:*

$$(x^n + 1) = g(x).h(x) + 0 \tag{3.30}$$

Đa thức h(x) - phần nguyên của phép chia được gọi là đa thức kiểm tra.

Từ các đặc tính dịch vòng và các đặc tính trên, chúng ta đưa ra một hệ quả quan trọng: *khi nhân (hoặc chia) đa thức mã V(x) với đa thức h(x) nào đó ($h(x) \neq 0$), thì được định nghĩa là phép nhân (hoặc chia) trong trường hữu hạn. Vì vậy, đa thức tích (hoặc thương) $V'(x)$ muốn trở*

thành đa thức mã của bộ mã đang xét, phải lấy mod($x^n + 1$) để bậc của nó không lớn hơn (n-1), tức là:

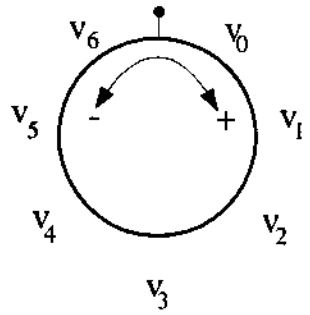
$$V'(x) = [V(x) \times h(x)] \bmod (x^n + 1) \tag{3.31}$$

Hoàn toàn tương tự như vậy cho *chia đa thức*, tức là:

$$V'(x) = \left[\frac{V(x)}{h(x)}\right] \bmod (x^n + 1) \tag{3.31a}$$

Đặc tính dịch vòng và đặc tính nhân (chia) các đa thức mã (cho mã C(7, 4)) được diễn giải như hình 3.5. Trong đó, các ký hiệu mã v_0, v_1,..., v_6 được gắn chặt vào vành tròn, trong khi đó, vành tròn có thể xoay sang phải (trái) tùy ý các vị trí. Có một vạch cố định để phân cách *đầu* với *đuôi* từ mã. Quy định, khi xoay vành tròn sang *phải*, tức theo chiều mũi tên với *dấu* +, i vị trí, tương ứng với phép *dịch vòng phải* từ mã V đi i ký hiệu mã. Và ngược lại, khi xoay vành tròn sang *trái*, tức theo chiều mũi tên với *dấu* –, i vị trí, tương ứng với phép *dịch vòng trái* từ mã V đi i ký hiệu mã. Mã C(7, 4) chỉ gồm $2^4 = 16$ từ mã (tức là số lượng từ mã là hữu hạn).

Chúng ta quay lại thí dụ ở công thức (3.24) và thấy: $V_3(x) = x.V_1(x) = V_1^{+1}(x)$ và $V_2(x) = \dfrac{V_1(x)}{x} = V_1^{-1}(x)$. Phép dịch vòng phải một vị trí tương ứng với phép nhân đa thức *V(x)* với x. Còn phép dịch vòng trái một vị trí tương ứng với phép chia đa thức *V(x)* cho x. Hơn nữa, V*(x)/x* $= \dfrac{x^6 + x^2 + 1}{x} = x^5 + x + x^{-1}.x^7 = x^6 + x^5 + x$. Trong công thức này $x^{-1} \bmod (x^7 + 1) = x^{-1}.x^7 = x^6$.



Hình 3.5. Diễn giải đặc tính dịch vòng của mã C(7, 4).

4.2. Đa thức sinh *g(x)*

Mọi đa thức mã *V(x)* của bộ mã C(n, k) đều được phát sinh từ đa thức các bit thông tin *I(x)* nhờ đa thức sinh *g(x)*.

Như đã chỉ ra từ *đặc tính 4* ở trên, mọi đa thức sinh *g(x)* của mã C(n, k) đều là *thừa số chia hết* của đa thức $(x^n + 1)$. Vì thế, để tìm một đa thức sinh cụ thể nào đó, cần phân tích đa thức trên ra thừa số. Các thừa số phải là các *đa thức tối giản (hay đa thức nguyên tố)*.

Chẳng hạn, xét đa thức sinh của bộ mã Cyclic với n = 7

$$(x^7 + 1) = (x+1)(x^3 + x + 1)(x^3 + x^2 + 1) \tag{3.32}$$

Từ 3 đa thức thừa số trên, chúng ta có thể chọn đa thức sinh *g(x)* và xác định các mã C(7, k) tương ứng như bảng 2.

Ở đây d_0 ký hiệu khoảng cách Hamming cực tiểu của bộ mã, $d_0 = W[g(x)]$.

Như vậy, với mã C(7, k) có 6 phương án chọn đa thức sinh *g(x)*. Có hai mã C(7, 4) và hai mã C(7, 3) với hai đa thức sinh khác nhau.

Bảng 2: Các phương án chọn *g(x)*

Đa thức sinh *g(x)*	Mã C(7,k) tương ứng	d_0
$g_1(x) = (x+1)$	C(7,6)	2
$g_2(x) = (x^3 + x + 1)$	C(7,4)	3
$g_3(x) = (x^3 + x^2 + 1)$	C(7,4)	3
$g_4(x) = (x+1)(x^3+x+1)$	C(7,3)	4
$= x^4 + x^3 + x^2 + 1$		
$g_5(x) = (x+1)(x^3+x^2+1)$	C(7,3)	4
$= x^4 + x^2 + x + 1$		
$g_6(x) = (x^3+x+1)\ x^3+x^2+1)$	C(7,1)	7

Tương như như trên, chúng ta có kết quả phân tích một vài đa thức $(x^n + 1)$, với các giá trị *n* khác nhau. Hơn nữa, các thừa số chỉ viết dưới dạng các hệ số, với chỉ số giảm dần:

$(x^9 + 1) = (11).(111).(1001001)$

$(x^{15} + 1) = (11).(111).(11111).(10011).(11001)$

$(x^{31} + 1) = (11)(100101)(101001)(101111)(110111)(111011)(111101)$

$(x^{23} + 1) = (11).(101011100011).(110001110101)$ → Mã Golay(23, 12), sửa được 3 lỗi, khi chọn thừa số thứ 2, hoặc thứ 3 làm đa thức sinh *g(x)*.

Trong các đa thức thừa số trên, với bậc $m \geq 3$ và $n = (2^m - 1)$ đều là *đa thức nguyên thuỷ*, trừ đa thức với hệ số (11111) không phải là đa thức nguyên thuỷ. Các *đa thức nguyên thuỷ* là các đa thức để tạo chuỗi tín hiệu giả ngẫu nhiên trong kỹ thuật trải phổ.

4.3. Thuật toán mã hoá

Giả thử rằng cần mã hoá véctơ các bit thông tin ***I***, dưới dạng:

$$I = (m_{k-1}, m_{k-2}, ..., m_2, m_1, m_0) \quad (3.33)$$

Đa thức phù hợp với nó có dạng:

$$I(x) = (m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + ... + m_2x^2 + m_1x + m_0) \quad (3.34)$$

Nhân $I(x)$ với x^{n-k} được:

$$I'(x) = I(x).\, x^{n-k} = (m_{k-1}x^{n-1} + m_{k-2}x^{n-2} + ... + m_2x^{n-k+2} + m_1x^{n-k+1} + m_0x^{n-k}) \quad (3.35)$$

Sử dụng lý thuyết về phép chia đa thức, có thể viết:

$$I'(x) = I(x).\, x^{n-k} = q(x).\, g(x) + r(x) \quad (3.36)$$

ở đây: $q(x)$ - đa thức phần nguyên, còn $r(x)$ - đa thức phần dư của phép chia đa thức $I'(x)$ cho đa thức sinh $g(x)$.

Bởi vì bậc của $g(x)$ bằng (n–k), thì bậc của $r(x)$ phải bằng (n–k–1) hoặc nhỏ hơn. Đa thức $r(x)$ có thể viết:

$$r(x) = r_{n-k-1}x^{n-k-1} + r_{n-k-2}x^{n-k-2} + ... + r_2x^2 + r_1x + r_0 \quad (3.37)$$

Từ (3.36), chúng ta có thể viết:

$$I'(x) + r(x) = q(x).g(x) \quad (3.38)$$

Rõ ràng, đa thức $I'(x) + r(x)$ là bội số của $g(x)$ và có bậc là (n-1) hoặc nhỏ hơn. Do đó, đa thức $I'(x) + r(x)$ là *đa thức mã*, kết quả của phép mã hoá đa thức các bit thông tin $I(x)$, tức là:

$$V(x) = I'(x) + r(x) = I(x).\, x^{n-k} + r(x)$$

Như vậy, từ mã gồm k bít thông tin và $r = (n-k)$ ký hiệu kiểm tra; được biểu diễn dưới dạng:

$$V(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + ... + v_2x^2 + v_1x + v_0 \quad (3.39)$$

Trong đa thức này: k hệ số đầu phù hợp với k bit thông tin, còn r hệ số cuối là các ký hiệu kiểm tra, được phát sinh từ thuật toán mã hoá.

Chúng ta có thể tóm tắt thuật toán mã hoá như sau:

Các đa thức mã $V(x)$ được xác định theo thuật toán mã hoá, gồm 3 bước sau:

Bước 1: Nhân I(x) với x^r, tức là: $I'(x) = x^r.I(x)$ (dịch vòng phải véctơ I đi r vị trí);

Bước 2: Xác định đa thức dư $r(x)$ của phép chia $I'(x)$ cho $g(x)$, tức là:

$$r(x) = I'(x) \bmod g(x);$$

Bước 3: Đa thức mã được xác định theo biểu thức:

$$V(x) = I'(x) + r(x).$$

Xét thí dụ: xét mã C(7, 4) với đa thức sinh $g(x) = (x^3 + x + 1)$. Biết véctơ các bít thông tin **I** = (1010), vậy đa thức tương ứng của nó: $I(x) = x^3 + x$. Các bước mã hoá như sau:

- $I'(x) = I(x).x^r = x^3(x^3 + x) = x^6 + x^4$

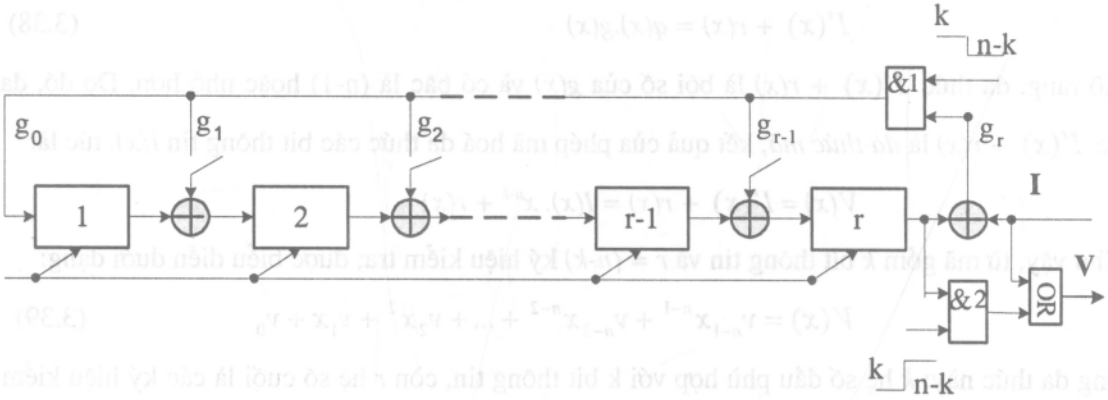
- Chia đa thức $I'(x)$ cho $g(x)$:

$$\begin{array}{l|l} x^6 + x^4 & x^3 + x + 1 \\ \hline \underline{x^6 + x^4 + x^3} & x^3 + 1 \\ 0 + x^3 & \\ \underline{x^3 + x + 1} & \\ x + 1 = r(x) & \end{array}$$

- Đa thức mã $V(x) = I'(x) + r(x) = x^6 + x^4 + x + 1$.

Từ mã tương ứng **V** = (1010011).

Thuật toán mã hoá dựa trên cơ sở phép *chia đa thức* có thể thực hiện bằng sơ đồ mạch chia. Đó là một thanh ghi dịch có *r* ô, trong đó mạch phản hồi kín phụ thuộc theo các hệ số của đa thức sinh *g(x)*, như hình 3.6.



Hình 3.6. Mạch điện mã hoá C(n, k).

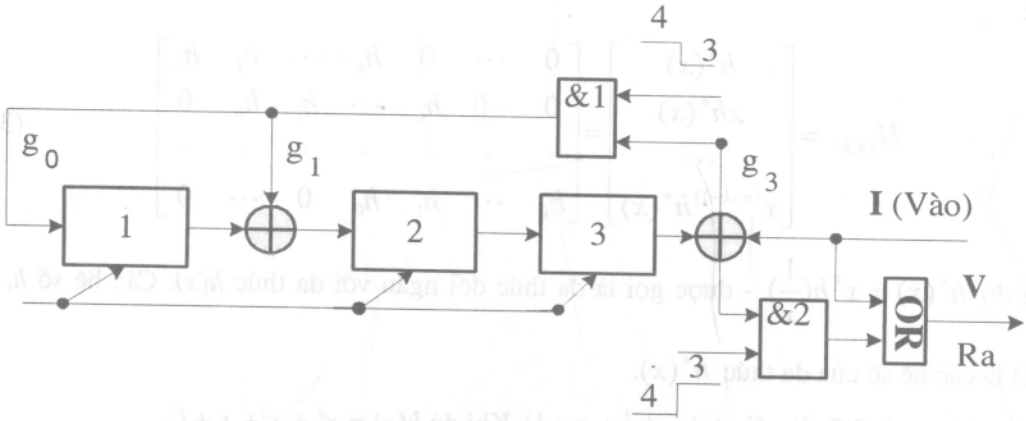
Trong mạch điện mã hoá C(n, k), hệ số g_i nào bằng 1, công tắc ứng với vị trí đó đóng và tồn tại bộ mod2; ngược lại, hệ số g_i nào bằng 0, công tắc ứng với vị trí đó ngắt và không tồn tại bộ mod2.

Mạch điện mã hoá làm việc trong n nhịp để hoàn thành quá trình mã hoá, và tạo ra từ mã **V**. Trong đó k nhịp đầu cổng AND1 mở, các bit thông tin đi vào mạch chia *I(x)/g(x);* sau k nhịp phép chia thực hiện xong, trong các ô ghi dịch chứa các ký hiệu dư. (n–k) nhịp tiếp theo cổng AND2 mở, các ký hiệu dư lần lượt đi ra ngoài, tạo nên r ký hiệu kiểm tra của **V**.

Mạch điện mã hoá của mã Hamming C(7, 4) ở thí dụ trên, với đa thức sinh: $g(x) = (x^3 + x + 1)$, như hình vẽ 3.7.

Khi xây dựng mạch mã hoá C(7, 4), cần chú ý: các hệ số g_0, g_1, $g_3 = 1$, còn hệ số $g_2 = 0$.

Mạch điện mã hoá của mã Hamming C(7, 4) làm việc trong n =7 nhịp, để hoàn thành một quá trình tạo ra từ mã **V**. Quá trình làm việc của C(7, 4), để mã hoá véctơ **I** = (1010) được trình bày trong bảng 3.



Hình 3.7. Mạch điện mã hoá của mã Hamming C(7, 4).

Bảng 3. Quá trình mã hoá véctơ I = 1010

Thứ tự nhịp	Trang thái các ô ghi dịch			Đầu vào	Đầu ra
	1	2	3	I	V
0	0	0	0	-	-
1	1	1	0	1	1
2	0	1	1	0	0
3	0	0	1	1	1
4	1	1	0	0	0
5	0	1	1	-	0
6	0	0	1	-	1
7	0	0	0	-	1

4.4. Đa thức kiểm tra

Như đã biết từ đặc tính 4 ở trên, đa thức sinh *g(x)* là thừa số chia không dư của đa thức $(x^n + 1)$, tức là:

$$(x^n + 1) = g(x).h(x) \tag{3.40}$$

trong đó *h(x)* gọi là đa thức kiểm tra. Hơn nữa $g(x).h(x) \equiv 0$.

Bởi vì *h(x)* được xác định duy nhất khi đã có *g(x)*, nó cũng xác định nên một mã, gọi là *mã đối lẫn* với mã do *g(x)* tạo nên.

Chẳng hạn, xét mã C(7, 4) với đa thức sinh $g(x) = (x^3 + x + 1)$. Khi đó đa thức kiểm tra được xác định $h(x) = (x+1)(x^3+x^2+1) = x^4 + x^2 + x + 1$. Nếu sử dụng $h(x)$ làm đa thức sinh, ta có mã C(7,3). Mã C(7,4) và mã C(7,3) này được gọi là *hai mã đối lẫn.*

Cùng với đa thức kiểm tra h(x), người ta còn xác định ma trận kiểm tra của mã Cyclic và có dạng:

$$H_{(n,k)} = \begin{bmatrix} h^*(x) \\ xh^*(x) \\ \cdots \\ x^{(n-k-1)}h^*(x) \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & h_k & \cdots & h_1 & h_0 \\ 0 & \cdots 0 & h_k & \cdots & h_1 & h_0 & 0 \\ . & . & . & . & . & . & . \\ h_k & \cdots & h_1 & h_0 & 0 & \cdots & 0 \end{bmatrix} \quad (3.41)$$

trong đó $h^*(x) = x^k h(\frac{1}{x})$ - được gọi là đa thức đối ngẫu với đa thức $h(x)$. Các hệ số h_i trong (3.41) là các hệ số của đa thức $h^*(x)$.

Chẳng hạn, mã C(7, 4) với $g(x) = (x^3 + x + 1)$. Khi đó $h(x) = x^4 + x^2 + x + 1$

$h^*(x) = x^k h(\frac{1}{x}) = x^4(\frac{1}{x^4} + \frac{1}{x^2} + \frac{1}{x} + 1) = 1 + x^2 + x^3 + x^4$. Viết theo trình tự giảm dần của các chỉ số được $h^*(x) = x^4 + x^3 + x^2 + 1$. Ma trận kiểm tra có dạng:

$$H_{(7,4)} = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (3.41a)$$

4.5. Tính Syndrome và sửa lỗi trong mã Cyclic.

Tính Syndrome đối với mã Cyclic tương đối đơn giản.

Giả thử V(x) và P(x) là đa thức mã khi phát và khi thu.

Chia P(x) cho g(x), chúng ta nhận được:

$$P(x) = q(x).g(x) + r(x) \quad (3.42)$$

ở đây: q(x) - đa thức phần nguyên của phép chia, còn r(x) - đa thức phần dư của phép chia. Đa thức Syndrome S(x) được định nghĩa là đa thức dư của phép chia, tức là:

$$S(x) \equiv r(x) \quad (3.42a)$$

Nếu $S(x) = 0$, từ mã thu được không mắc lỗi (với giả thiết số lỗi nằm trong khả năng kiểm soát của bộ mã); ngược lại $S(x) \neq 0$, từ mã thu được mắc lỗi.

Như vậy, $S(x) \neq 0$ xác định điều kiện có lỗi trong từ mã thu được.

Đa thức Syndrome S(x), trong trường hợp chung là đa thức có bậc (n-k-1), và được biểu diễn:

$$S(x) = s_{(n-k-1)}x^{(n-k-1)} + \ldots + s_1 x + s_0 \quad (3.43)$$

Rõ ràng, sơ đồ tính Syndrome S là sơ đồ chia tương tự như hình 3.6.

Khi từ mã thu được mắc lỗi, thì véctơ Syndrome **S** tối thiểu có một phần tử khác không.

Chúng ta hãy xác định mối quan hệ giữa S(x) và đa thức lỗi e(x), nhờ đó có thể định vị được lỗi và cho khả năng sửa chúng.

Giả thử e(x) có dạng:

$$e(x) = e_{(n-1)}.x^{(n-1)} + ... + e_1 x + e_0 \quad (3.44)$$

Như đã biết, khi thu có lỗi, P(x) biểu diễn dưới dạng:

$$P(x) = V(x) + e(x) = q(x).g(x) + S(x)$$

Hơn nữa, chúng ta lại có: $V(x) = m(x).\ g(x)$

Từ đó, chúng ta có:

$$e(x) = [q(x) + m(x)].g(x) + S(x) = f(x).g(x) + S(x) \quad (3.45)$$

Như vậy, *đa thức Syndrome S(x) là phần dư của phép chia e(x) cho g(x).* Từ đây chúng ta thấy rằng khi biết S(x) có thể xác định được e(x); hay xét về hoạt động của mạch điện giải mã khi biết **S** có thể giải mã được **e** và nhờ vậy có thể sửa được lỗi.

Thí dụ: Mã C(7,4) với đa thức sinh $g(x) = (x^3 + x + 1)$. Chúng ta có bảng 4, về mối giữa e(x) và S(x):

Trong bảng 4, giá trị của **S** trong hệ 2 (cột 3) có được từ phép chia **P/g** bằng mạch chia.

Bảng 4. Mối giữa e(x) và S(x)

Đa thức lỗi e(x)	Đa thức Syndrome S(x)	Véctơ **S** hệ 2	Véctơ **S** hệ 10
x^6	$x^2 + 1$	100	4
x^5	$x^2 + x + 1$	010	2
x^4	$x^2 + x$	001	1
x^3	$x + 1$	101	5
x^2	x^2	111	7
x	x	011	3
1	1	110	6

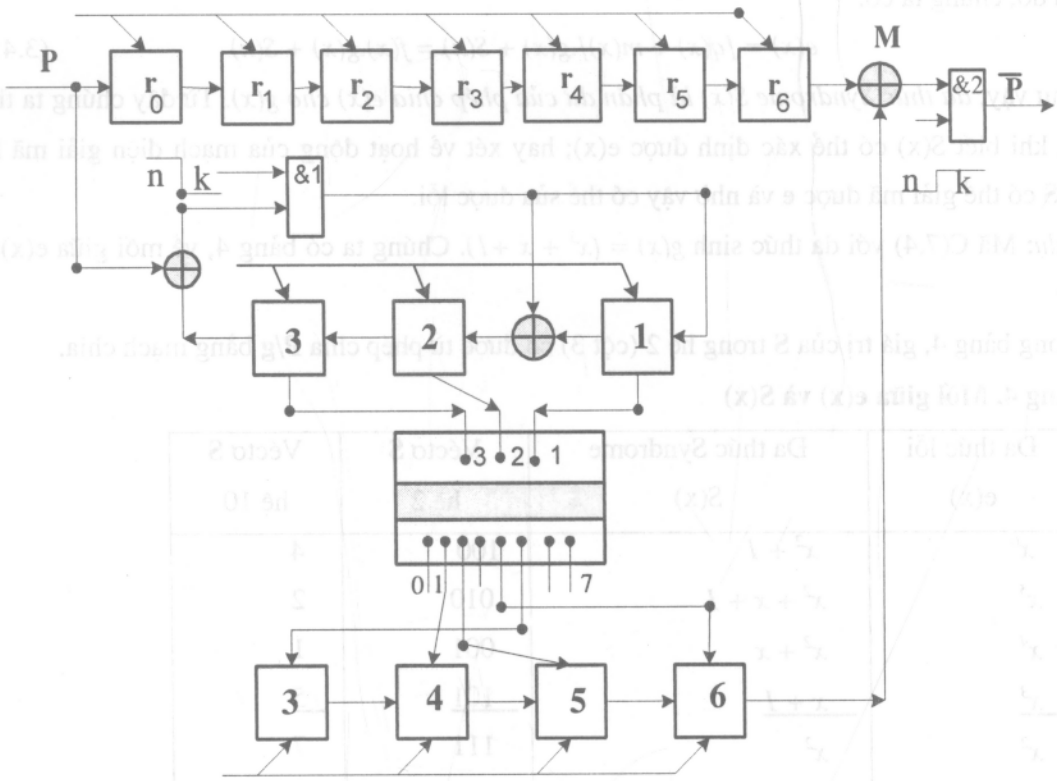
Với tất cả các phân tích ở trên, chúng ta có thể xây dựng mạch điện giải mã, tương tự như mạch điện giải mã của mã khối tuyến tính M(7, 4). Mạch điện giải mã C(7, 4) như hình 3.8.

Mạch điện giải mã bao gồm:

- Mạch thực hiện phép chia **P/g**, để xác định **S**;
- Mạch giải mã **S** từ hệ 2 sang hệ 10. Mạch này chỉ làm việc trong nhịp n (ở đây n = 7), thiết lập ở đầu ra tương ứng mức logic 1 (tương ứng với đầu vào có lỗi), còn các đầu ra còn lại mức logic 0.

- Hai thanh ghi dịch đệm. Thanh ghi dịch vẽ phía trên, gồm 7 ô, để chứa các ký hiệu của từ mã P. Thanh ghi dịch vẽ dưới, gồm 4 ô, để chứa dấu sửa sai. Khi bit thông tin nào bị mắc lỗi, ô ghi dịch với số hiệu tương ứng chứa dấu 1, các ô còn chứa dấu 0.
- Mạch sửa lỗi XOR, ký hiệu M.
- Cổng AND1 mở trong n nhịp đầu, để chia **P/g** và xác định **S**, đóng trong k nhịp tiếp theo. Cổng AND2 đóng trong n nhịp đầu, chỉ mở trong k nhịp tiếp theo, để đưa k bit thông tin đã sửa ra ngoài.

Mạch điện giải mã làm việc trong (n+k) nhịp, ở đây n+k = 7+4 = 11, để hoàn thành một quá trình giải mã. Mạch này chỉ sửa được một lỗi.



Hình 3.8. Mạch giải mã C(7, 4).

4.6. Mạch điện giải mã theo thuật toán chia dịch vòng

a). Thuật toán chia dịch vòng

Giả thử mã C(n, k), có khả năng sửa được t lỗi ($t \le (d_0 - 1)/2$), **V** là từ mã gửi đi, còn **P** là từ mã thu nhận. Thuật toán chia dịch vòng để sửa lỗi gồm 3 bước:

- *Bước 1*: Chia P(x) cho g(x) để tìm phần dư của phép chia, ký hiệu $r_0(x)$. Lấy trọng số của $r_0(x)$ và so sánh với t:

♣ Nếu $w[r_0(x)] \le t$, thì: $\overline{P}(x) = P(x) + r_0(x)$,

♣ Ngược lại, nếu $w[r_0(x)] > t$, phải chuyển sang *bước 2*.

- *Bước 2*: Dịch vòng phải (hoặc trái) P(x) một vị trí, tức là xác định:

$$P_1(x) = x.P(x) \qquad \text{(chỉ xét trường hợp dịch vòng phải)}$$

Đem $P_1(x)$ chia cho g(x), để xác định phần dư phép chia, ký hiệu $r_1(x)$. Lấy trọng số của $r_1(x)$ và so sánh với t:

♣ Nếu $w[r_1(x)] \le t$, thì: $\overline{P}(x) = \dfrac{P_1(x) + r_1(x)}{x}$, phép chia cho x, tương đương phép dịch vòng trái 1 vị trí.

♣ Nếu $w[r_1(x)] > t$, phải chuyển sang *bước 3*.

- *Bước 3*: Dịch vòng phải (hoặc trái) P(x) đi *i* vị trí, rồi tương tự như trên, tìm ra đa thức dư $r_i(x)$. Lấy trọng số của $r_i(x)$ và so sánh với t:

♣ Nếu $w[r_i(x)] \le t$, thì: $\overline{P}(x) = \dfrac{P_i(x) + r_i(x)}{x^i}$, phép chia cho x^i, tương đương phép dịch vòng trái *i* vị trí.

♣ Nếu sau (n-1) lần dịch vòng P(x), mà $w[r_{(n-1)}(x)] > t$. Kết luận: từ mã này có số lỗi *vượt khỏi khả năng sửa* của bộ mã.

Thí dụ: Xét mã C(7,4) với đa thức sinh $g(x) = (x^3 + x + 1)$. Mã này với $t = 1$. Từ mã phát **V** = (1010011). Giả thử từ mã thu **P** = (11*10011), ký hiệu có dấu (*) chỉ vị trí lỗi, tức là $e(x) = x^5$. Quá trình giải mã bằng thuật toán chia dịch vòng như sau:

- *Bước 1*: Đem P(x) chia cho g(x):

$$x^6 + x^5 + x^4 + x + 1 \;\Big|\; \frac{x^3 + x + 1}{x^3 + x^2}$$

$$\underline{x^6 + x^4 + x^3}$$

$$x^5 + x^3 + x + 1$$

$$\underline{x^5 + x^3 + x^2}$$

$$x^2 + x + 1 = r_0(x).$$

Rõ ràng $w[r_0(x)] = 3 > t = 1 \rightarrow$ *Bước 2*

- *Bước 2*: Xác định: $P_1(x) = x.P(x) = x(x^6 + x^5 + x^4 + x + 1) \bmod (x^7+1) =$

$$= (x^6 + x^5 + x^2 + x + 1)$$

Chia $P_1(x)$ cho g(x):

$$(x^6 + x^5 + x^2 + x + 1) \;\Big|\; \frac{x^3 + x + 1}{x^3 + x^2 + x}$$

$$\underline{x^6 + x^4 + x^3}$$

$$x^5 + x^4 + x^3 + x^2 + x + 1$$

$$\underline{x^5 + x^3 + x^2}$$
$$x^4 + x + 1$$
$$\underline{x^4 + x^2 + x}$$
$$x^2 + 1 = r_1(x)$$

Rõ ràng $w[r_1(x)] = 2 > t \rightarrow$ *Bước 3*

- *Bước 3*: Xác định: $P_2(x) = x.P_1(x) = x^6 + x^3 + x^2 + x + 1$

Chia $P_2(x)$ cho $g(x)$:

$$x^6 + x^3 + x^2 + x + 1 \;\Big|\; \frac{x^3 + x + 1}{x^3 + x}$$

$$\underline{x^6 + x^4 + x^3}$$
$$x^4 + x^2 + x + 1$$
$$\underline{x^4 + x^2 + x}$$
$$1 = r_2(x)$$

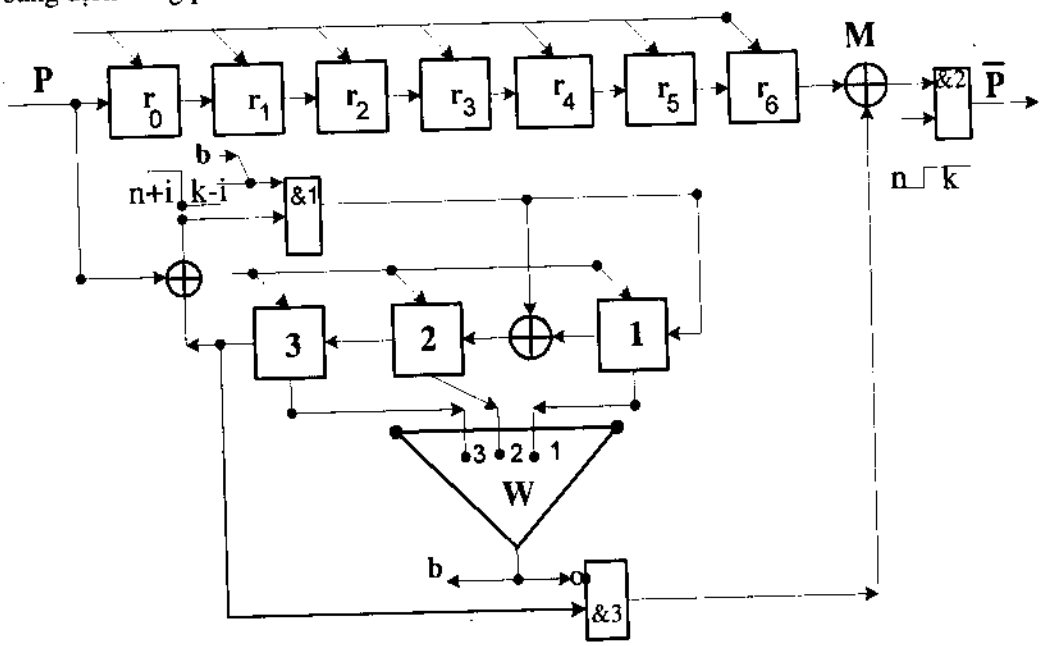
Rõ ràng $w[r_2(x)] = 1 = t$.

$$\overline{P}(x) = \frac{x^6 + x^3 + x^2 + x + 1 + 1}{x^2} = x^4 + x + 1 + x^{-1}.x^7 = x^6 + x^4 + x + 1.$$

Từ mã được sửa $\overline{P} = (1010011) \rightarrow$ hoàn toàn phù hợp giả thiết ban đầu.

b). Mạch điện giải mã

Khi xây dựng mạch điện giải mã theo thuật toán dịch vòng, người ta thay dịch vòng từ mã **P** bằng dịch vòng phần dư **r** (hoặc **S**).



Hình 3.9. Mạch điện giải mã theo thuật toán dịch vòng phần dư.

Chúng hoàn toàn tương đương nhau. Mạch điện giải mã theo thuật toán dịch vòng phần dư vẽ trên hình 3.9.

Cũng tương tự như hình 3.8, mạch điện giải mã theo thuật toán dịch vòng phần dư bao gồm:

- Mạch thực hiện phép chia **P** cho **g** để tìm phần dư **r**. Mạch này làm việc trong (n+i) nhịp, với *i* là số nhịp để dịch vòng **r** (trong trường hợp cụ thể này *i=1*),
- Mạch tính trọng số của phần dư, ký hiệu W. Mạch này làm việc trong k nhịp cuối. Nó có vai trò tính trọng số của phần dư **r**:

♣ Nếu W[r] > t, nó tạo tín hiệu logic 1 ở đầu vào b, tiếp tục mở cổng AND1, để dịch vòng **r** trong i nhịp, cho đến khi W[r] ≤ t, thì cổng AND1 đóng. Lúc đó cổng AND3 mở (vì có mức logic 1), đưa ký hiệu sửa từ mạch chia qua AND3 đi tới XOR M và sửa được bit thông tin bị lỗi, ở vị trí tương ứng với nhịp đó,

♣ Nếu W[r] ≤ t, cổng AND3 mở, đưa ký hiệu sửa từ mạch chia ở nhịp tương ứng đến XOR M, để sửa.

Mạch tính trọng số của phần dư, ký hiệu W, trong trường hợp cụ thể này là mạch ngưỡng. Nó làm việc theo đa số:- nếu có 2 đầu vào 0 (tương ứng với W[r] = t =1), nó đưa ra 0: AND1 đóng, nhờ đảo AND3 mở, đưa dấu sửa qua XOR để sửa - ngược lại, nếu có 2 đầu vào 1 (tương ứng với W[r] > t = 1), nó đưa ra 1, tiếp tục mở AND1 để dịch vòng S (trong trường hợp cụ thể này *i=1*, thực hiện dịch vòng trong 1 nhịp). Với cổng AND3, nhờ đảo thành 0, nên đóng.

Bảng 5. Quá trình giải mã nhờ thuật toán dịch vòng

Thứ tự nhịp	Trạng thái các ô ghi dịch			Đầu vào	Đầu ra
	1	2	3	**P**	$\overline{P}$
7	1	0	1	1	-
8	1	0	0	-	1
9	0	1	0	-	0
10	0	0	1	-	1
11	0	0	0	-	0

- Cổng AND2 mở trong k nhịp cuối, đưa k bit thông tin đã sửa ra ngoài.

Chúng ta lấy thí dụ về quá trình giải mã từ mã $P = (1011^{*}011)$, ở đây từ mã bị lỗi ở bit thông tin thứ 4, tức $e(x) = x^3$. Quá trình giải mã trình bày trong bảng 5. Hơn nữa, chỉ trình bày từ nhịp thứ 7, nhờ sử dụng kết quả của bảng 4.

Trong bảng 5, chúng ta thấy:

- Nhịp 7, kết thúc phép chia **P** cho **g**, trong các ô ghi dịch ghi các ký hiệu dư, ở đây ***r*** = 101. Như vậy, W[r] = 2 > t. Mạch W đưa ra mức 1,cổng AND1 tiếp tục mở, thực hiện dịch vòng **r,**
- Nhịp 8, thực hiện dịch vòng **r,**
- Nhịp 11 là nhịp sửa lỗi, ký hiệu r_3 bị mắc lỗi, như chúng ta giả thiết, được sửa thành $\overline{r}_3 = 0$. Kết quả $\overline{P} = (1010011)$.

Cũng như mạch giải mã C(7,4) khác, mạch này làm việc trong (n+k =7+4 =11) nhịp, để hoàn thành sửa 1 lỗi của từ mã.

4.7. Các phương pháp không đại số giải mã cyclic

Tất cả các phương pháp giải mã khối tuyến tính có thể chia ra thành hai nhóm: *đại số và không đại số.*

Cơ sở của các phương pháp đại số là nhằm giải hệ thống phương trình để xác định số lỗi và vị trí của chúng. Các phương pháp xác định véctơ (đa thức) syndrome thuộc nhóm này.

Trong khi đó các phương pháp không đại số đạt được điều này nhờ khái niệm cấu trúc đơn giản của lý thuyết mã hoá, chúng cho phép tìm ra tổ hợp lỗi một cách đơn giản hơn.

Một trong các phương pháp không đại số là phương pháp giải mã với việc sử dụng thuật toán Meggit, để sửa một lỗi cũng như m lỗi (thực tế m ≤ 3).

Khi giải mã theo thuật toán Meggit, tất nhiên cũng phải tính véctơ **S** (nhờ mạch chia **P/g**). Tuy nhiên, việc sử dụng nó có khác so với các phương pháp đã biết trước đây.

Cơ sở của thuật toán Meggit rất đơn giản, và dựa trên các đặc tính sau đây của mã cyclic:

- Tồn tại mối quan hệ duy nhất giữa tập các lỗi cần sửa và tập của véctơ **S** (xem bảng 4);
- Nếu đa thức S(x) phù hợp với đa thức lỗi e(x), thì **x.S(x) modg(x)** phù hợp với **x.e(x) mod(x^n + 1)**.

Đẳng thức a(x) = b(x) modp(x), được hiểu là: a(x) bằng b(x) theo modp(x), tức là a(x) và b(x) có cùng phần dư khi đem chia cho p(x).

Đặc tính thứ hai chỉ ra rằng: nếu dịch vòng e(x) sang phải một vị trí, thì để nhận được **S** mới, cần phải dịch các ký hiệu dư chứa trong ghi dịch nhờ đường phản hồi ngược, tất nhiên cũng một vị trí.

Sơ đồ giải mã cyclic theo thuật toán Meggit được trình bày trên hình 3.10. Trên hình 3.10a) chỉ ra sơ đồ giải mã chung. Sơ đồ bao gồm:

- Thanh ghi dịch đệm với n ô, để ghi nhớ n dấu mã của từ mã thu được;
- Mạch thực hiện phép chia P(x) cho g(x), để xác định các ký hiệu dư. Cả hai mạch này làm việc trong n nhịp thì kết thúc.

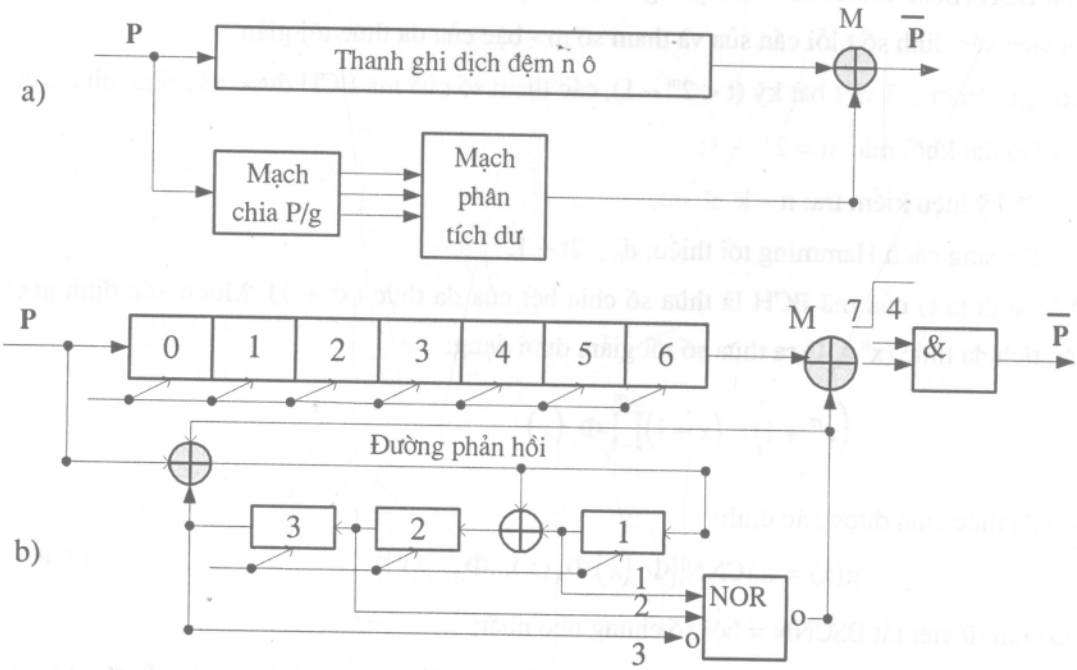
- Mạch phân tích dư, có chức năng xác định đúng vị trị của lỗi trong từ mã P, để phát sinh ký hiệu sửa lỗi thích hợp. Mạch này cần làm việc trong k nhịp tiếp theo;
- Mạch XOR để sửa lỗi.

Nếu số lỗi nằm trong khả năng sửa của bộ giải mã, thì sau (n+k) nhịp, mạch sẽ sửa xong một từ mã.

Trên hình 3.10b). trình bày mạch giải mã C(7, 4) với đa thức sinh $g(x) = x^3 + x + 1$ theo phương pháp Meggit.

- Trong 7 nhịp đầu, các ký hiệu mã P, được nạp vào thanh ghi theo trình tự từ ký hiệu a_6, cuối cùng đến a_0. Đồng thời mạch chia cũng làm việc, để khi kết thúc, xác định được các ký hiệu dư. Chúng được nhớ trong thanh syndrome.
- Mạch phân tích ký hiệu dư, bao gồm: phần tử NOR với 3 đầu vào, trong đó đầu vào 3 là đầu vào đảo. Một đường phản hồi từ đầu ra về đầu vào mạch chia. Mạch này có chức năng, sau khi sửa xong lỗi, sẽ đưa trạng thái các ô nhớ của thanh ghi syndrome về 0.

Chúng ta trình bày hoạt động của mạch để sửa lỗi từ mã P =1001*101, trong đó ký hiệu có dấu (*) chỉ ký hiệu mắc lỗi. Quá trình giải mã trình bày trên bảng 6, hơn nữa chỉ trình bày từ nhịp 7.



Hình 3.10. Sơ đồ khối chung của phương pháp giải mã Meggit a);

mạch giải mã của phương pháp này với đa thức sinh $g(x) = x^3 + x + 1$ b).

Sau khi kết thúc nhịp 7, trạng thái thanh ghi sydrome là 1- 0- 1. Vào nhịp 8, ba đầu vào của phần tử NOR là: 1- 0- 0, đầu ra NOR là 0 (do tác dụng đảo). Với mạch chia, ký hiệu dịch vòng

trở về ô số 1. Các nhịp 9, 10 tiếp theo là các nhịp để dịch ký hiệu đến ô thứ 3, đồng thời đầu ra NOR luôn là 0. Nhịp 11, ba đầu vào NOR là 0-0-0, cho nên đầu ra là 1. Đó là nhịp để sửa x^3. Đồng thời ở nhịp này, ký hiệu sửa sai theo đường phản hồi quay về đầu vào mạch chia, cộng mod2 với ký hiệu 1 ra ở đó, để đưa thanh ghi syndrome về toàn 0.

Sau (7 + 4 = 11) nhịp, mạch giải mã kết thúc quá trình giải mã sửa lỗi một từ mã (tất nhiên, chỉ một lỗi).

Bảng 7. Quá trình giải mã P

TT xung nhịp	Trạng thái các ô nhớ thanh ghi sydrome 1	2	3	$\overline{P}$ đầu ra
7	1	0	1	-
8	1	0	0	1
9	0	1	0	0
10	0	0	1	0
11	0	0	0	0

5. CÁC MÃ BCH

Các mã BCH (Bose Chaudhuri Hocquenghen) là một nhánh của mã Cyclic. Mã được xây dựng dựa trên việc xác định số t lỗi cần sửa và tham số m - bậc của đa thức tối giản.

Với số nguyên $m \geq 3$ và t bất kỳ ($t < 2^m - 1$), các tham số của mã BCH được xác định như sau:

- Độ dài khối mã: $n = 2^m - 1$;
- Số ký hiệu kiểm tra: $n - k \leq mt$;
- Khoảng cách Hamming tối thiểu: $d_0 \geq 2t + 1$.

Đa thức sinh g(x) của mã BCH là thừa số chia hết của đa thức $(x^n + 1)$. Muốn xác định g(x) cần phân tích đa thức $(x^n + 1)$ ra thừa số tối giản, dưới dạng:

$$\left(x^n + 1\right) = (x+1)\prod_{i=1}^{\leq m} \Phi_i(x)$$

Khi đó đa thức sinh được xác định:

$$g(x) = \text{BSCNN}\left\{\Phi_1(x).\Phi_3(x)...\Phi_{2t-1}(x)\right\} \quad (3.46)$$

trong đó cụm từ viết tắt BSCNN = bội số chung nhỏ nhất;

$\Phi_1(x)$ - đa thức bậc m, phải là *đa thức sinh cơ sở*. Một vài đa thức sinh cơ sở, với các bậc m khác nhau, được trình bày trong bảng 8.

Gọi α là nghiệm của $\Phi_1(x)$, thì nghiệm của $\Phi_3(x)$ phải là α^3, còn nghiệm của $\Phi_5(x)$ phải là α^5. Tức là nghiệm của các đa thức $\Phi_i(x)$ phải sắp xếp theo trình tự: $\alpha, \alpha^3, \alpha^5, \ldots, \alpha^{2i-1}$.

Thí dụ 1: xét trường hợp m = 3, n = 2^3-1 = 7. Phân tích đa thức (x^7+1) được hai đa thức là thừa số tối giản, như chúng ta đã biết trước đây. Đa thức sinh cơ sở $g_1(x)$ được chọn theo bảng 8.

Bảng 8. Các đa thức sinh cơ sở bậc m (m ≤ 32)

m	*g(x)*	m	*g(x)*
1	$x+1$	17	$x^{17}+x^2+1$
2	x^2+x+1	18	$x^{18}+x^7+1$
3	x^3+x+1	19	$x^{19}+x^6+x^2+x+1$
4	x^4+x+1	20	$x^{20}+x+1$
5	x^5+x^2+1	21	$x^{21}+x^2+1$
6	x^6+x+1	22	$x^{22}+x+1$
7	x^7+x+1	23	$x^{23}+x^5+1$
8	$x^8+x^4+x^3+x^2+1$	24	$x^{24}+x^7+x^2+x+1$
9	x^9+x^4+1	25	$x^{25}+x^3+1$
10	$x^{10}+x^3+1$	26	$x^{26}+x^8+x^2+x+1$
11	$x^{11}+x^2+1$	27	$x^{27}+x^5+x^2+x+1$
12	$x^{12}+x^8+x^4+x+1$	28	$x^{28}+x^3+1$
13	$x^{13}+x^4+x^3+x+1$	29	$x^{29}+x^2+1$
14	$x^{14}+x^{10}+x^6+x+1$	30	$x^{30}+x^{23}+x^2+x+1$
15	$x^{15}+x+1$	31	$x^{31}+x^3+1$
16	$x^{16}+x^{12}+x^3+x+1$	32	$x^{32}+x^{22}+x^2+x+1$

$$g_1(x) = \Phi_1(x) = x^3+x+1. \qquad (3.47)$$

Gọi nghiệm của $\Phi_1(x)$ là α, chúng ta có: $\alpha^3+\alpha+1=0$, hay $\alpha^3=\alpha+1$. Từ quan hệ này, chúng ta tiếp tục xác định được các α^i, với số mũ i khác nhau. Hơn nữa, chúng ta thấy các α^i được xác định theo quan hệ $\alpha^i \bmod (\alpha^7+1)$. Các biểu thức của hàm mũ α^i dưới dạng đa thức và nhị phân được trình bày trong bảng 9a.

Bảng 9a. Các biểu thức của hàm mũ α^i dưới dạng đa thức và nhị phân (m = 3)

Hàm mũ	Đa thức	Nhị phân	Hàm mũ	Đa thức	Nhị phân
0	0	000	α^3	$1+\alpha$	011
1	1	001	α^4	$\alpha^2+\alpha$	110
α	α	010	α^5	$\alpha^2+\alpha+1$	111
α^2	α^2	100	α^6	α^2+1	101

Dễ dàng xác định được $\Phi_3(x)$ như sau:

$$\Phi_3(x) = x^3 + x^2 + 1$$

Bởi vì $\Phi_3(\alpha^3) = 0$. Như vậy, đa thức sinh $g_2(x)$, được xác định:

$$g_2(x) = \left(x^3 + x + 1\right)\left(x^3 + x^2 + 1\right) \tag{3.48}$$

Kết quả này, không xa lạ đối với chúng ta, như đã biết khi nghiên cứu mã Cyclic.

Thí dụ 2: xét trường hợp m = 4, n = 2^4-1 = 15. Phân tích đa thức (x^{15} +1) thành các đa thức là thừa số tối giản được:

$$(x^{15}+1) = (x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)$$

Đa thức sinh cơ sở $g_1(x)$ được chọn là $\Phi_1(x)$, có dạng:

$$g_1(x) = \Phi_1(x) = x^4 + x + 1. \tag{3.49}$$

Tương ứng với mã C(15, 11), có $d_0 = 3$, sửa được t = 1 lỗi.

Nếu gọi nghiệm của $\Phi_1(x)$ là α. Chúng ta có $\alpha^4 + \alpha + 1 = 0$, hay $\alpha^4 = \alpha + 1$. Từ quan hệ này, chúng ta tiếp tục xác định được các α^i, với số mũ i khác nhau. Hơn nữa, chúng ta thấy các α^i được xác định theo quan hệ α^i mod (α^{15} +1). Các biểu thức của hàm mũ α^i dưới dạng đa thức và nhị phân được trình bày trong bảng 9b.

Để xác định đa thức sinh $g_2(x)$, cần xác định $\Phi_3(x)$. Hơn nữa, $\Phi_3(x)$ phải có nghiệm là α^3. Dễ dàng thử (hoặc phân tích ra thừa số của nghiệm) để xác được $\Phi_3(x)$:

$$\Phi_3(x) = x^4 + x^3 + x^2 + x + 1$$

Độc giả dễ dàng thay α^3 vào biểu thức trên, để được $\Phi_3(\alpha^3) = 0$. Như vậy, đa thức sinh $g_2(x)$ được xác định:

$$\begin{aligned} g_2(x) &= \Phi_1(x)\Phi_3(x) = (x^4 + x + 1)(x^4 + x^3 + x^2 + x + 1) \\ &= x^8 + x^7 + x^6 + x^4 + 1 \end{aligned} \tag{3.50}$$

Tương ứng với mã C(15, 7), có $d_0 = 5$, sửa được t = 2 lỗi.

Khi cần sửa t = 3 lỗi, cần xác định $\Phi_5(x)$.Dễ dàng xác định được $\Phi_5(x)$ có dạng:

$$\Phi_5(x) = x^2 + x + 1$$

Vì $\Phi_5(\alpha^5) = 0$. Như vậy, đa thức sinh $g_3(x)$ được xác định:

$$g_3(x) = g_2(x).\Phi_5(x) = x^{10} + x^8 + x^5 + x^4 + x^2 + x + 1 \qquad (3.51)$$

Tương ứng với mã C(15, 5), có $d_0 = 7$, sửa được t = 3 lỗi.

Bảng 9b. Các biểu thức của hàm mũ α^i dưới dạng đa thức và nhị phân (m = 4)`

Hàm mũ	Đa thức	Nhị phân	Hàm mũ	Đa thức	Nhị phân
0	0	0000	α^7	$\alpha^3 + \alpha + 1$	1011
1	1	0001	α^8	$\alpha^2 + 1$	0101
α	α	0010	α^9	$\alpha^3 + \alpha$	1010
α^2	α^2	0100	α^{10}	$\alpha^2 + \alpha + 1$	0111
α^3	α^3	1000	α^{11}	$\alpha^3 + \alpha^2 + \alpha$	1110
α^4	$\alpha + 1$	0011	α^{12}	$\alpha^3 + \alpha^2 + \alpha + 1$	1111
α^5	$\alpha^2 + \alpha$	0110	α^{13}	$\alpha^3 + \alpha^2 + 1$	1101
α^6	$\alpha^3 + \alpha^2$	1100	α^{14}	$\alpha^3 + 1$	1001

Thuật toán mã hóa và giải mã của mã BCH hoàn toan tương tự như mã C(n, k). Các giải thuật phát triển để giải mã BCH, độc giả có thể tham khảo ở tài liệu khác, chẳng hạn [13, 14]

6. MÃ REED-SOLOMON

Mã Reed-Solomon là một lớp con đặc biệt của mã BCH không nhị phân. Mã Reed-Solomon trong trường hợp chung được gọi là mã q–ary. Mã Reed-Solomon sửa sai t khối lỗi, sử dụng các ký hiệu trong trường **GF(q)**, có các tham số như sau:

- Độ dài khối mã: $n = q - 1$, với $q = p^m$, p là số nguyên tố, $m \geq 3$;
- Độ dài khối ký hiệu kiểm tra: $n - k = 2t$, với t là số khối mắc lỗi cần sửa;
- Khoảng cách tối thiểu của bộ mã: $d_0 = 2t + 1$.

Người ta thường ký hiệu mã Reed-Solomon: RS(n, k, 2t).

Hiện thời mã Reed-Solomon được sử dụng trong trường hợp đặc biệt $q = 2^m$, khi đó ta có mã Reed-Solomon trong trường **GF(2^m)**. Đa thức sinh g(x) trong trường hợp này được xác định như sau: gọi α là nghiệm của đa thức nguyên thủy bậc m, được chọn làm đa thức cơ sở cho tất cả các đa thức bậc $\leq m$. Gọi t là số khối mắc lỗi cần sửa, thì đa thức sinh được xác định:

$$g(x) = (x + \alpha)(x + \alpha^2)...(x + \alpha^{2t})$$

$$= g_0 + g_1 x + g_2 x^2 + ... + g_{2t-1} x^{2t-1} + x^{2t}$$

Rõ ràng g(x) có các số $\alpha, \alpha^2, ..., \alpha^{2t}$ là nghiệm và có các hệ số thuộc trường **GF(2^m)**. Sau đây, chúng ta xác định đa thức sinh g(x) cho vài mã RS đơn giản nhất.

Thí dụ 1: Xét mã RS (7, 5, 2) với tham số m = 3, n = 7, t = 1, k = 5. Đa thức sinh $g_1(x)$ có dạng:

$$g_1(x) = (x+\alpha)(x+\alpha^2) = x^2 + \alpha^4 x + \alpha^3 \tag{3.52}$$

Còn mã RS (7, 3, 4) với tham số m = 3, n = 7, t = 2, k = 3. Đa thức sinh $g_2(x)$ có dạng:

$$g_2(x) = g_1(x)(x+\alpha^3)(x+\alpha^4) = x^4 + \alpha^3 x^3 + 1.x^2 + \alpha.x + \alpha^3 \tag{3.53}$$

Thí dụ 2: Xét mã RS(15, k, 2t) với tham số m = 4, n = 15. Các đa thức sinh $g_i(x)$ được xác dưới dạng:

. Khi t = 1, $g_1(x) = (x+\alpha)(x+\alpha^2) = x^2 + \alpha^5 x + \alpha^3 \rightarrow$ RS(15, 13, 2)

. Khi t = 2, $g_2(x) = g_1(x)(x+\alpha^3)(x+\alpha^4) = x^4 + \alpha^{13}x^3 + \alpha^8 x^2 + \alpha^3 x + \alpha^{10}$

$\rightarrow$ RS(15, 11, 4)

. Khi t = 3, chúng ta có:

$$g_3(x) = g_2(x)(x+\alpha^5)(x+\alpha^6)$$
$$= x^6 + \alpha^{10}x^5 + 0.x^4 + \alpha^5 x^3 + \alpha^7 x^2 + \alpha^9 x + \alpha^6$$

$\rightarrow$ RS(15, 9, 6) (3.54)

6.1. Quá trình mã hóa: Giả thử chúng ta có đa thức biểu diễn các khối bit thông tin dưới dạng:

$$I(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + a_2 x^2 + a_1 x + a_0$$

trong đó các hệ số a_i là các phần tử của GF(2^m) và chúng là các số nhị phân m bit - gọi chung là khối m ký hiệu, có thể biểu diễn dưới dạng đa thức bậc m, với hệ số nhị phân có các giá trị ở cột 3 và cột 6 bảng 9 (B9a với m = 3; B9b với m = 4). Thuật toán mã hóa gồm 3 bước:

- Xác định: I'(x) = $x^r \times$I(x);
- Thực hiện tìm đa thức dư $r(x)$ = I'(x)modg(x);
- Đa thức mã V(x) = I'(x) + r(x)

Chúng ta thấy thuật toán này giống thuật toán mã hóa ở mã Cyclic. Tuy nhiên, điều khác biệt cơ bản ở đây là *phép nhân* và *phép chia* các hệ số của I(x) với g(x) là phép nhân và chia các đa thức bậc m, phải thực hiện theo *mod(α^n +1)*. Còn *phép cộng* và *phép trừ* các hệ số cùng bậc là *như nhau* và là phép *cộng mod2*. Để minh họa thuật toán mã hóa trên, chúng ta xét một vài thí dụ sau đây.

Thí dụ 1: Xét quá trình mã hóa với mã RS(7, 5, 2). Giả thử chúng ta có chuỗi số liệu, được trình diễn trong dạng các khối 3 bit, như sau:

I = 110_001_101_111_100.

Đa thức biểu diễn I(x) có dạng:

$$I(x) = \alpha^4 x^4 + 1.x^3 + \alpha.x^2 + \alpha^5 x + \alpha^2$$

Bước 1: Xác định I'(x) = x^r . I(x) = $x^2.I(x) = \alpha^4 x^6 + 1.x^5 + \alpha.x^4 + \alpha^5 x^3 + \alpha^2 x^2$

Bước 2: Xác định r(x) = I'(x)modg(x). Các tính toán trong mục này luôn sử dụng bảng 9a).

$$\alpha^4 x^6 + 1.x^5 + \alpha.x^4 + \alpha^5 x^3 + \alpha^2 x^2 \,\Big|\, \frac{x^2 + \alpha^4 x + \alpha^3}{\alpha^4 x^4 + \alpha^3 x^3 + \alpha x^2 + \alpha^6 x + 1}$$

$$\underline{\alpha^4 x^6 + \alpha x^5 + 1.x^4}$$

$$\alpha^3 x^5 + \alpha^3 x^4 + \alpha^5 x^3 + \alpha^2 x^2$$

$$\underline{\alpha^3 x^5 + 1.x^4 + \alpha^6 x^3}$$

$$\alpha x^4 + \alpha x^3 + \alpha^2 x^2$$

$$\underline{\alpha x^4 + \alpha^5 x^3 + \alpha^4 x^2}$$

$$\alpha^6 x^3 + \alpha.x^2$$

$$\underline{\alpha^6 x^3 + \alpha^3 x^2 + \alpha^6 x}$$

$$1.x^2 + \alpha^6 x$$

$$\underline{1.x^2 + \alpha^4 x + \alpha^3}$$

$$\alpha^3 x + \alpha^3 = r(x)$$

Bước 3: V(x) = I'(x) + r(x)

$$= \alpha^4 x^6 + 1.x^5 + \alpha.x^4 + \alpha^5 x^3 + \alpha^2 x^2 + \alpha^3 x + \alpha^3 \qquad (3.55)$$

Từ mã V tương ứng:

V = 110_001_101_111_100_011_011 (3.55a)

Qua thí dụ đơn giản này, chúng ta hãy suy nghĩ về phép chia I'(x)/g(x), thấy rằng trong đó ẩn chứa *quá nhiều phép tính số học* đối với các đa thức bậc m trong trường hữu hạn GF(2^m). Từ đó, sẽ hiểu vì sao thuật toán mã RS ra đời từ năm 1960, nhưng mãi gần đây nhờ xuất hiện các DSP (Digital Signal Processor) tốc độ cao, mã RS mới tìm thấy ứng dụng thực tiễn.

Thí dụ 2. Xét thí dụ khác về quá trình mã hóa với mã RS(7, 5, 2).

Giả thử chúng ta có đa thức biểu diễn khối các bít thông tin:

$$I(x) = \alpha^5 x^4 + \alpha^2 x^3 + \alpha^6 x^2 + \alpha^2 x + \alpha^4$$

Bằng các tính toán tương tự, chúng ta nhận được:

$$V(x) = \alpha^5 x^6 + \alpha^2 x^5 + \alpha^6 x^4 + \alpha^2 x^3 + \alpha^4 x^2 + \alpha^6 x + \alpha^5 \qquad (3.56)$$

Thí dụ 3. Xét quá trình mã hóa với mã RS(7, 3, 4). Giả thử chúng ta có chuỗi số liệu, được trình diễn trong dạng các khối 3 bit, như sau:

I = 111_011_110

đa thức mã tương ứng I(x) = $\alpha^5x^2 + \alpha^3x + \alpha^4$

Bước 1: Xác định I'(x) = x^r . I(x) = $x^4.I(x) = \alpha^5x^6 + \alpha^3x^5 + \alpha^4x^4$

Bước 2: Xác định r(x) = I'(x)modg(x)

$$\alpha^5x^6 + \alpha^3x^5 + \alpha^4x^4 \left| \frac{x^4 + \alpha^3x^3 + 1.x^2 + \alpha x + \alpha^3}{\alpha^5x^2 + 1.x + \alpha} \right.$$

$$r(x) = \alpha x^3 + \alpha x^2 + \alpha^5x + \alpha^4$$

Bước 3: V(x) = I'(x)+r(x) = $\alpha^5x^6 + \alpha^3x^5 + \alpha^4x^4 + \alpha x^3 + \alpha x^2 + \alpha^5x + \alpha^4$ (3.57)

Từ mã V tương ứng:

V = 111_011_110_010_010_111_110 (3.57a)

Thí dụ 4: Xét trường hợp RS(15, 9, 6), sửa được t = 3 khối lỗi. Giả thử chúng ta có chuỗi các khối bit thông tin dưới đây, hơn nữa, để tiện lợi cho sử dụng, chuỗi các khối bit thông tin được trình diễn dưới dạng các khối 4 bit như sau:

I = 0001_0111_0000_0110_1011_1010_1100_0001_0111;

tương ứng với đa thức:

$$I(x) = x^8 + \alpha^{10}x^7 + 0.x^6 + \alpha^5x^5 + \alpha^7x^4 + \alpha^9x^3 + \alpha^6x^2 + 1.x + \alpha^{10}$$

- *B1*: I'(x) được xác định theo I'(x) = x^r ×I(x), với r = 6, chúng ta có:

I'(x) = $x^{14} + \alpha^{10}x^{13} + 0.x^{12} + \alpha^5x^{11} + \alpha^7x^{10} + \alpha^9x^9 + \alpha^6x^8 + 1.x^7 + \alpha^{10}x^6$

- *B2*: Để tính đa thức dư r(x), chúng ta thực hiện phép chia: I'(x)/g(x), được:

$$x^{14} + \alpha^{10}x^{13} + 0.x^{12} + \alpha^5x^{11} + \alpha^7x^{10} + \alpha^9x^9 + \alpha^6x^8 + 1.x^7 + \alpha^{10}x^6 \,\underline{|g_3(x)}$$

Đa thức phần nguyên: q(x) = $x^8 + x$;

Đa thức phần dư: r(x) = $0.x^5 + \alpha^5x^4 + \alpha^7x^3 + \alpha^9x^2 + \alpha^6x + 0$

- *B3*: Đa thức V(x) tương ứng: V(x) = I'(x) + r(x)

$$V(x) = x^{14} + \alpha^{10}x^{13} + 0.x^{12} + \alpha^5x^{11} + \alpha^7x^{10} + \alpha^9x^9 + \alpha^6x^8 + 1.x^7 + \alpha^{10}x^6$$
$$+ 0.x^5 + \alpha^5x^4 + \alpha^7x^3 + \alpha^9x^2 + \alpha^6x + 0 \quad (3.58)$$

Từ mã V tương ứng:

V= 0001_0111_0000_0110_1011_1010_1100_0001_0111_0000
_0110_1011_1010_1100_0000.

6.2. Quá trình giải hóa

a). Giải mã RS sửa 1 lỗi

Gọi V(x) là từ mã phát, có dạng chung như sau:

$$V(x) = v_{n-1}x^{n-1} + v_{n-2}x^{n-2} + ... + v_2x^2 + v_1x + v_0$$

trong đó các hệ số v_i là khối m ký hiệu nhị phân;

Gọi P(x) là từ mã thu, có dạng chung như sau:

$$P(x) = r_{n-1}x^{n-1} + r_{n-2}x^{n-2} + ... + r_2x^2 + r_1x + r_0 = V(x) + e(x)$$

trong đó e(x) - đa thức lỗi, có dạng chung:

$$e(x) = e_{n-1}x^{n-1} + e_{n-2}x^{n-2} + ... + e_2x^2 + e_1x + e_0$$

các hệ số r_i , e_i là khối m ký hiệu nhị phân. Giữa chúng có mối quan hệ:

$$r_i = v_i + e_i$$

Quá trình giải mã RS giống quá trình giải mã BCH, cần phải xác định véctơ Syndrome, dưới dạng:

$$S = s_0, s_1, \ldots, s_{2t} .$$

Trong mã RS sửa 1 lỗi, các giá trị Syndrome được xác định:

$$s_0 = P(\alpha) = e(\alpha)$$

$$s_1 = P(\alpha^2) = e(\alpha^2) \qquad (3.59)$$

Khi không mắc lỗi, tất nhiên $s_0 = s_1 = 0$. Khi khối m ký hiệu thứ i bị lỗi, giá trị của các Syndrome sẽ là:

$$s_0 = e_i . \alpha^i,$$

$$s_1 = e_i . \alpha^{2i}.$$

Từ công thức này, chúng ta xác định được vị trí khối m ký hiệu thứ i bị lỗi và giá trị khối lỗi e_i tương ứng, theo công thức:

$$\begin{cases} \alpha^i = \dfrac{s_1}{s_0} \rightarrow x^i \\ e_i = \dfrac{s_0^2}{s_1} \rightarrow \hat{\alpha}_i \end{cases} \qquad (3.60)$$

trong công thức trên, ký hiệu x^i - chỉ vị trí của biến x mũ i bị mắc lỗi;

còn $\hat{\alpha}$ - giá trị khối lỗi. Sử dụng nó để tìm ra $\overline{\alpha}_i$ (khối đúng) theo công thức:

$$\overline{\alpha}_i = \hat{\alpha}_i + \alpha_i^* \quad (\alpha_i^* \text{ - hệ số } \alpha \text{ bị mắc lỗi ở vị trí tương ứng với } x^i)$$

Thí dụ 1: Chúng ta thử nghiệm các tính toán của thuật toán trên, với thí dụ 2, ở phần mã hóa. Các tính toán trong mục này luôn sử dụng bảng 9a). Khi đó, chúng ta có đa thức mã phát V(x) (xem công thức (3.56)) như sau:

$$V(x) = \alpha^5 x^6 + \alpha^2 x^5 + \alpha^6 x^4 + \alpha^2 x^3 + \alpha^4 x^2 + \alpha^6 x + \alpha^5$$

Giả thử từ mã thu có lỗi ở vị trí (i = 6), lỗi được đánh dấu (*)

$$P(x) = \alpha^{4*} x^{6*} + \alpha^2 x^5 + \alpha^6 x^4 + \alpha^2 x^3 + \alpha^4 x^2 + \alpha^6 x + \alpha^5$$

Chúng ta xác định được các thành phần Syndrome theo (3.59):

$$s_0 = P(\alpha) = \alpha^4\alpha^6 + \alpha^2\alpha^5 + \alpha^6\alpha^4 + \alpha^2\alpha^3 + \alpha^4\alpha^2 + \alpha^6\alpha + \alpha^5 = \alpha^6;$$

$$s_1 = P(\alpha^2) = \alpha^4\alpha^{12} + \alpha^2\alpha^{10} + \alpha^6\alpha^8 + \alpha^2\alpha^6 + \alpha^4\alpha^4 + \alpha^6\alpha^2 + \alpha^5 = \alpha^5;$$

Xác định theo công thức (3.60):

- vị trí lỗi: $\alpha^i = \dfrac{\alpha^5}{\alpha^6} = \alpha^{-1}.\alpha^7 = \alpha^6 \rightarrow$ lỗi ở vị trí x^6;

- giá trị của lỗi: $e_i = \dfrac{(\alpha^6)^2}{\alpha^5} = \alpha^7 = 001 = \hat{\alpha}_i$

Hệ số α_i đúng, được xác định:

$$\overline{\alpha}_i = \hat{\alpha}_i + \alpha_i^* = 001 \oplus 110 = 111 = \alpha^5.$$

Đúng như giả thiết từ đầu.

b). Giải mã RS sửa 2 lỗi

Chúng ta cũng giả thiết từ mã phát V(x), có bậc (n-1), trong đó có (n - 4) khối bit thông tin, mỗi khối m bit và 4 khối ký hiệu kiểm tra, mỗi khối m ký hiệu. Từ mã thu được P(x), giả thiết có 2 lỗi. Giả sử khối ký hiệu thứ i và thứ k bị mắc lỗi. Để sửa các lỗi này cần tính các Syndrome sau:

$$s_0 = P(\alpha) = e(\alpha) = e_i\alpha^i + e_k\alpha^k$$

$$s_1 = P(\alpha^2) = e(\alpha^2) = e_i\alpha^{2i} + e_k\alpha^{2k}$$

$$s_2 = P(\alpha^3) = e(\alpha^3) = e_i\alpha^{3i} + e_k\alpha^{3k}$$

$$s_3 = P(\alpha^4) = e(\alpha^4) = e_i\alpha^{4i} + e_k\alpha^{4k}$$

Ký hiệu Y_1 và Z_1 là giá trị lỗi và vị trí lỗi thứ nhất, còn Y_2 và Z_2 là giá trị lỗi và vị trí lỗi thứ hai ($Y_i \equiv e_i$, $Z_i \equiv \alpha^i$). Khi đó chúng ta có:

$$\begin{bmatrix} Z_1 & Z_2 \\ Z_1^2 & Z_2^2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} s_0 \\ s_1 \end{bmatrix} \quad (3.61a) \qquad \begin{bmatrix} Z_1^3 & Z_2^3 \\ Z_2^4 & Z_2^4 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} s_2 \\ s_3 \end{bmatrix} \quad (3.61b)$$

Chúng ta định nghĩa đa thức xác định vị trí của lỗi như sau [13, 14]:

$$\sigma(x) = (x + Z_1)(x + Z_2) = x^2 + \sigma_1(x) + \sigma_2$$

Gọi Z_u (u =1, 2) - là nghiệm của phương trình (3.62), phải thỏa mãn điều kiện sau:

$$Z_u^2 + \sigma_1 Z_u + \sigma_2 = 0 \tag{3.62}$$

Để xác định σ_1, σ_2, chúng ta nhân cả hai vế của phương trình (3.62) với $Y_u.Z_u^{i+1}$, rồi cộng từng vế, khi biến đổi chú ý đến các công thức (3.61), kết quả được:

$$\sum_{u=1}^{2}\left(Y_u.Z_u^{3+i} + \sigma_1 Y_u Z_u^{2+i} + \sigma_2 Y_u Z_u^{1+i}\right) = s_{i+2} + \sigma_1 s_{i+1} + \sigma_2 s_1 = 0 \tag{3.63}$$

trong đó i = 1, 2, 3, 4. Sử dụng công thức (3.63) với i = 0, 1, chúng ta có:

$$\begin{bmatrix} s_1 & s_0 \\ s_2 & s_1 \end{bmatrix} \times \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} = \begin{bmatrix} s_2 \\ s_3 \end{bmatrix}$$

Từ phương trình trên, chúng ta xác định được σ_1, σ_2 như sau:

$$\sigma_1 = \frac{s_1 s_2 + s_0 s_3}{s_1^2 + s_0 s_2}$$

$$\sigma_2 = \frac{s_1 s_3 + s_2^2}{s_1^2 + s_0 s_2} \tag{3.64}$$

Để xác định vị trí lỗi Z_1, Z_2 cần thực hiện phép thử, bằng cách thay thế lần lượt α^0 đến α^{n-1} vào công thức (3.62), để tìm ra các α^i thích hợp.

Thay thế các Z_i thích hợp vào (3.61a), chúng ta xác định được:

$$Y_1 = \frac{Z_2^2.s_0 + Z_2.s_1}{Z_1.Z_2^2 + Z_1^2.Z_2}$$

$$Y_2 = \frac{Z_1^2.s_0 + Z_1.s_1}{Z_1.Z_2^2 + Z_1^2.Z_2} \tag{3.65}$$

Chú ý: chỉ xảy ra hai lỗi khi $\Delta_2 = s_1^2 + s_0 s_2 \neq 0$, nếu ngược lại $\Delta_2 = s_1^2 + s_0 s_2 = 0$, thì từ mã đang xét chỉ mắc một lỗi.

Thí dụ 1: Xét mã RS(7, 3, 4). Giả thử từ mã phát V(x) có dạng:

$$V(x) = \alpha^3 x^6 + \alpha^4 x^5 + \alpha^6 x^4 + \alpha^3 x + \alpha^6$$

Giả thử từ mã thu được có 2 lỗi, tương ứng với vị trí i = 5, k = 6, có dấu (*) như sau:

$$P(x) = \alpha^{4*} x^{6*} + \alpha^{6*} x^{5*} + \alpha^6 x^4 + \alpha^3 x + \alpha^6$$

Chúng ta tính các giá trị syndrome như sau:

$$s_0 = e(\alpha) = \alpha^{10} + \alpha^{11} + \alpha^{10} + \alpha^4 + \alpha^6 = \alpha^6$$

$$s_1 = e(\alpha^2) = \alpha^{16} + \alpha^{16} + \alpha^{14} + \alpha^5 + \alpha^6 = \alpha^3$$

$$s_2 = e(\alpha^3) = \alpha^{22} + \alpha^{21} + \alpha^{18} + \alpha^6 + \alpha^6 = \alpha^6$$

$$s_3 = e(\alpha^4) = \alpha^{28} + \alpha^{26} + \alpha^{22} + \alpha^7 + \alpha^6 = 0$$

Xác định $\Delta_2 = s_1^2 + s_0 s_2$, được: $\Delta_2 = \alpha$. Từ đó, theo (3.64), chúng ta xác định được:

$$\sigma_1 = \alpha^2 / \alpha = \alpha$$

$$\sigma_2 = \alpha^5 / \alpha = \alpha^4$$

Sau khi xác định được σ_1, σ_2, chúng thử với các α^i với i = 0,1,..,6 vào phương trình $Z_u^2 + \sigma_1 Z_u + \sigma_2 = 0$. Chẳng hạn, với $\alpha^i = \alpha^5$ (i = 5), chúng ta có:

$$\alpha^{10} + \alpha.\alpha^5 + \alpha^4 \equiv 0$$

với $\alpha^i = \alpha^6$ (k =6), chúng ta có:

$$\alpha^{12} + \alpha.\alpha^6 + \alpha^4 \equiv 0$$

Như vậy, vị trí lỗi ứng với i = 5, k = 6, ($Z_1 = \alpha^5$, $Z_2 = \alpha^6$) như đã giả thiết. Độc giả hãy tự thử với các α^i còn lại, i = 0, 1, ..., 4, xem có giá trị nào thỏa mãn không.

Chúng ta xác định các giá trị lỗi theo công thức (3.65), như sau:

$$Y_1 = \frac{\alpha^{12}.\alpha^6 + \alpha^6.\alpha^3}{\alpha^5.\alpha^{12} + \alpha^{10}.\alpha^6} = \frac{\alpha^4 + \alpha^2}{\alpha^3 + \alpha^2} = \frac{\alpha}{\alpha^5} = \alpha^{-4}.\alpha^7 = \alpha^3$$

$$Y_2 = \frac{\alpha^{10}.\alpha^6 + \alpha^5.\alpha^3}{\alpha^5.\alpha^{12} + \alpha^{10}.\alpha^6} = \frac{\alpha^2 + \alpha}{\alpha^3 + \alpha^2} = \frac{\alpha^4}{\alpha^5} = \alpha^{-1}.\alpha^7 = \alpha^6$$

Xác định giá đúng của khối m ký hiệu như sau:

$$\overline{\alpha}_5 = Y_1 + \alpha^{6*} = \alpha^3 + \alpha^6 = \alpha^4$$

$$\overline{\alpha}_6 = Y_2 + \alpha^{4*} = \alpha^6 + \alpha^4 = \alpha^3$$

Kết quả này hoàn toàn phù hợp với giả thiết ở trên.

Quá trình giải mã RS dạng đơn giản như trên cũng rất phức tạp, phải thực hiện quá nhiều phép tính số học với đa thức bậc m trong trường hữu hạn GF(2^m).

Mã RS là loại mã khối hiệu quả nhất hiện nay (theo quan điểm để sửa t khối lỗi, chỉ cần 2t khối ký hiệu dư), đặc biệt nó rất thích hợp để sửa lỗi đám dài.

7. CÁC MÃ XOẮN

Các phương pháp mã hoá và giải mã, mà chúng ta vừa khảo sát trong các tiết trên, thuộc loại mã khối. Khi sử dụng các mã như thế, chuỗi ký hiệu thông tin nối tiếp được chia thành các khối và được mã hoá riêng biệt. Như vậy, kết quả mã hoá chúng ta được các từ mã không phụ thuộc lẫn nhau, và có cùng độ dài.

Khi sử dụng mã xoắn dòng số liệu được chia thành các khối có độ dài k nhỏ hơn nhiều (trong trường hợp phổ biến k = 1), được gọi là *các đoạn bit thông tin*.

Các đoạn bit thông tin được mã hoá, để nhận được từ mã dài *n* ký hiệu. Mã xoắn là loại *mã có nhớ*, trong mạch điện mã hoá bao gồm: *k bit thông tin vào, m ô nhớ, n bộ cộng mod2 và một bộ dồn kênh ở đầu ra*. Nhờ bộ dồn kênh mà cứ k bit thông tin vào, sẽ tạo ra n ký hiệu mã đầu ra. Hơn nữa, *n* ký hiệu mã đầu ra không chỉ phụ thuộc vào *k bit thông tin vào ở cùng thời điểm*, mà còn phụ thuộc vào *m bit thông tin* vào ở các *thời điểm trước đó*, được ghi nhớ trong *m* ô ghi dịch. Vì thế, mã xoắn thường ký hiệu là X(n, k, m).

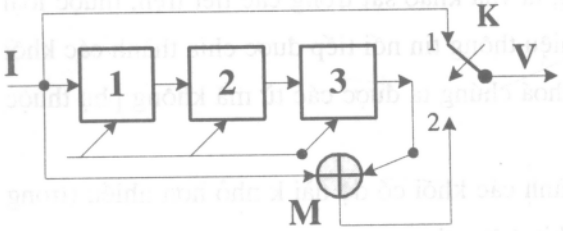
Sự phát triển của mã xoắn khác với sự phát triển của mã khối. Khi xây dựng mã khối, các *phương pháp đại số* thường được sử dụng rộng rãi để giải mã chúng. Trong trường hợp mã xoắn thì không. Phần lớn các mã xoắn thường tìm con đường nghiên cứu nhờ *máy tính điện tử* để chọn ra các đặc tính ưu việt nhất. Giải mã xoắn được diễn ra bằng phương pháp *hàm hợp lý cực đại*, hơn nữa, trong trường hợp này chúng được thực hiện khá đơn giản.

7.1. Mã hoá mã xoắn

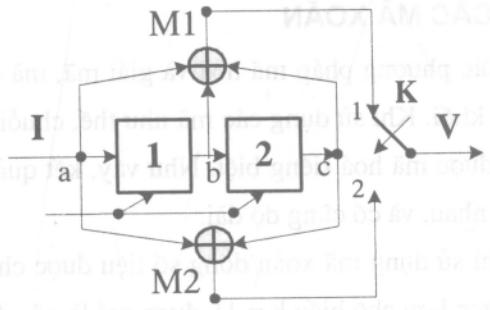
Trên hình 3.11, trình bày ba mạch điện mã hoá mã xoắn đơn giản nhất. Hình 3.11a) biểu diễn mã xoắn hệ thống X(2, 1, 3). Các mạch điện như thế này được gọi là *mã xoắn hệ thống*, vì trong các ký hiệu mã đầu ra, chúng ta có thể phân biệt được đâu là các bit thông tin, đâu là các ký hiệu kiểm tra. Còn trong hình 3.11b) biểu diễn mã xoắn không hệ thống X(2, 1, 2). Các mạch điện như thế này được gọi là *mã xoắn không hệ thống*, vì trong các ký hiệu mã đầu ra, chúng ta *không thể* phân biệt được đâu là các bit thông tin, đâu là các ký hiệu kiểm tra. Trên hình 3.11c) biểu diễn mạch điện mã hoá mã xoắn Vâyner Esse. Mạch điện này điển hình cho mã xoắn *k bit thông tin đầu vào* (k = 3) và *n ký hiệu mã đầu ra (n = 4)*.

Tốc độ mã hoá (cũng gọi tỷ lệ mã) của các mạch mã hoá được ký hiệu là R và được xác định $R = k/n$. Tham số này cũng đặc trưng cho độ dư thừa của bộ mã $g_c = k/n$. Ở hai mạch mã hoá đầu $R = 1/2$, còn ở mạch mã hoá Vâyner Esse $R = 3/4$.

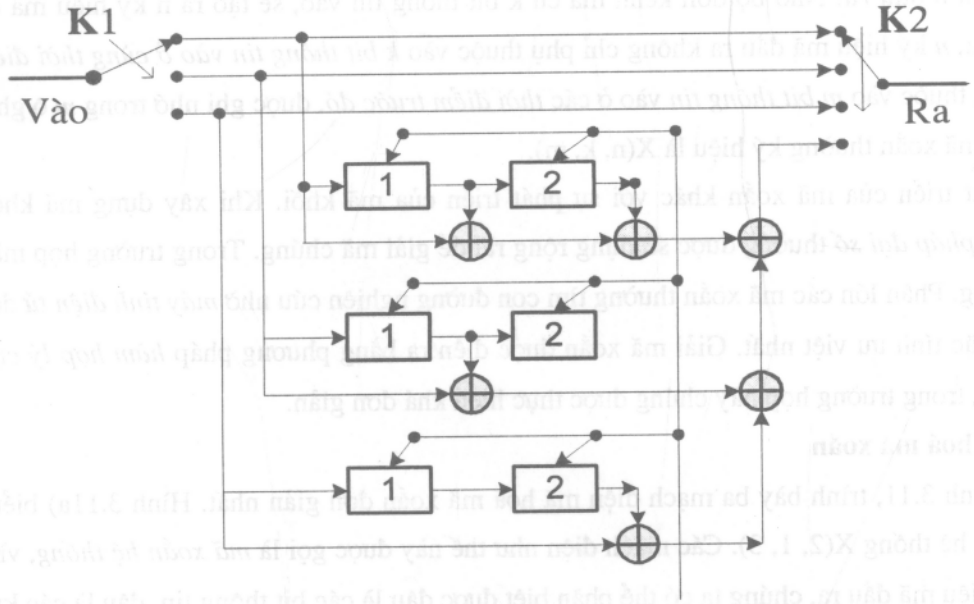
Bây giờ, chúng ta nghiên cứu các phương pháp phân tích hoạt động của mạch mã hoá mã xoắn, để làm cơ sở cho các quá trình giải mã.



a) Mã xoắn hệ thống X(2,1,3)



b) Mã xoắn không hệ thống X (2,1,2)



c) Mã xoắn Vâyner Esse (4, 3, 2)

Hình 3.11. Các mạch điện mã hoá mã xoắn.

7.2. Các phương pháp phân tích hoạt động của mạch mã hoá mã xoắn

Trong phần này chúng ta hạn chế chỉ phân tích hoạt động của mạch mã hoá mã xoắn không hệ thống X(2, 1, 2) như hình 3.11b).

a). Phương pháp bảng

Hoạt động của mạch mã hoá mã xoắn, nói chung, có thể trình bày bằng phương pháp bảng. Trước hết, chúng ta nhận thấy rằng: các ký hiệu mã ở đầu ra bộ XOR M_1, ký hiệu là u_1 được xác định như sau:

Mức logic u_1 = mức logic $a \oplus$ mức logic $b \oplus$ mức logic c, khi xét ở cùng thời điểm t.

Các ký hiệu mã ở đầu ra bộ XOR M_2, ký hiệu là u_2 được xác định như sau:

Mức logic u_2 = mức logic $a \oplus$ mức logic c, khi xét ở cùng thời điểm t.

Khi biết các bit thông tin đầu vào **I**, chẳng hạn **I** = 1101... các ký hiệu mã đầu ra **V**, được xác định như bảng 10.

Bảng 10. Kết quả mã hoá

Thứ tự nhịp	Đầu vào I	Trạng thái các ô ghi dịch 1	2	Mức logic a	b	c	Đầu ra u_1	u_2
0	-	0	0	-	-	-	-	-
1	1	1	0	1	0	0	1	1
2	1	1	1	1	1	0	0	1
3	0	0	1	0	1	1	0	1
4	1	1	0	1	0	1	0	0
5	1	1	1	1	1	0	0	1
...	...	. . .	. .	.	.	.	.	. . .

Kết quả mã hoá từ bảng 10, cho biết **I** = 1101... → **V** = 11- 01- 01- 00 - 01- ..., với mỗi cặp dấu mã, chúng ta sử dụng một dấu cách, để chỉ ra cặp dấu mã ấy được mã hoá tương ứng với bit thông tin nào.

Phương pháp bảng cho kết quả trực quan, dễ hiểu, nhưng chỉ thích hợp cho các trường hợp đơn giản.

b). Phương pháp đa thức kết nối

Với phương pháp này, mỗi dấu mã đầu ra u_1 u_2 được phát sinh nhờ đa thức sinh $g(x)$ tương ứng, khi đã biết đa thức biểu diễn các bit thông tin đầu vào. Các đa thức sinh có dạng chung như sau:

$$g(x) = g_0 + g_1 x + ... + g_m x^m \qquad (3.61)$$

trong đó: - m bậc của đa thức sinh, bằng số lượng ô ghi dịch của mạch mã hoá;

- hệ số g_i được xác định bằng sự kết nối với bộ XOR tương ứng. Hệ số $g_i = 1$, nếu có kết nối của vị trí tương ứng với bộ XOR, ngược lại, hệ số $g_i = 0$.

Với mạch mã hoá như hình 3.11b), chúng ta có:

- Đa thức sinh tạo ra đa thức mã $u_1(x)$, có dạng:

$$g_1(x) = 1 + x + x^2 \quad \text{hay biểu diễn dưới dạng hệ số } G_1 = 111$$

- Đa thức sinh tạo ra đa thức mã $u_2(x)$, có dạng:

$$g_1(x) = 1 + x^2 \quad \text{hay biểu diễn dưới dạng hệ số } G_2 = 101$$

Khi đó:

$$u_1(x) = I(x).g_1(x)$$

$$u_2(x) = I(x).g_2(x) \qquad (3.62)$$

ở đây $I(x)$ - đa thức các bit thông tin đầu vào.

Đa thức của tập dấu mã đầu ra được xác định dưới dạng:

$$V(x) = u_1(x) * u_2(x) \tag{3.63}$$

ở đây dấu * chỉ phép xoắn các hệ số cùng bậc tương ứng của hai đa thức.

Chẳng hạn, chúng ta sử dụng lại các số liệu ở mục trên, tức là các bit thông tin đầu vào có dạng I = 11011..., đa thức biểu diễn chúng được xác định:

$$I(x) = 1 + x + x^3 + x^4 + ...$$

Các đa thức $u_1(x)$ và $u_2(x)$ được xác định như sau:

$$u_1(x) = I(x).g_1(x) = (1 + x + x^3 + x^4 + ...).(1 + x + x^2) = 1 + x + x^3 + x^4 + ...$$

$$x + x^2 + x^4 + x^5 + ...$$

$$\underline{x^2 + x^3 + x^5 + x^6 + ...}$$

$$= 1 + 0x + 0x^2 + 0x^3 + 0x^4 + ...$$

$$u_2(x) = I(x).g_2(x) = (1 + x + x^3 + x^4 + ...).(1 + x^2) = 1 + x + x^3 + x^4 + ...$$

$$\underline{x^2 + x^3 + x^5 + x^6 + ...}$$

$$= 1 + x + x^2 + 0x^3 + x^4 + ...$$

Ở đây cần chú ý một điều là: đa thức biểu diễn các bit thông tin có bậc đến bao nhiêu, thì bậc của các đa thức $u_1(x)$ và $u_2(x)$ cũng chỉ lấy đến bấy nhiêu. Trong thí dụ cụ thể ở trên là bậc 4.

Kết quả xoắn hai hệ số cùng bậc tương ứng của các đa thức $u_1(x)$ và $u_2(x)$ được:

V = 11- 01- 01- 00 - 01-...

Rõ ràng kết quả này phù hợp với phương pháp bảng đã xét ở trên.

Phương pháp đa thức kết nối rất thuận tiện, bởi vì từ dạng cấu trúc của thiết bị mã hoá, ta có thể xác định ngay được các đa thức sinh.

Bảng 11. Các hệ số của đa thức sinh

G1	G2
111	101
1111	1101
11101	10011
111011	10100111
110001	11111001

Trong bảng 11 trình bày các đa thức sinh (ở dạng các hệ số), mà các thiết bị mã hoá mã xoắn thường sử dụng trong thực tiễn.

c). Phương pháp sơ đồ trạng thái

Từ các mạch mã hoá mã xoắn, chúng ta nhận thấy rằng nếu thanh ghi dịch có *m* ô, thì số lượng trạng thái $N = 2^m$. Hơn nữa, cứ mỗi bit thông tin đi vào thì trạng thái các ô ghi dịch sẽ thay đổi

và đưa ra n ký hiệu mã. Có thể biểu diễn mối quan hệ giữa trạng thái các ô ghi dịch với các ký hiệu mã đầu ra nhờ phương pháp sơ đồ trạng thái.

Chúng ta hãy bắt đầu từ trường hợp đơn giản nhất, với mạch mã hoá như hình 3.11b). Mạch này có hai ô ghi dịch, số lượng các trạng thái $N = 2^2 = 4$. Chúng ta ký hiệu các trạng thái ấy là A, B, C và D, với:

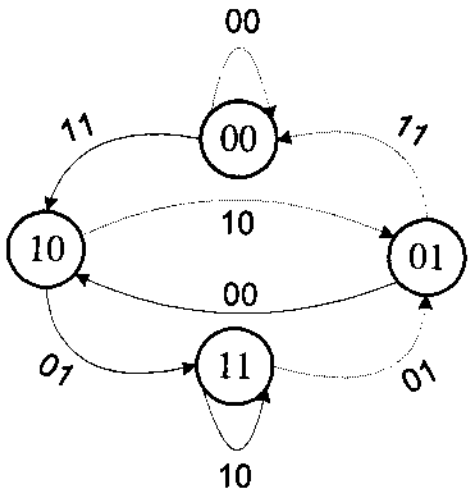
A = 00 - ô 1 có trạng thái 0, ô 2 có trạng thái 0;

B = 10 - ô 1 có trạng thái 1, ô 2 có trạng thái 0;

C = 01 - ô 1 có trạng thái 0, ô 2 có trạng thái 1;

D = 11 - ô 1 có trạng thái 1, ô 2 có trạng thái 1.

Trong phương pháp này, trạng thái các ô ghi dịch được biểu diễn là các hình tròn, bên trong ghi trạng thái tương ứng. Liên hệ giữa các trạng thái được biểu diễn bằng các mũi tên, chiều của mũi tên chỉ chiều thay đổi trạng thái. Hơn nữa, qui định khi bit thông tin đầu vào là 1, mũi tên được vẽ bằng nét liền, còn ngược lại, khi bit thông tin đầu vào là 0, mũi tên được vẽ bằng nét đứt. Trên các mũi tên ghi cặp dấu mã đầu ra, gọi là *từ mã nhánh*. Sơ đồ trạng thái trong trường hợp này được biểu diễn trên hình 3.12.



Hình 3.12. Sơ đồ trạng thái của mã xoắn X(2, 1, 2).

Từ sơ đồ trạng thái ở hình 3.12, nếu chúng ta biết chuỗi các bit thông tin đầu vào, chẳng hạn **I** = 11011..., và chọn trạng thái ban đầu của các ô ghi dịch là A = 00, thì xác định được chuỗi các trạng thái như sau: A→ B→ D→ C→ B→ D... Chuỗi ký hiệu mã đầu ra tương ứng sẽ là: **V** = 11- 01- 01- 00- 01... Nếu chúng ta chọn trạng thái ban đầu của các ô ghi dịch là D = 11, thì xác định được chuỗi các trạng thái như sau: D→ D→ D→ C→ B→ D... Chuỗi ký hiệu mã đầu ra tương ứng sẽ là: **V** = 10- 10- 01- 00- 01...

Từ sơ đồ trạng thái, chúng ta thấy rằng cấu trúc đặc trưng của các mã xoắn có tính chất lặp. Trạng thái của các ô ghi dịch, kéo theo chuỗi mã V có cấu trúc lặp. Chu kỳ lặp lại ngắn nhất là (m + 1), với m là số lượng ô ghi dịch, tiếp theo là (m + 2). Khi không để ý đến các trạng thái lặp trung gian (hoặc sự lặp của các từ mã nhánh trung gian), thì chu kỳ lặp còn dài hơn.

d). Phương pháp sơ đồ cây

Phương pháp sơ đồ trạng thái hoàn toàn có thể xác định được chuỗi các ký hiệu mã đầu ra, khi biết chuỗi bit thông tin đầu vào và trạng thái ban đầu các ô ghi dịch. Tuy nhiên, phương pháp này chưa biểu diễn được *chuỗi ký hiệu mã đầu ra như là hàm của thời gian t*. Phương pháp sơ đồ cây bổ sung thêm tham số thời gian, để xác định *thời gian xác lập các nút của cây* (cũng có nghĩa là *thời gian xác lập trạng thái các ô ghi dịch*). Sơ đồ cây của mã X(2,1,2) được trình bày trên hình 3.13. Sơ đồ này được vẽ như sau: cây được vẽ từ trái sang phải, bắt đầu từ nút gốc, tương ứng với thời điểm t_1, các nút tiếp theo tương ứng với thời điểm t_2, t_3, t_4,... Từ nút gốc mọc ra 2 nhánh (vì mã cơ số 2), một nhánh đi lên trên, một nhánh đi xuống dưới. Quy ước nhánh đi lên trên tương ứng với bit thông tin vào là 0, ngược lại, nhánh đi xuống dưới tương ứng với bit thông tin vào là 1. Trên các nhánh ghi cặp dấu mã đầu ra tương ứng, gọi là "*từ mã nhánh*". Cây tiếp tục mọc dài ra về bên trái, cho đến khi bit thông tin vào sau cùng.

Từ sơ đồ cây hình 3.13, nếu biết các bit thông tin đầu vào **I** = 1101... Bắt đầu từ nút gốc (tương ứng với trạng thái ban đầu các ô ghi dịch là A = 00) đi theo đường mũi tên nét đứt trên hình vẽ, chúng ta xác định được chuỗi ký hiệu mã đầu ra **V** = 11-01-01-00-...

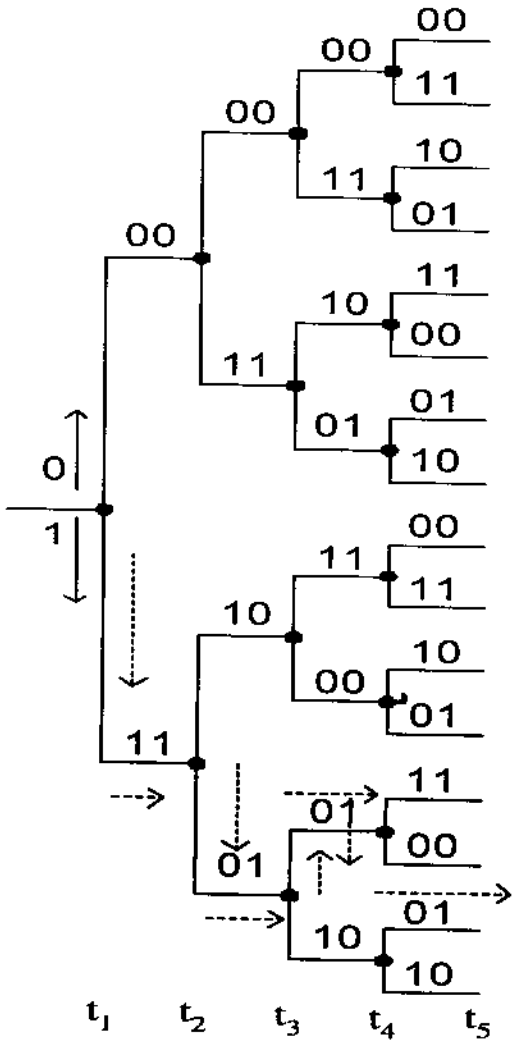
Nếu ký hiệu độ dài chuỗi bit thông tin đầu vào là k, thì tổng số nhánh của cây sẽ là:

$$\textit{Số nhánh} = \sum_{i=1}^{k} 2^i$$

Còn số nhánh sau thời điểm có bit vào thứ k là 2^k. Như vậy, cây mã càng phức tạp- càng nhiều nhánh, khi số bit đầu vào tăng lên.

Một đặc điểm khác của cây mã là tối thiểu sau *m+1* lần phân nhánh (m - số ô ghi dịch), thì cấu trúc của cây lặp lại. Mã xoắn X(2, 1, 2) có m = 2, nên tối thiểu sau ba lần phân nhánh, cấu trúc cây lặp lại (bắt đầu từ thời điểm t_4).

Sơ đồ cây không chỉ sử dụng để biểu diễn đặc trưng mã khi *mã hoá*, mà còn dùng để *giải mã*.



Hình 3.13. Sơ đồ cây của mã xoắn X(2, 1, 2).

e). Phương pháp sơ đồ lưới

Sơ đồ lưới được xây dựng từ sự kết hợp của sơ đồ cây và sơ đồ trạng thái, vừa xét ở trên. Cũng như đã biết ở trên, các giá trị t_i để ghi nhận thời điểm xác lập trạng thái các ô ghi dịch.

Lưới gồm các nút và các nhánh. Nút biểu diễn trạng thái ô ghi dịch. Nhánh lưới được biểu diễn bằng các mũi tên, nối các nút của lưới, chiều của mũi tên xác định hướng thay đổi trạng thái ô ghi dịch. Quy ước:

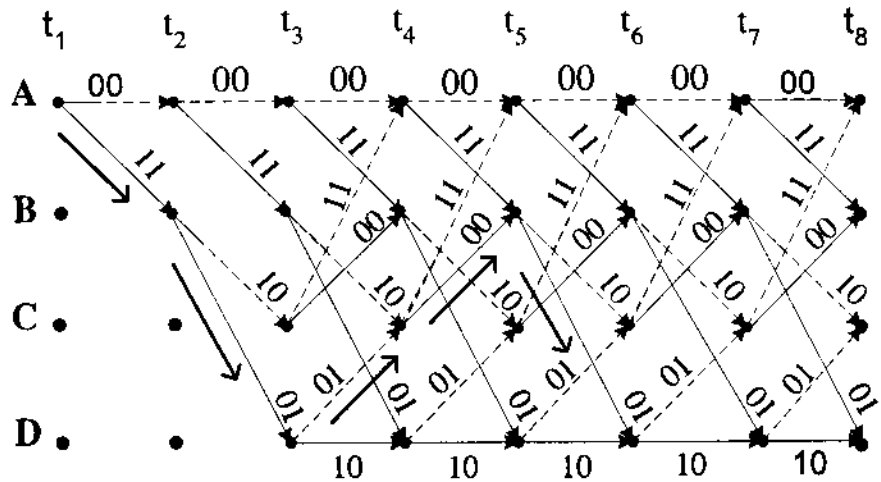
- Các nút biểu diễn cùng một trạng thái, nhưng xác lập ở các thời điểm t_i khác nhau, được vẽ trên cùng một hàng,
- Các nút biểu diễn các trạng thái khác nhau, nhưng có thể xác lập ở cùng thời điểm t_i, được vẽ trên cùng một cột,

- Mũi tên vẽ bằng nét liền, khi bit thông tin vào là 1. Ngược lại, mũi tên vẽ bằng nét đứt, khi bit thông tin vào là 0. Trên các nhánh ghi cặp dấu mã đầu ra tương ứng với sự thay đổi trạng thái, và gọi là "*từ mã nhánh*"

Sơ đồ lưới của mã xoắn X(2, 1, 2) được vẽ trên hình 3.14.

Từ sơ đồ lưới, khi biết chuỗi các bit thông tin vào và chọn một trạng thái ban đầu của các ô ghi dịch, chúng ta hoàn toàn xác định được chuỗi các ký hiệu mã đầu ra. Chẳng hạn, các bit thông tin vào **I** = 11011..., chọn trạng thái ban đầu của các ô ghi dịch là A = 00, chúng ta xác định được chuỗi các trạng thái sau: $A(t_1) \rightarrow B(t_2) \rightarrow D(t_3) \rightarrow C(t_4) \rightarrow B(t_5) \rightarrow D(t_6)$, tương ứng chúng ta xác định chuỗi ký hiệu mã đầu ra: V= 11- 01- 01- 00 - 01...

Sơ đồ lưới không chỉ sử dụng để biểu diễn đặc trưng mã khi *mã hoá*, mà còn dùng để *giải mã*.



Hình 3.14. Sơ đồ lưới của mã xoắn X(2, 1, 2).

7.3. Giải mã xoắn theo thuật toán Viterbi

Ký hiệu chuỗi mã thu là P, còn chuỗi mã phát là V.

Phương pháp giải mã xoắn tốt nhất là *phương pháp sự giống nhau cực đại*, mà khi giải mã cần xác định *bộ các xác suất có điều kiện* $p(P/V_i)$ theo tất cả các khả năng có thể của chuỗi mã phát V_i. Ứng với chuỗi mã phát V_i nào đó, mà *xác suất có điều kiện* $p(P/V_i)$ đạt giá trị cực đại (trong trường hợp P không lỗi, thì $p(P/V_i) = 1$), thì chuỗi mã thu $P \equiv V_i$ đó, hay nói cách khác: *chuỗi mã thu P giống chuỗi mã phát V_i ấy cực đại.*

Đối với *kênh đối xứng nhị phân không nhớ* (kênh trong đó xác suất truyền các ký hiệu 0 và 1, cũng như xác suất lỗi dạng $0 \rightarrow 1$ và $1 \rightarrow 0$ là như nhau, đồng thời các lỗi ở ký hiệu j và i không phụ thuộc lẫn nhau) giải mã theo phương pháp *sự giống nhau cực đại* dẫn đến giải mã theo *khoảng cách Hamming cực tiểu*. Khi đó cần xác định khoảng cách Hamming giữa chuỗi mã thu P với tất cả các khả năng của chuỗi mã phát V_i. Ứng với chuỗi mã phát V_i nào đó, mà $d(P, V_i)$

cực tiểu (trong trường hợp P không lỗi, thì $d(P, V_i) = 0$), thì chuỗi mã thu được xác định theo $P \equiv V_i$ đó.

Tuy nhiên, khi các tham số của mạch mã hóa là lớn, đồng thời chuỗi bit thông tin đầu vào là dài thì phương pháp giải mã này có vẻ như không thực tiễn. Rất may là, như đã biết từ các mục trên, *cấu trúc đặc trưng của chuỗi mã xoắn có tính chất lặp* (cấu trúc của chuỗi mã sẽ lặp lại sau một số từ mã nhánh nhất định, chu kỳ lặp tối thiểu phụ thuộc vào tham số m - số lượng ô nhớ trong mạch điện mã hóa), điều này cho phép thuật toán trở thành có tính thực tiễn.

Ý tưởng đầu tiên của phương pháp giải mã này được Viterbi đưa ra vào năm 1967. Thuật toán làm việc như sau:

Giả thử ở đầu vào của mạch giải mã có chuỗi mã thu P dài tùy ý, nhưng hữu hạn. Chuỗi mã thu P được cắt ra thành những *mảnh có độ dài b*, vượt chiều dài của tham số m – số lượng ô ghi dịch trong mạch mã hóa. Tham số b được gọi là *chiều sâu giải mã*. Xác định trọng số Hamming giữa từ mã P (trong khuôn khổ chiều sâu giải mã) với tất cả các khả năng của chuỗi mã phát V_i, cũng trong khuôn khổ chiều sâu giải mã ấy, với qui ước trạng thái ban đầu của các ô nhớ (thông thường trạng thái ban đầu của các ô nhớ ứng với lần xác định thứ nhất là trạng thái toàn mức 0). Tương ứng với chuỗi mã phát V_i nào, mà trọng số Hamming của *con đường*, được xác định là tổng trọng số Hamming của các nhánh thích hợp, là cực tiểu, thì lấy $P \equiv V_i$ ấy. Từ đó, khôi phục được các bit thông tin phía phát đã gửi đi tương ứng với chiều sâu giải mã đang xét.

Quá trình giải mã sẽ *lặp lại* với những mảnh của chuỗi mã thu tiếp theo, cho đến khi kết thúc của chuỗi mã P.

Như vậy, thuật toán giải mã xoắn theo Viterbi *gia công liên tiếp các mảnh* của chuỗi mã thu trong khuôn khổ của chiều sâu giải mã. Từ đó xác định được một con đường (trên sơ đồ cây hay trên sơ đồ lưới) có trọng số Hamming cực tiểu, tính từ điểm xuất phát cho đến điểm cuối cùng, được gọi là *đường đi hợp lý* hay *tối ưu*. Trên con đường đó *chuỗi mã thu P giống chuỗi mã phát V_i cực đại*. Chúng chỉ khác nhau ở các ký hiệu mắc lỗi.

Chúng ta trình bày hoạt động của thuật toán giải mã xoắn theo Viterbi bằng một thí dụ đơn giản nhất - mạch mã hóa mã xoắn không hệ thống X(2,1,2) như hình 3.11b) và quá trình giải mã sử dụng sơ đồ lưới như hình 3.14.

Giả thử mảnh đầu tiên của chuỗi mã thu có dạng P_1 = 11- 01- 01- 10 - 01 (Chiều sâu giải mã trong trường hợp này được chọn bằng: b = m + 3 = 5). Kết quả quá trình tính trọng số Hamming tất cả các nhánh được trình bày trên hình 3.15a).

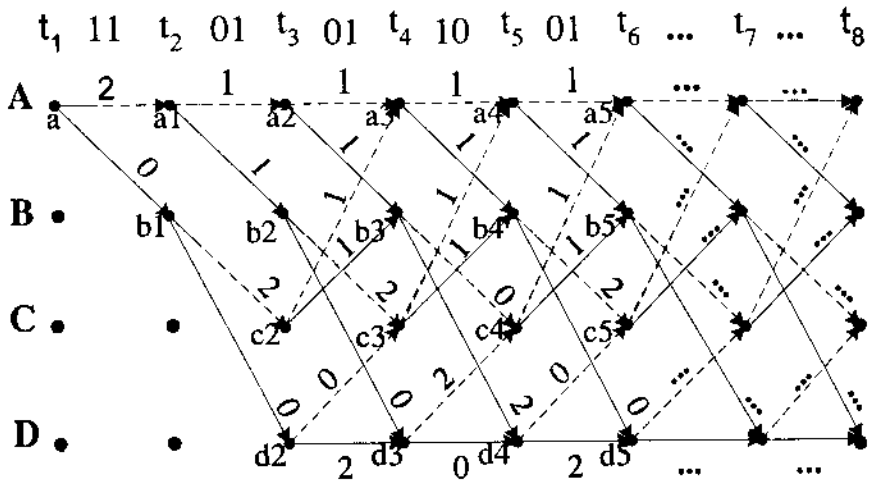
Tại các nút A = 00, ứng với các thời điểm t_i (i = 1, 2, 3, 4, 5, 6), chúng ta ký hiệu các nút là a, a_1, a_2 , a_3, a_4, a_5 . Cũng tương tự như vậy cho các nút B = 10, các nút C = 01, các nút D = 11. Trạng thái ban đầu của các ô ghi dịch là A = 00. Trọng số Hamming của mỗi nhánh được xác định theo công thức:

Wnhánhi = W(từ mã nhánhi ⊕ từ mã thu, - *tính ở cùng khoảng thời gian* t_i - t_{i+1}).

1). Tại thời điểm ban đầu t_1, trạng thái các ô ghi dịch a = 00. Từ trạng thái 00 chỉ có thể dẫn đến ở thời điểm t_2 hai trạng thái: hoặc 00 (a1) hoặc 10 (b1), tùy thuộc bit thông tin vào mạch mã hóa là 0 hay 1 và trên sơ đồ lưới có các "từ mã nhánh" hoặc 00 hoặc 11, còn từ mã thu được tương ứng với khoảng thời gian đó là 11. Kết quả tính trọng số các nhánh như sau:

Waa1 = W(00 ⊕ 11) = 2, Wab1 = W(11 ⊕ 11) = 0,

Kết quả trình bày trên hình 3.15b1).



Hình 3.15a). Kết quả tính trọng số Hamming tất cả các nhánh.

2).Ở' thời điểm t_2 tồn tại một trong hai trạng thái hoặc 00(a1), hoặc 11(b1). Chuyển sang thời điểm t_3, tùy thuộc bit thông tin vào mạch mã hóa là 0 hay 1, mà có thể có bốn trạng thái tất cả: a1→ a2 hay b2, hoặc b1→ c2 hay d2, với các từ mã nhánh tương ứng, còn từ mã thu được tương ứng với khoảng thời gian đó là 01. Kết quả tính trọng số các nhánh như sau:

Wa1a2 = W(00 ⊕ 01) = 1, Wa1b2 = W(11 ⊕ 01) = 1,

Wb1c2 = W(10 ⊕ 01) = 2, Wb1d2 = W(01 ⊕ 01) = 0,

Kết quả trình bày trên hình 3.15b2).

3). Ở' thời điểm t_3 tồn tại một trong bốn trạng thái là a2, b2, c2, d2. Khi chuyển sang thời điểm t_4, cũng sẽ tồn tại một trong bốn trạng thái a3, b3, c3, d3, với các từ mã nhánh tương ứng, còn từ mã thu được trong khoảng thời gian t_3→ t_4 là 01. Kết quả tính trọng số các nhánh như sau:

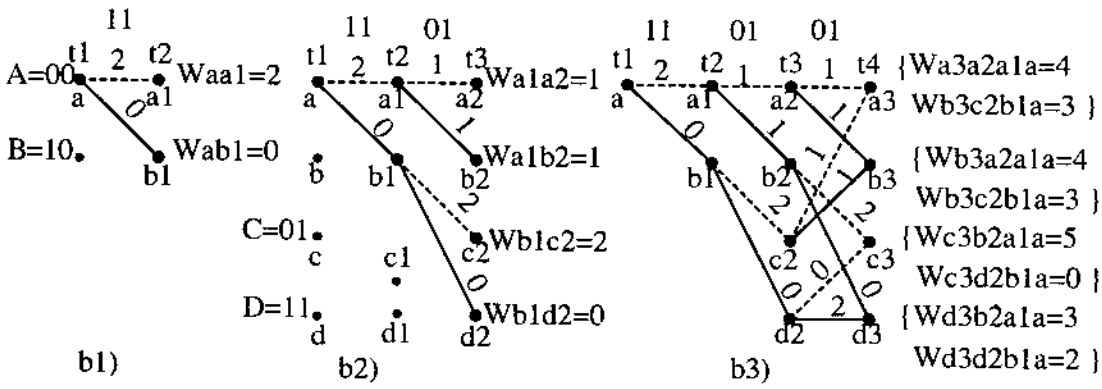
Wa2a3 = W(00 ⊕ 01) = 1, Wa2b3 = W(11 ⊕ 01) = 1,

Wb2c3 = W(10 ⊕ 01) = 2, Wb2d3 = W(01 ⊕ 01) = 0,

Wc2a3 = W(11 ⊕ 01) = 1, Wc2b3 = W(00 ⊕ 01) = 1,

Wd2c3 = W(01 ⊕ 01) = 0, Wd2d3 = W(10 ⊕ 01) = 2,

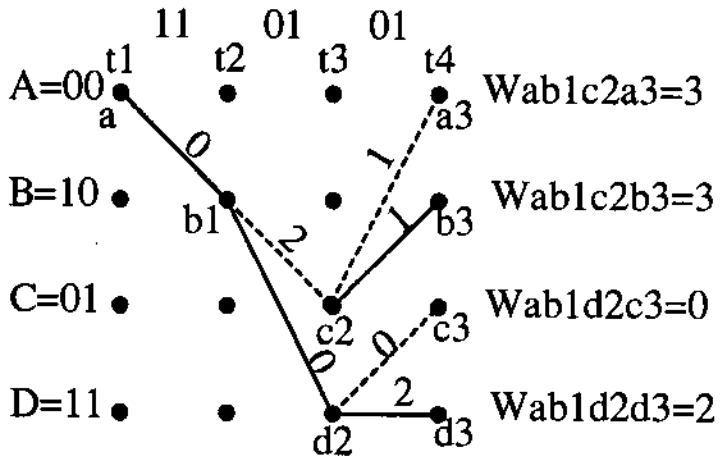
4). Ở' thời điểm t_4 chúng ta nhận thấy là mỗi nút đều có hai nhánh dẫn vào. Vì thế thuật toán giải mã thực hiện *tính ngược* trọng số các con đường tính từ thời điểm t_4 trở về thời điểm ban đầu t_1. Kết quả tính ngược các trọng số được trình bày trên hình 3.15b3).



Hình 3.15b). Các bước tính trọng số Hamming các nhánh và bước tính ngược.

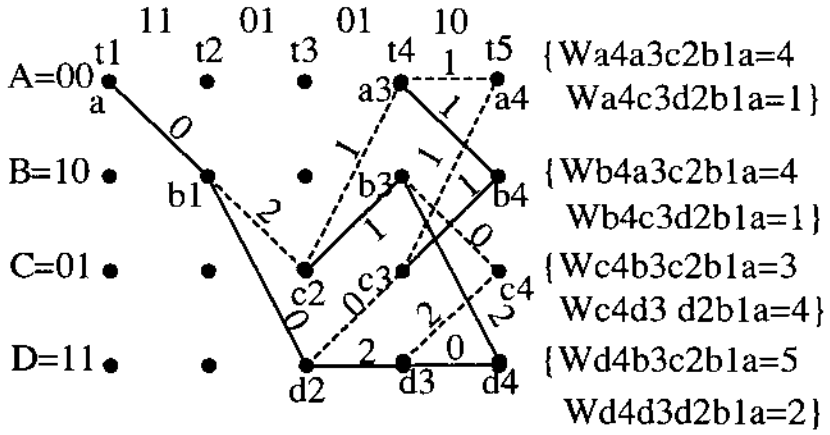
Sau kết quả tính ngược, thuật toán giải mã sẽ *lược bỏ* con đường nào có tổng trọng số Hamming lớn hơn, với mỗi nút chỉ giữ lại một con đường dẫn tới nó. Nếu gặp trường hợp cả hai con đường dẫn tới một nút đều có cùng tổng trọng số Hamming, thì thuật toán giải mã giữ lại cả hai, chờ xử lý ở bước tiếp theo.

Kết quả lược bỏ được trình bày trên hình 3.15c).

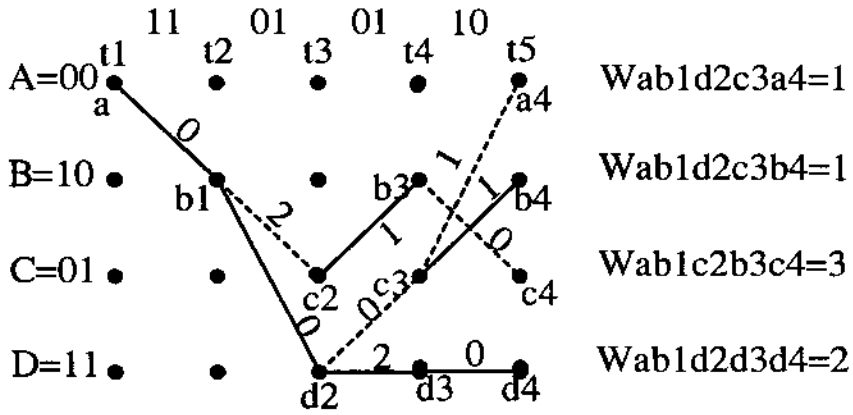


Hình 3.15c). Kết quả loại bỏ các đường có trọng số Hamming lớn hơn đến các nút a3, b3, c3, d3 ở thời điểm t4.

Các bước tiếp theo, từ thời điểm t_4 đến thời điểm t_5 tính toán tương tự. Kết quả trình bày trên hình 3.15d) và hình 3.15e).



Hình 3.15d). Kết quả tính trọng số Hamming các nhánh từ t4 - t5.

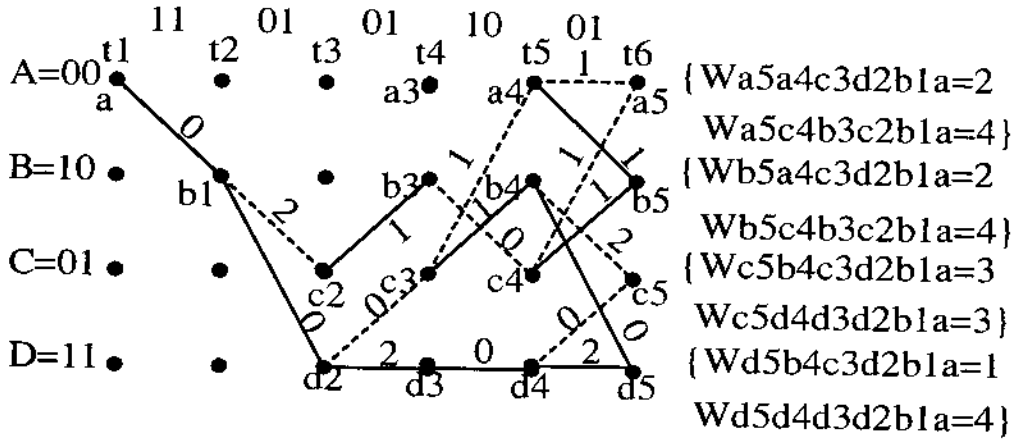


Hình 3.15e). Kết quả loại bỏ các con đường có trọng số Hamming lớn hơn đối với các nút ở thời điểm t5.

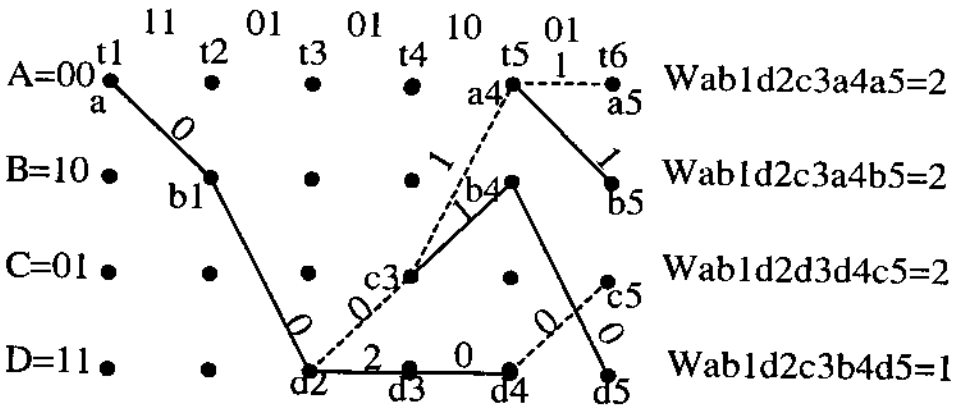
Từ thời điểm t_5 đến thời điểm t_6, cũng thực hiện tính toán tương tự như các bước ở trên. Kết quả tính toán được trình bày trên hình 3.15f) và hình 3.15g).

Kết quả lược bỏ các con đường có tổng trọng số Hamming lớn hơn được trình bày trên hình 3.15g).

Từ hình 3.15g) chúng ta thấy các nút a5, b5, c5, d5, mỗi nút có một con đường dẫn đến với tổng trọng số Hamming tương ứng: Wa5 = 2, Wb5 = 2, Wc5 = 2, Wd5 = 1. Thuật toán đã thực hiện các tính toán xong đối với một chiều sâu giải mã và xác định được một đoạn đường có tổng trọng số Hamming nhỏ nhất. Đó là đoạn đường bắt đầu từ nút a và kết thúc ở nút d5: a→ b1→ d2→ c3→ b4→ d5, có tổng trọng số Hamming cực tiểu và bằng 1.



Hình 3.15f). Kết quả tính trọng số Hamming các nhánh từ thời điểm t5 - t6.



Con đường hợp lý (tối ưu): a- b1- d2- c3- b4- d5

Hình 3.15g). Kết quả loại bỏ các con đường có tổng trọng số Hamming các nhánh lớn hơn và chọn ra con đường hợp lý.

Sau khi xác định được *đoạn đường hợp lý*, mảnh đầu tiên của chuỗi mã P được xác định bằng *các từ mã nhánh trên đường đi hợp lý*, tức là:

$$\overline{P} = 11 - 01 - 01 - 00 - 01$$

Chúng ta cũng nhận thấy rằng trên đường đi hợp lý ở nhánh nào có trọng số Hamming khác 0, tương ứng với khoảng thời gian đó từ mã P mắc lỗi. Trong thí dụ cụ thể này, đó là khoảng thời gian từ $t_4 \rightarrow t_5$.

Các bít thông tin gửi đi từ phía phát được khôi phục theo qui tắc sau: đi trên đường hợp lý từ điểm khởi đầu cho đến điểm cuối: nếu đi trên *đường nét liền* của sơ đồ lưới, bit thông tin được khôi phục *bằng 1*, còn ngược lại đi trên *đường nét đứt* của sơ đồ lưới, bit thông tin được khôi phục *bằng 0*. Trong thí dụ cụ thể này, các bít thông tin được khôi phục có dạng:

I = 1- 1- 0 - 1- 1 (hai nhánh nét liền - tiếp đến nhánh nét đứt - rồi hai nhánh nét liền).

Quá trình *giải mã tiếp theo* được bắt đầu *từ nút d5* và với cùng chiều sâu giải mã, toàn bộ các tính toán được lặp lại tương tự như trên. Vì đặc điểm này, mà thuật toán Viterbi được gọi là *thuật toán giải mã lặp*.

Với mã xoắn X(2, 1, 2), khi chọn chiều sâu giải mã như trên, nếu lỗi không thuộc cặp ký hiệu mã cuối, *chắc chắn giải mã đúng cho một lỗi*. Trong trường hợp *mắc hai lỗi*, tùy theo vị trí của lỗi, có trường hợp giải mã đúng, có trường hợp không; nếu tăng chiều sâu giải mã thêm 1, có thể giải mã đúng được. Chiều sâu giải mã là một *đại lượng ngẫu nhiên*, không thể định trước được, nó phụ thuộc vào xác suất xuất hiện lỗi, cũng như vào thời điểm xuất hiện lỗi. Vấn đề chọn tối ưu chiều sâu giải mã là khó. Thông thường chọn $m < b \leq m + 3$.

Chúng ta cần xác định khối lượng tính toán và dung lượng nhớ của thuật toán Viterbi, trong khuôn khổ chiều sâu giải mã. Số trạng thái ở thời điểm t_i là 2^m, mỗi nút ở thời điểm t_i có 2 nhánh từ 2 nút ở thời điểm t_{i-1} dẫn đến. Như vậy, từ thời điểm t_{i-1} - t_i có tất cả $2.2^m = 2^{m+1}$ nhánh, cho nên khối lượng tính toán cũng như dung lượng nhớ cần 2^{m+1}. Trong khoảng chiều sâu giải mã cần khối lượng phép toán tính trọng số Hamming các nhánh và cũng như số lượng thanh ghi để ghi nhớ chúng, được tính theo công thức:

$$T = [2^{m+1} \times (b - m) \times) + (2 + 2^2 + \ldots + 2^m)]. \qquad (3.64)$$

Chúng ta nhận thấy rằng trong m khoảng thời gian đầu, số lượng các nhánh được tính như sau: từ thời điểm t_1 - t_2 chỉ có 2 nhánh, từ thời điểm t_2 - t_3 có 2^2 nhánh,..., từ thời điểm t_m - t_{m+1} có 2^m nhánh. Đó là một phần để giải thích công thức trên.

Chẳng hạn, khi số lượng ô nhớ m = 2, còn chiều sâu giải mã b = 5, thì số lượng thanh ghi dịch của mạch giải mã để ghi nhớ trọng số Hamming các nhánh, được tính theo công thức trên là:

$$T(m = 2, b = 5) = [2^3 \times (5 - 2) + (2 + 2^2) = 24 + 6 = 30$$

Mạch điện giải mã theo thuật toán Viterbi cần T thanh ghi dạng ROM để ghi nhớ từ mã nhánh trong khuôn khổ chiều sâu giải mã. Chiều dài mỗi thanh ghi được xác định như sau:

- n ô để ghi nhớ từ mã nhánh,
- 2 ô để ghi trạng thái nhánh: "nét liền" = 1, "nét đứt" = 0,
- 2m ô để ghi: trạng thái nút khởi đầu của nhánh và trạng thái nút kết thúc của nhánh
- Kết quả chiều dài thanh ghi là: L = n +2 +2m.

Mạch điện giải mã theo thuật toán Viterbi cũng cần T thanh ghi dạng RAM để ghi nhớ trọng số Hamming của các nhánh trong khuôn khổ chiều sâu giải mã.

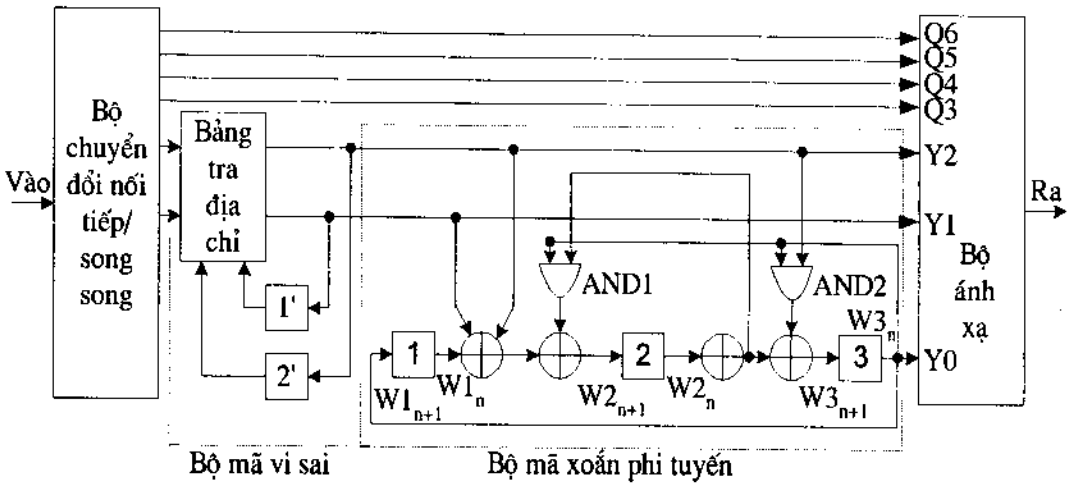
Một vấn đề khác khi giải mã theo Viterbi là vấn đề bảo đảm đồng bộ từ mã thu được với từ mã nhánh, tức là phải xác định đúng đồng bộ thời điểm khởi đầu, để tính trọng số các nhánh được chính xác.

Ngoài thuật toán Viterbi, mã xoắn còn được giải bằng thuật toán Fano.

8. MÃ TCM (TRELLIS CODED MODULATION)

Trên hình 3.16 biểu diễn mạch điện mã hóa của mã TCM theo chuẩn V.32bis. Mạch mã hóa bao gồm:

- Bộ chuyển đổi nối tiếp thành song song ở đầu vào;
- Bộ mã hóa vi sai;
- Bộ mã xoắn phi tuyến;
- Bộ ánh xạ đầu ra.



Hình 3.16. Mạch điện mã hóa của mã TCM.

Bộ mã xoắn phi tuyến có tốc độ 2/3 với 3 phần tử nhớ, đầu ra của phần tử nhớ thứ ba được đưa hồi tiếp về đầu vào của phần tử nhớ thứ nhất. Quan hệ giữa các trạng thái hiện tại và trạng thái kế sau được mô tả bằng phương trình:

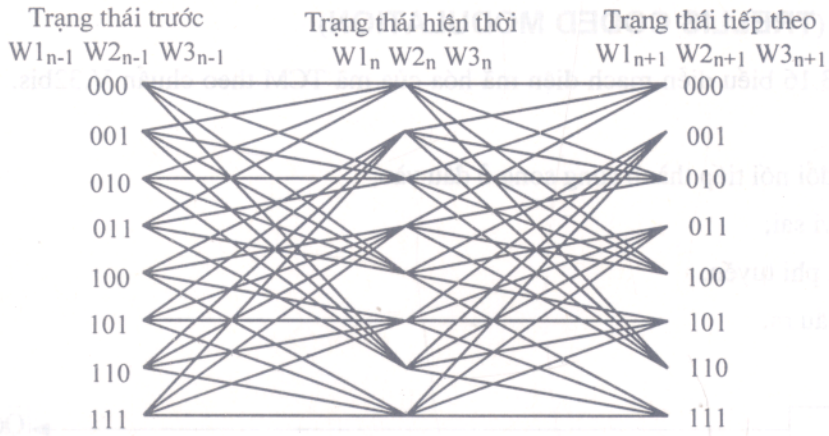
$$W1_{n+1} = W3_n$$
$$W2_{n+1} = W1_n \oplus Y1_n \oplus Y2_n \oplus \left[\left(W2_n \oplus Y2_n\right) \otimes W3_n\right]$$
$$W3_{n+1} = \left(W2_n \oplus Y2_n\right) \oplus \left(W3_n \otimes Y1_n\right)$$

Theo đó, biểu diễn hình học sự chuyển đổi trạng thái sẽ có dạng như trên hình 3.17.

Nhận xét thấy các trạng thái hiện tại 0, 1, 2, 3 luôn chuyển tới các trạng thái tiếp sau là 0, 2, 4, 6; còn đối với 4, 5, 6,7 là 1, 3, 5, 7. Đánh giá này sẽ cho phép thực hiện một thuật toán giải mã (Viterbi) hiệu quả.

Bộ ánh xạ có tác dụng dãn cách các điểm biểu diễn tín hiệu trong mặt phẳng biên độ-pha (cũng gọi là biểu diễn chòm sao), cũng đồng nghĩa với việc nâng cao khoảng cách d_0 của bộ mã.

Mã TCM được xây dựng theo nguyên tắc số ký hiệu dư chỉ là 1, nhưng hiệu quả chống nhiễu cao.

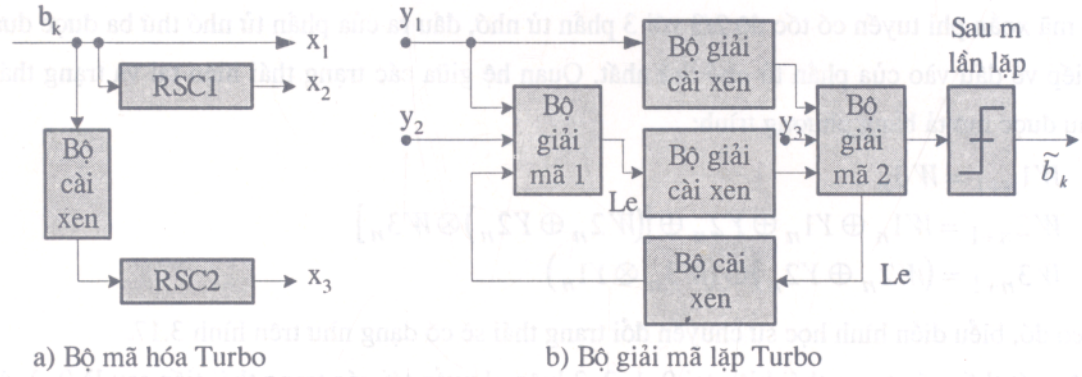


Hình 3.17. Lưu đồ chuyển đổi trạng thái của mã TCM.

9. MÃ TURBO

9.1). Nguyên lý chung mã Turbo

Mã Turbo cũng thường được gọi là mã tích Turbo (turbo product code), vì cấu trúc của chúng gồm xâu chuỗi hai bộ mã xoắn hệ thống thành phần (RSC1, RSC2) với bộ cài xen chen giữa (Hình 3.18a). Kết quả xâu chuỗi tạo ra bộ mã mạnh hơn nhiều, ngay cả khi các bộ mã thành phần yếu. Bộ cài xen có tác dụng xáo trộn các bit trước khi đưa vào bộ mã thành phần thứ hai, để nâng cao hiệu quả chống lỗi của toàn bộ mã. Bộ mã Turbo được đánh giá là bộ mã khối hiệu quả nhất hiện nay.



a) Bộ mã hóa Turbo b) Bộ giải mã lặp Turbo

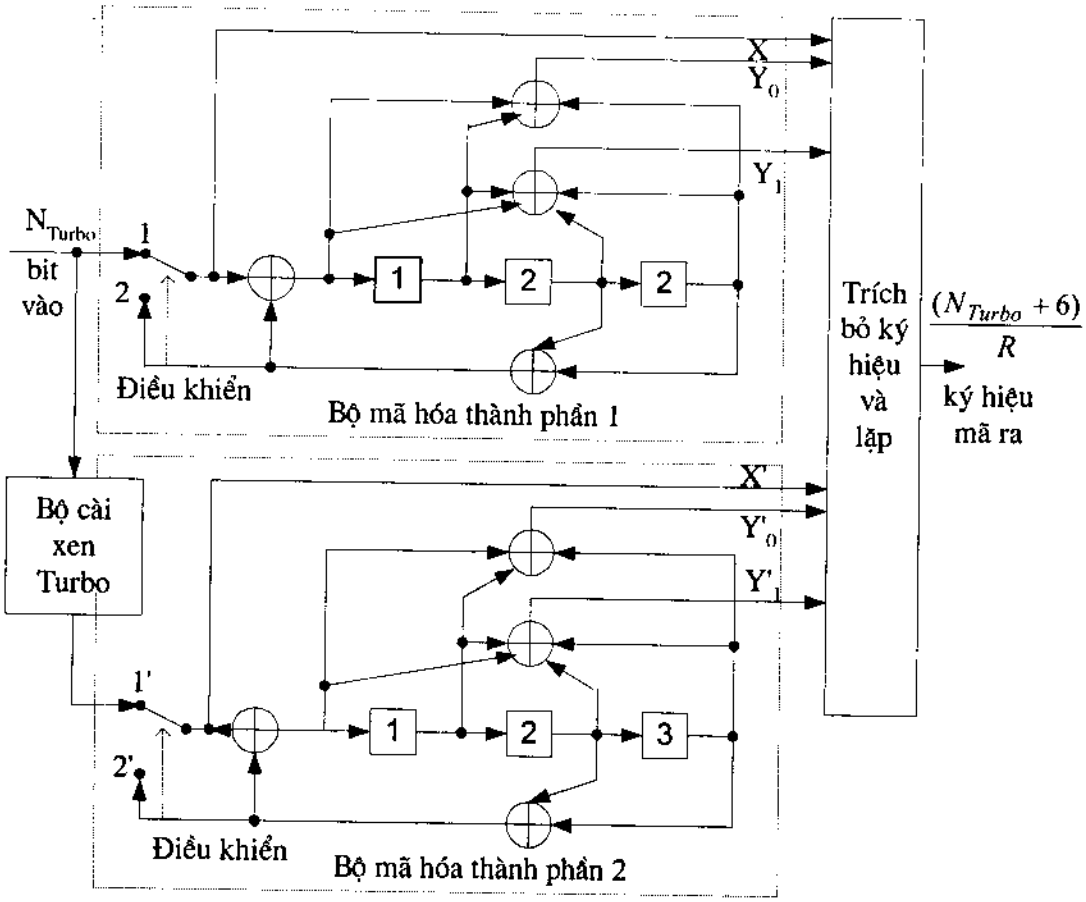
Hình 3.18 Bộ mã hóa và giải mã lặp Turbo.

Nguyên tắc cơ bản của thuật toán giải mã Turbo là giải mã lặp giữa các bộ giải mã thành phần (Hình 3.18b). Quá trình giải mã được thực hiện lần lượt với từng bộ giải mã thành phần. Mỗi bộ giải mã thành phần sẽ gửi các *ước đoán hậu nghiệm* (*posteriority*) về khả năng *likelihood* của chuỗi bit đã được giải mã, Le, đến bộ giải mã kia và sử dụng mức độ giống nhau nhất của bộ mã kia làm *khả năng tiên nghiệm* (*priority*). Các bit dữ liệu không được mã, x_1, sẽ bị ảnh hưởng nhiều hơn bởi kênh truyền so với các bit được mã, x_2 và x_3, sẽ được đưa vào mỗi bộ giải mã để

khởi đầu các khả năng tiên nghiệm. Quá trình giải mã này được lặp lại giữa các đầu ra các bộ giải mã thành phần cho đến khi đạt được hiệu quả sửa lỗi cao nhất thì thôi. Mã Turbo cũng có thể giải mã bằng thuật toán MAP (Max a Posteriori), xác định chính xác đến từng bit, chứ không phải chuỗi bit, nhưng phức tạp hơn.

9.2). Cấu trúc bộ mã hóa Turbo

Trên hình 3.19 biểu diễn mạch điện mã hóa của mã Turbo sử dụng trong CDMA-2000, với các tỷ lệ mã R = 1/2, 1/3, 1/4. Mạch điện bao gồm:



Hình 3.19. Mạch điện mã hóa mã Turbo.

- Hai bộ mã hóa mã xoắn hệ thống hồi quy (Recursive Systematic Covolutional);
- Bộ cài xen trong;
- Bộ trích bỏ ký hiệu và lặp ở đầu ra.

Bộ mã hóa Turbo làm việc như sau:

Chuyển mạch 1 (bộ mã hóa thành phần 1) ở vị trí 1 và dịch từng bit thông tin của khối N_{turbo} bit (giá trị N_{turbo} phụ thuộc vào kích cỡ của bộ cài xen) vào bộ mã hóa thành phần 1; sau đó chuyển mạch ở vào vị trí 2 và dịch từng ký hiệu đuôi trong số ba ký hiệu đuôi của bộ mã hóa thành phần 1 ra ngoài; rồi ngừng dịch cho ba ký hiệu đuôi của bộ mã hóa thành phần 2 dịch ra ngoài.

Chuyển mạch II (bộ mã hóa thành phần 2) ở vị trí 1' (cùng thời điểm với chuyển mạch I) và dịch từng bit thông tin của khối N_{turbo} bit ghi vào bộ cài xen trong, rồi tiếp theo đọc từng bit bộ cài xen theo một thuật toán định trước vào bộ mã hóa thành phần 2; sau đó ngừng dịch, để dịch ba ký hiệu đuôi của bộ mã hóa thành phần 1; rồi chuyển mạch ở vị trí 2' và dịch từng ký hiệu đuôi của bộ mã hóa thành phần 2 ra ngoài.

9.3). Kết cuối mã hóa Turbo

Bộ trích bỏ ký hiệu số liệu và lặp làm việc tuân theo quy định ở bảng 12. Giá trị "0" ở mẫu trích bỏ có nghĩa ký hiệu số liệu này bị bỏ qua (không đưa ra ngoài), còn giá trị "1" ở mẫu trích bỏ có nghĩa là ký hiệu số liệu này được đưa ra ngoài.

Bảng 12. Mẫu trích bỏ cho các chu kỳ của ký hiệu số liệu

Đầu ra	Tỷ lệ mã		
	1/2	1/3	1/4
X	11	11	11
Y_0	10	11	11
Y_1	00	00	10
X'	00	00	00
Y'_0	01	11	01
Y'_1	00	00	11

Chú ý: Đối với từng tỉ lệ mã, bảng trích bỏ sẽ được đọc từ trên xuống dưới, sau đó từ trái qua phải.

Đối với mỗi bit thông tin đầu vào, đầu ra của các bộ mã hóa thành phần sẽ đưa ra chuỗi ký hiệu số liệu theo chu kỳ: X, Y_0, Y_1, X', Y'_0, Y'_1. Chu kỳ này sẽ được lặp cho đến hết N_{turbo} bit vào. Tuy nhiên, theo quy ước ở bảng 12, thực chất ký hiệu số liệu X' luôn luôn bị bỏ qua. Như vậy, mã hóa Turbo tạo ra N_{turbo}/R ký hiệu số liệu ở đầu ra.

Bảng 13. Mẫu trích bỏ cho các chu kỳ bit đuôi

Đầu ra	Tỷ lệ mã		
	1/2	1/3	1/4
X	111000	111000	111000
Y_0	111000	111000	111000
Y_1	000000	000000	111000
X'	000111	000111	000111
Y'_0	000111	000111	000111
Y'_1	000000	000000	000111

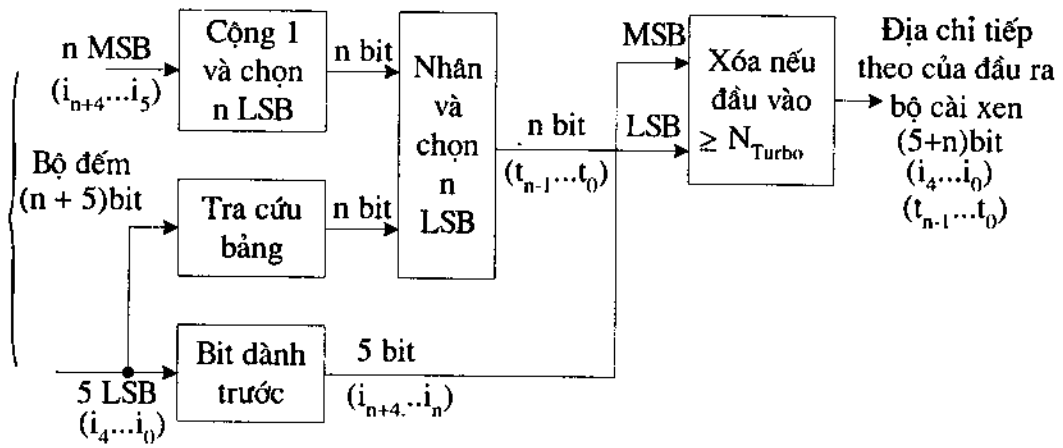
Tiếp theo, mã hóa Turbo tạo ra 6/R ký hiệu đuôi ở đầu ra trong 6 chu kỳ. Mẫu trích bỏ cho các chu kỳ bit đuôi được trình bày trong bảng 13. Đối với mã Turbo 1/2, các ký đuôi ra đối với ba chu kỳ ký hiệu đuôi đầu tiên sẽ là XY_0, còn các ký hiệu đuôi ra đối với ba chu kỳ bit đuôi còn lại sẽ là $X'Y'_0$. Đối với mã Turbo 1/3, các ký đuôi ra đối với ba chu kỳ ký hiệu đuôi đầu tiên sẽ là XXY_0, còn các ký hiệu đuôi ra đối ba chu kỳ bit đuôi còn lại sẽ là $X'X'Y'_0$. Đối với mã Turbo 1/4, các ký đuôi ra đối với ba chu kỳ ký hiệu đuôi đầu tiên sẽ là XXY_0Y_1, còn các ký hiệu đuôi ra đối với ba chu kỳ bit đuôi còn lại sẽ là $X'X'Y'_0Y'_1$.

Chú ý: Đối với mã Turbo 1/2, bảng trích bỏ được đọc từ trên xuống dưới, sau đó từ trái qua phải. Đối với mã Turbo 1/3, 1/4, bảng trích bỏ được đọc từ trên xuống dưới, lặp với X và X', sau đó từ trái qua phải.

Kết quả, mã hóa Turbo tạo ra $\frac{(N_{Turbo}+6)}{R}$ ký hiệu mã đầu ra. Từ nguyên tắc làm việc của mã Turbo, chúng ta thấy rằng mã Turbo là *sự kết hợp của mã xoắn và mã khối.*

9.4). Các bộ cài xen Turbo

Bộ cài xen Turbo là một bộ phận của bộ mã hóa Turbo. Bộ cài xen Turbo là một ma trận bao gồm (M hàng× N cột) = N_{turbo} các ô nhớ. Các ô nhớ được đánh số địa chỉ từ 0 đến N_{turbo} -1, chẳng hạn, theo cột.



Hình 3.20. Thủ tục tính toán địa chỉ đầu ra bộ cài xen Turbo.

Bộ cài xen Turbo làm việc như sau: N_{turbo} bit thông tin đầu vào được viết vào ma trận các ô nhớ theo các địa chỉ bắt đầu từ 0, cuối cùng $N_{turbo} - 1$.

Toàn bộ các số liệu này được đọc ra theo một trình địa chỉ được xác định theo các thủ tục trình bày ở hình 3.20, gồm 9 bước:

1).Bước 1: Xác định thông số bộ đan xen Turbo n, trong đó n là số nguyên nhỏ nhất để $N_{Turbo} \le 2^{n+5}$. Bảng 14 cho chúng ta các thông số này;

2).Bước 2: Khởi đầu đưa (n+5) ô nhớ của thanh ghi đệm về trạng thái toàn 0. Rồi ghi (n+5) bit thông tin ở các địa chỉ từ 0 đến (n+4) vào thanh ghi đệm. Các bit thông tin được biểu diễn trong dạng: $(i_{n+4}i_{n+3}...i_2i_1i_0)$, trong đó các bit $(i_4...i_1i_0)$ được gọi là 5bit biểu diễn giá trị thấp nhất của chuỗi, ký hiệu LSB (Least Significant Bit), còn các bit $(i_{n+4}...i_6i_5)$ được gọi là nbit biểu diễn giá trị cao nhất của chuỗi, ký hiệu MSB (Most Significant Bit);

3).Bước 3: Lấy ra n bit MSB từ bộ đếm và cộng 1, để được giá trị mới. Sau đó lấy ra n bit LSB từ giá trị mới này để sử dụng cho bước 5, rồi xóa tất cả;

4).Bước 4: Tra cứu bảng 15 bằng cách từ 5bit LSB chuyển thành số hệ 10, để xác định chỉ số. Khi đã biết n (Bước 1) sẽ xác định được giá trị tương ứng là số giao của dòng chỉ số với cột n;

5).Bước 5: Nhân các giá trị nhận được ở bước 3 và bước 4 (hệ 10). Chuyển kết quả sang hệ 2, sau đó lấy ra nbit LSB;

Bảng 14. Thông số bộ cài xen Turbo

Kích cỡ N_{turbo}	Thông số bộ cài xen-n
378	4
570	5
762	5
1.146	6
1.530	6
2.298	7
3.066	7
4.602	8
6.138	8
9.210	9
10.282	9
20.730	10

6).Bước 6: Đảo vị trí 5bit LSB của bộ đếm thành 5bit MSB;

7).Bước 7: Tạo địa chỉ ra thử từ 5bit MSB (Bước 6) với nbit LSB (Bước 5);

8).Bước 8: Tiếp nhận địa chỉ ra thử này, nếu nó nhỏ hơn N_{turbo}, ngược lại thì xóa bỏ;

9) .Bước 9: Tăng địa chỉ bộ đếm lên 1, rồi lặp lại bước 3 đến bước 8, cho đến khi nhận được N_{turbo} địa chỉ ra cho bộ cài xen.

Thí dụ: Để minh chứng cho các bước trên, chúng ta xét một thí dụ cụ thể. Chọn N_{turbo} = 1530 = 34 × 45.

- B1: Từ bảng 14 xác định được thông số bộ cài xen n = 6;

- B2: Giả thử các bit thông tin ở (n+5) địa chỉ đầu tiên của bộ cài xen có dạng: (100011_00101). Như vậy, LSB = 00101; còn MSB = 100011;

- B3: MSB+1 = 100011_00001→ nbit LSB = $(100001)_2$ = $(33)_{10}$;

Bảng 15. Bảng tra cứu bộ cài xen Turbo

Các chỉ số	n=4	n=5	n=6	n=7	n=8	n=9	n=10
0	5	27	3	15	3	13	1
1	15	3	27	127	1	335	349
2	5	1	15	89	5	87	303
3	15	15	13	1	83	15	721
4	1	13	29	31	19	15	973
5	9	17	5	15	179	1	703
6	9	23	1	61	19	333	761
7	15	13	31	47	99	11	327
8	13	9	3	127	23	13	453
9	15	3	9	17	1	1	95
10	7	15	15	119	3	121	241
11	11	3	31	15	13	155	187
12	15	13	17	57	13	1	497
13	3	1	5	123	3	175	909
14	15	13	39	95	17	421	769
15	5	29	1	5	1	5	349
16	13	21	19	85	63	509	71
17	15	19	27	17	131	215	557
18	9	1	15	55	17	47	197
19	3	3	13	57	131	425	499
20	1	29	45	15	211	295	409
21	3	17	5	41	173	229	259
22	15	25	33	93	231	427	335
23	1	29	15	87	171	83	253
24	13	9	13	63	23	409	577
25	1	13	9	15	147	387	717
26	9	23	15	13	243	193	313
27	15	13	31	15	213	57	757
28	11	13	17	81	189	501	189
29	3	1	5	57	51	313	15
30	15	13	15	31	15	489	75
31	5	13	33	69	67	391	163

-B4: 5bit LSB của bước 2 = $(00101)_2 = (5)_{10}$. Từ bảng 15 ứng với chỉ số 5 và n = 6, xác định được giá trị 5;

-B5: 33×5 = 165 = $(10100101)_2$, → nbit LSB = (100101);

-B6: Đảo vị trí 5bit LSB của bước 2 thành 5bit MSB = 00101;

-B7: Tính thử địa chỉ (00101_100101) = $2^8+2^6+2^5+2^2+1 = (357) < N_{turbo} = 1530$. Chấp nhận làm địa chỉ đọc đầu tiên của bộ cài xen. Như vậy, các địa chỉ đọc ra của bộ cài xen phụ thuộc cả vào số liệu.

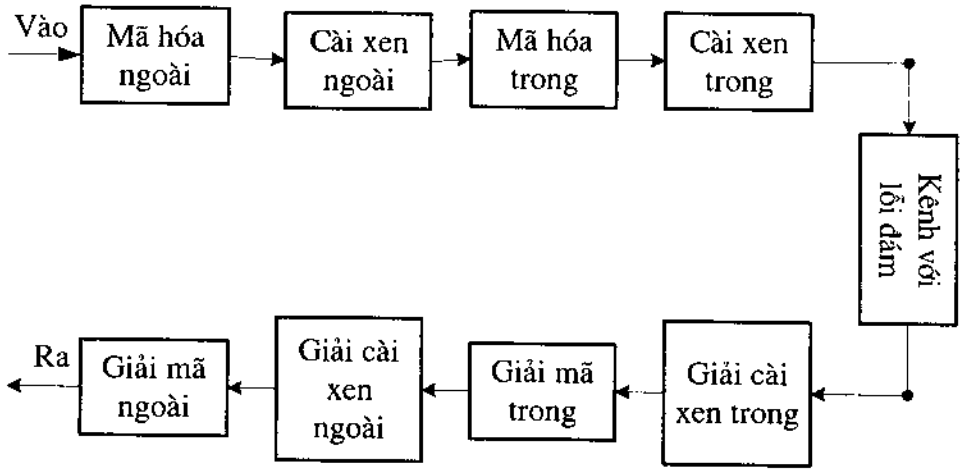
10. XÂU CHUỖI CÁC BỘ MÃ, CÙNG CÁC BỘ CÀI XEN

Trong thực tiễn, bắt gặp các hệ thống truyền tin, mà kênh truyền của chúng thường xuyên bị nhiễu rất mạnh. Với các hệ thống truyền tin như thế, khi truyền tin tức qua kênh, sẽ phát sinh lỗi đám dài, vượt khỏi khả năng phát hiện và sửa lỗi của một bộ mã riêng lẻ, mặc dù chúng đã được thiết kế đủ phức tạp. Cũng cần thấy rằng thời gian mã hóa/giải mã phụ thuộc vào độ phức tạp (số lỗi có thể sửa được) của bộ mã. Chính điều này dẫn đến hạn chế tốc độ truyền tin (chứ không phải khả năng thông qua của kênh). Vì vậy, người ta đưa ra ý tưởng xâu chuỗi các bộ mã (thường là hai), với năng lực thích hợp, cùng với chúng là xâu chuỗi các bộ cài xen (interliaver), để nâng cao năng lực chống nhiễu của hệ thống.

Trong hai bộ mã, theo trình tự từ ngoài vào trong kênh truyền, bộ mã thứ nhất gọi là bộ mã ngoài, cùng với nó là bộ cài xen ngoài, tiếp đến bộ mã trong và bộ cài xen trong. Sơ đồ biểu diễn chúng trên hình 3.21.

Giả thử bộ mã ngoài có tham số (n_1,k_1), bộ mã trong có tham số (n_2,k_2), thì bộ mã tương đương do chúng tạo nên có tham số: $n = (n_1\times n_2)$, $k = (k_1\times k_2)$. Như vậy, năng lực của bộ mã tương đương cũng tăng theo, nhưng độ dư thừa cũng tăng theo tỉ số: $\frac{k}{n} = \frac{k_1.k_2}{n_1.n_2}$. Trong thực tế thường chọn bộ mã ngoài là *mã khối*, bộ mã trong là *mã xoắn*.

Bộ cài xen là một ma trận (M cột×N hàng) các ô nhớ, để ghi các ký hiệu mã vào theo một trình tự nào đó, còn khi đọc ra theo một trình tự khác, với chức năng để phân tán lỗi đám tốt nhất (theo quan điểm sửa lỗi) khi giải cài xen. Có một vài phương án ghi/đọc các ký hiệu mã ở bộ cài xen. Phương án đơn giản nhất là cài xen theo chu kỳ.



Hình 3.21. Xâu chuỗi các bộ mã, cùng các bộ cài xen.

a). Cài xen khối theo chu kỳ

Trong trường hợp đơn giản này, khi *cài xen*, người ta *ghi* các ký hiệu mã vào khối, chẳng hạn, theo *cột*, còn khi *đọc* ra theo *hàng*; khi *giải cài xen*, thực hiện theo trình tự ngược lại, tức là *ghi* các ký hiệu mã vào khối *theo hàng*, còn khi *đọc* ra theo *cột*. (Hình 3.22). Trong trường hợp như thế, bộ cài xen (gồm cài xen và giải cài xen) có tác dụng phân tán lỗi đám với chiều dài nhỏ hơn, hoặc bằng M, thành lỗi đơn, như trường hợp a) trong hình 3.22. Khi gặp các lỗi đám dài hơn M ký hiệu, bộ cài xen sẽ phân tán thành các lỗi đám có chiều dài nhỏ, đủ nằm trong khả năng sửa của bộ mã.

Tuy nhiên, trong trường hợp xấu nhất, bộ cài xen có thể vô tình chuyển lỗi đơn (các lỗi đơn xuất hiện có chu kỳ, trong cùng một cột) thành lỗi đám, như trường hợp b) trong hình 3.22.

b). Cài xen giả ngẫu nhiên

Phương án cài xen tốt nhất là phương án để các bộ mã có thể phát hiện và sửa được *mọi dạng lỗi* xuất hiện trên kênh truyền, khi đã xác định năng lực của bộ mã và dạng điều chế. Phương án cài xen giả ngẫu nhiên được đề cập đến. Người ta thường xác định qui luật ghi/đọc trên cơ sở mô phỏng, với tiêu chí cải thiện tốt nhất tỉ số S/N ở đầu ra bộ giải mã. Thực tiễn, xác định được một vài qui luật như thế. Chúng ta dẫn ra đây quy luật của bộ cài xen CDMA - 2000.

Các ký hiệu đầu vào được ghi vào bộ cài xen lần lượt theo địa chỉ từ 0 đến N - 1, chẳng hạn, theo cột. Khi đọc các ký hiệu ra theo địa chỉ A_i, được xác định theo thuật toán sau:

$$A_i = 2^m (i \bmod J) + BRO_m([i/J])$$

trong đó: i = 0 đến N-1;

tham số m, J được xác theo bảng 16;

[x] - phép tính giá nguyên của x;

$BRO_m([y])$ - biểu thị giá trị m bit đảo của y (xem thí dụ).

Chuỗi đã mã hóa (1, 2, 3,..., 34, 35, 36)

Đọc →

Ghi ↓

1	5	9	13	17	21	25	29	33
2	6	10	14	18	22	26	30	34
3	7	11	15	19	23	27	31	35
4	8	12	16	20	24	28	32	36

Chuỗi đã cài xen (1, 5, 9,...,28, 32, 36)

Kênh với lỗi đám

a).Chuỗi thu (1, 5,...,6*, 10*,14*,...,32, 36)
(lỗi đám)

b).Chuỗi thu (1*,5,...,2*,...,3*,...,4*,...,32,36)
(lỗi đơn)

Ghi →

Đọc ↓

1*	5	9	13	17	21	25	29	33
2*	6*	10*	14*	18	22	26	30	34
3*	7	11	15	19	23	27	31	35
4*	8	12	16	20	24	28	32	36

a). Chuỗi giải cài xen (1, 2, 3,...,5, 6*,..., 9, 10*,..., 13, 14*,...,35, 36)
(lỗi đơn)

b). Chuỗi giải cài xen (1*, 2*,3*,4*, 5,...,35, 36)
(lỗi đám)

Hình 3.22. Cài xen và giải cài xen chu kỳ, với ma trận (9 × 4) ô nhớ.

Bảng 16. Các tham số bộ cài xen (trích)

Kích thước N của bộ cài xen	m	J
48	4	3
96	5	3
192	6	3

Bảng 16 tiếp theo

Kích thước N của bộ cài xen	m	J
384	6	6
768	6	12
1.536	6	24
3.072	6	48
6.144	7	48
12.288	7	96
72	3	9
144	4	9

Thí dụ: xét bộ cài xen N = 96 (m = 5, J = 3), chúng ta thử tính một vài địa chỉ đọc đầu tiên:

- i = 0 → i mod J = 0, [i/J] = 0, → $A_i = 0$ (quá trình đọc chỉ giữ chậm 1 ký hiệu);

- i = 1 → i mod J = 1 mod 3 = 1, [i/J] = [1/3] = 0, → $A_i = 2^5 = 32$ (quá trình đọc bị giữ chậm 33 ký hiệu);

- i = 2 → i mod J = 2mod3 = 2, [i/J] = [2/3] = 0, → $A_i = 2^5 \times 2 = 32\times2 = 64$ (quá trình đọc bị giữ chậm 65 ký hiệu);

- i = 3 → i mod J = 3 mod 3 = 1, [i/J] = [3/3] = 1, → tính $BRO_5([1])$ như sau: $(1)_{10} = (00001)_2$, đảo của 5 bit này là $(10000)_2 = (16)_{10}$, suy ra $BRO_5([1]) = 16 \rightarrow A_i = 2^5 \times 1 + 16 = 80$ (quá trình đọc bị giữ chậm 81 ký hiệu) ; v.v...Độ giữ chậm toàn khối được xác định bằng max trong các giá trị trên.

Như vậy, các bộ cài xen gây *giữ chậm* các ký hiệu mã, do quá trình ghi/đọc.

CHƯƠNG 4

MỘT SỐ THUẬT TOÁN MẬT MÃ

1. VẤN ĐỀ AN TOÀN THÔNG TIN TRONG QUÁ TRÌNH TRUYỀN

◆ Từ cổ xưa, khi con người có nhu cầu truyền tin cho nhau, thì những *thông tin quan trọng và nhạy cảm* đã cần được giữ bí mật. Như vậy, bài toán *giữ bí mật thông tin* đã xuất hiện rất sớm trong lịch sử của nhân loại. Ngày nay, bài toán này càng thêm bức xúc, phức tạp. Hơn nữa, do sự xuất hiện của nhiều hình thức *giao dịch điện tử* như chính phủ điện tử, thương mại điện tử, ..., từ đó xuất hiện những bài toán mới - vấn đề *xác thực*, bảo đảm *toàn vẹn* nội dung tin tức.

◆ Bài toán giữ bí mật thông tin khi truyền, có thể được giải quyết bằng ba giải pháp sau:

- Sử dụng kênh truyền riêng, hoàn toàn bí mật;
- Sử dụng cách viết che giấu (ẩn giấu);
- Sử dụng cách *biến đổi tin tức* từ dạng đọc (hoặc nghe, hoặc nhìn) *hiểu được* thành ra dạng đọc (hoặc nghe, hoặc nhìn) *không hiểu được*.

Nhận xét về ba giải pháp:

- *Giải pháp 1*: không khả thi, vì xu hướng hội nhập của các mạng thông tin ngày nay, chỉ có như vậy mới có thể chia sẻ thông tin chung.

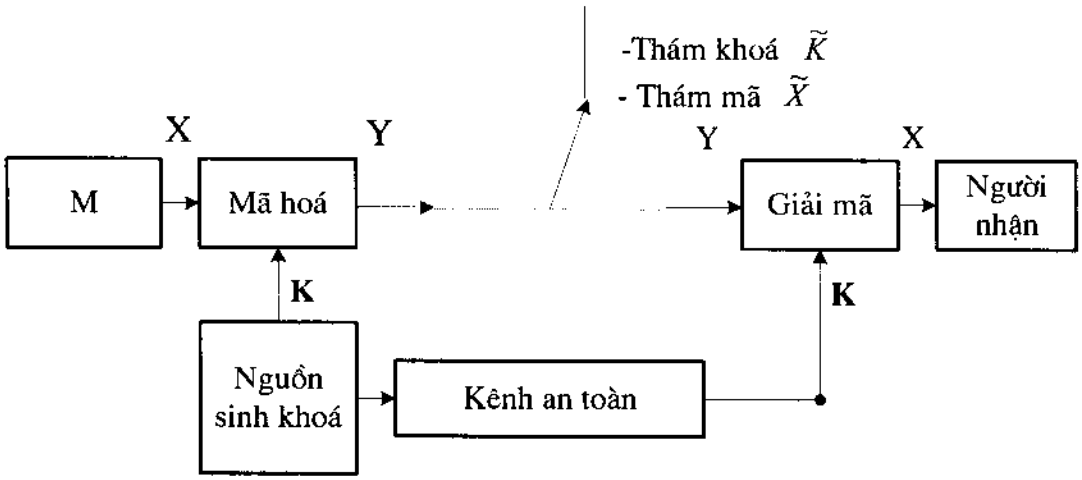
- *Giải pháp 2*: Viết che giấu tin tức cần giữ bí mật vào tin tức thông thường - *Steganography*. Chẳng hạn:

  - Tạo ra bản tin thông thường, mà mỗi ký tự đầu của các từ hình thành nên bản tin cần che giấu;
  - Viết bản tin thông thường, rồi dùng mực hoá học chấm vào các ký tự để hình thành nên bản tin cần che giấu;
  - Một bức ảnh đã số hoá, có những bit có thể thay đổi giá trị mà nội dung bức ảnh không thay đổi. Các bit này dùng để tải tin tức cần che giấu. Ngày nay, tồn tại các *thuật toán nhúng* (Digital Watermarking) từ kỹ thuật này.

- *Giải pháp 3*: Thực chất là sử dụng *mật mã học* để bảo mật văn bản, tín hiệu thoại, tín hiệu ảnh.

An toàn thông tin trên mạng, khi sử dụng mật mã học, có thể bảo đảm được ba chức năng sau: *bảo mật, xác thực và toàn vẹn* nội dung tin tức trong suốt quá trình truyền (cho rằng *mã kênh* có hiệu quả tuyệt đối). Trong chương này chúng ta sẽ nghiên cứu một số thuật toán mật mã, có thể bảo đảm được ba chức năng trên.

Trước hết chức năng *bảo mật tin tức*, thông thường được thực hiện bằng sơ đồ *mật mã truyền thống*, với các thuật toán mật mã hiện đại. Sơ đồ khối đơn giản của mật mã truyền thống được trình bày trên hình 4.1a).



a) Sơ đồ mật mã (mã hoá/giải mã) với một khoá **K**

Hình 4.1a). Sơ đồ khối mật mã truyền thống - mật mã với một khoá mật K.

Trên hình 4.1a) chỉ ra sơ đồ quá trình của mật mã truyền thống. Các tin tức rõ nghĩa khởi đầu thường được gọi là *văn bản rõ* (*plain text* - văn bản công khai, văn bản gốc), được biến đổi thành dạng *không rõ nghĩa* của chuỗi các kí tự (hoặc các bit), được gọi là *văn bản mã (cipher text).* Quá trình mã hoá, bao gồm việc sử dụng thuật toán và khoá K nào đó. Khoá - đó là một giá trị, được gọi là *khoá mật*, không phụ thuộc vào văn bản rõ. Kết quả mã hóa, khi thực hiện thuật toán, phụ thuộc vào việc sử dụng khoá K này. Sự thay đổi khoá dẫn đến việc thay đổi kết quả nhận được của thuật toán.

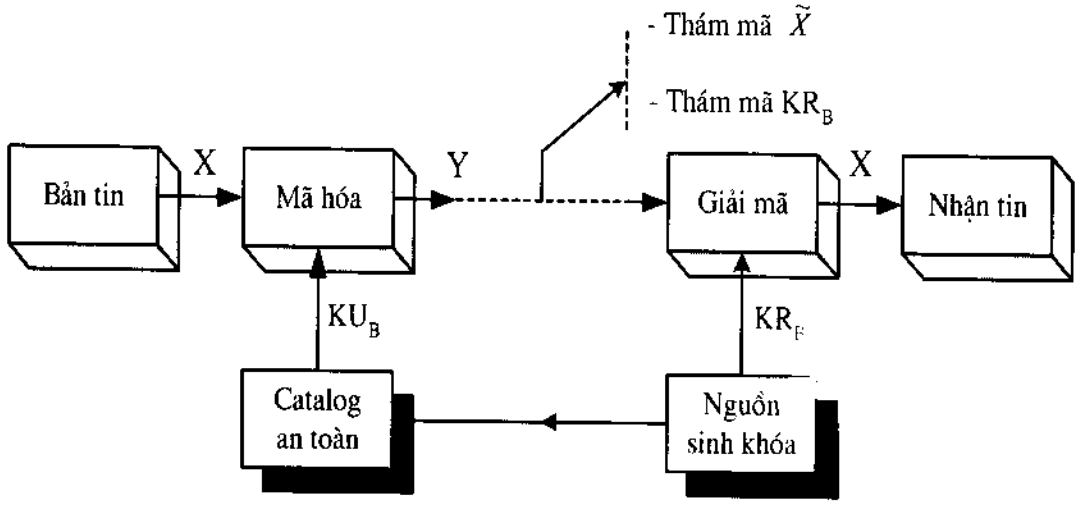
Văn bản mã nhận được có thể gửi cho người nhận. Sau khi thu nhận được văn bản mã, người nhận có thể biến đổi ngược lại thành dạng văn bản rõ, nhờ sử dụng thuật toán phù hợp và chính khoá K đã được sử dụng khi mã hoá.

Độ tin cậy của mật mã truyền thống phụ thuộc vào một vài yếu tố. Trước hết, thuật toán mật mã *cần phải phức tạp*, để không có khả năng giải mã, khi chỉ có văn bản mã. Thứ hai, yếu tố cơ bản độ tin cậy của mật mã truyền thống là *khoá mật*, trong khi đó chính thuật toán mật mã không cần bí mật. Do đó, giả thiết rằng cần phải bảo đảm không có khả năng thực tiễn giải mã tin tức trên cơ sở biết văn bản mã, *thậm* chí biết thuật toán mã hoá/giải mã. Nói một cách khác, *không* yêu cầu giữ bí mật thuật toán, nhưng *cần* bảo đảm giữ bí mật khoá K.

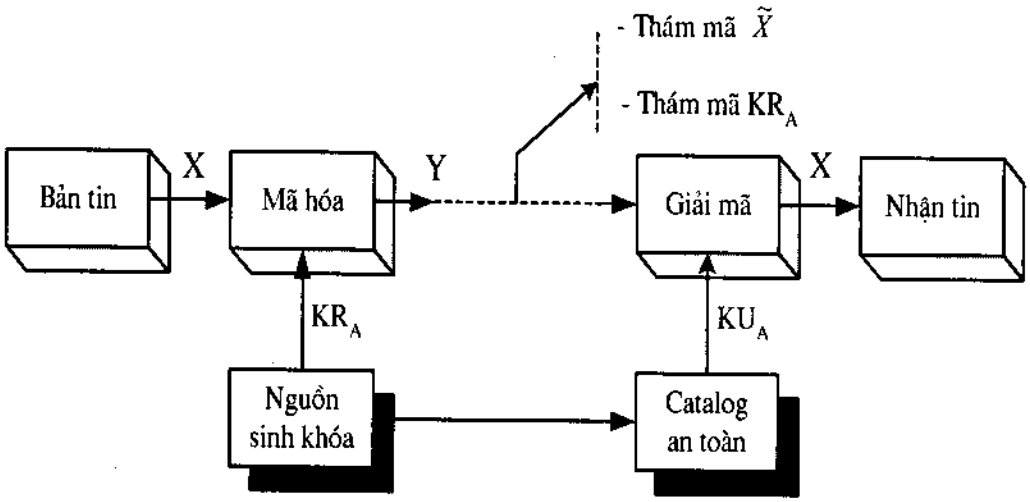
Mật mã truyền thống với phương pháp hiện đại, được thực hiện bằng hàng loạt thuật toán phức tạp, tin cậy như: DES, IDEA, CAST-128, RC2, v.v.. Trong chương này, chúng ta chỉ đưa ra thuật toán AES - Advanced Encryption Standard làm thí dụ.

Chức năng *xác thực và toàn vẹn* nội dung tin tức được thực hiện cùng một lúc nhờ sử dụng *chữ ký số*. Sơ đồ chữ ký số phổ thông nhất ngày nay được thực hiện nhờ *thuật toán RSA* - Rivest

Shamir Adleman, hoặc *chuẩn chữ ký số DSS* - Digital Signature Standard, hoặc chữ ký số xây dựng trên đường cong elliptic.



a). Chức năng bảo mật của mật mã với khóa công khai



b). Chức năng chữ ký của mật mã với khóa công khai

Hình 4.1b). Hai chức năng chính của mật mã với khóa công khai.

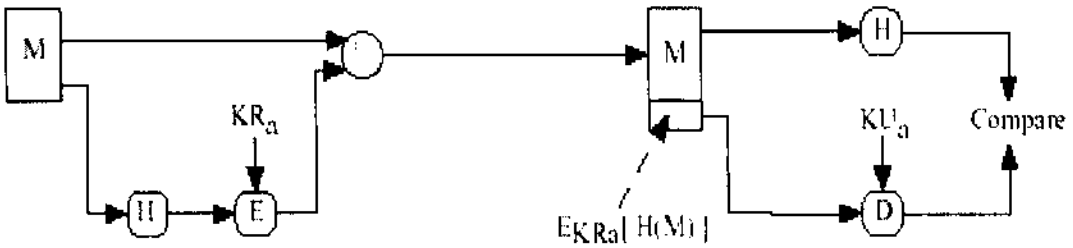
Thuật toán mật mã với khóa công khai cung cấp cho mỗi người trong cuộc một cặp khóa. Một khóa gọi là *khóa riêng*, ký hiệu là KR - phải giữ bí mật; một khóa gọi là *khóa công khai*, ký hiệu là KU - để cấp cho đối tác. Khi đối tác mã hóa bằng khóa công khai KU, thì chức năng của sơ đồ là *bảo mật*.

Vì để giải mã chỉ người có khóa riêng KR tương ứng mới thực hiện được. Còn khi mã hóa bằng khóa riêng KR, thì chức năng của sơ đồ là *xác thực* - hoặc *chữ ký*. Vì chỉ người có khóa

riêng mới làm được điều đó. Hai chức năng chính của mật mã với khóa công khai được trình bày trên hình 4.1b).

Tuy nhiên, mã hóa/giải mã bằng khóa công khai có tốc độ rất chậm (chậm hơn *hàng nghìn lần* so với mật mã truyền thống). Vì thế, để ký xác nhận một tin tức có độ dài bất kỳ, trước hết cần phải *tóm lược* bản tin đó. Chức năng này được thực hiện nhờ *hàm băm có tính một chiều*. Sơ đồ hàm băm có tính một chiều được thực hiện bằng hàng loạt thuật toán, chẳng hạn, MD5, SHA-1, RIPEMD-160, v.v..

Sơ đồ đơn giản thực hiện hàm băm và ký xác nhận được trình bày trên hình 4.2.



Hình 4.2. Ký xác nhận tin tức và kiểm tra chữ ký theo thuật toán RSA.

Chú thích : M - Bản tin, KU_a - Khoá công khai của A,

H - Hàm băm, KR_a - Khoá riêng của A,

E/D - Thuật toán mã hoá/giải mã, | | - Phép ghép hai bản tin.

H(M) - Tóm lược bản tin M,

$E_{KRa}[H(M)]$ - Mã hoá H(M) bằng khoá riêng (chức năng ký, xác nhận).

2. CHUẨN MẬT MÃ DỮ LIỆU TIÊN TIẾN (AES)

2.1. Sự phát triển

AES - Advanced Encryption Standard (tên ban đầu Rijndael) là thuật toán mật mã khối được công nhận làm chuẩn của nước Mỹ (NIST – Viện công nghệ và tiêu chuẩn quốc gia Mỹ công bố theo US FIBS PUB 197 vào tháng 11/2001 sau quá trình từ 5 tuyển chọn). AES là thuật toán mật mã được lựa chọn để thay thế thuật toán DES - Data Encryption Standard (chuẩn mật mã dữ liệu *cũ*).

Rijndael là thuật toán được phát triển từ thuật toán mật mã khối Square của 2 nhà mật mã học người Bỉ *Daemen* và *Rijmen* (Square được công bố vào năm 1977. Cấu trúc của chúng dựa trên mạng mạng hoán vị - thay thế (SPN - Substitution Permutation Network) có 8 vòng, làm việc với 128 bit dữ liệu và 128 bit khóa). Thuật toán Square được phát triển từ thuật toán Shark (Shark là thuật toán mật mã khối, làm việc với 64 bit dữ liệu và 128 bit khóa. Cấu trúc của nó gồm 6 vòng SPN).

Không giống như DES, thuật toán AES có cấu trúc SPN (không phải là mạng Feistel). AES hoạt động rất nhanh trên cả phần cứng và mềm, nó rất dễ ràng thực hiện, và yêu cầu ít bộ nhớ.

2.2. Mô tả sơ bộ của thuật toán

Mô tả chung:

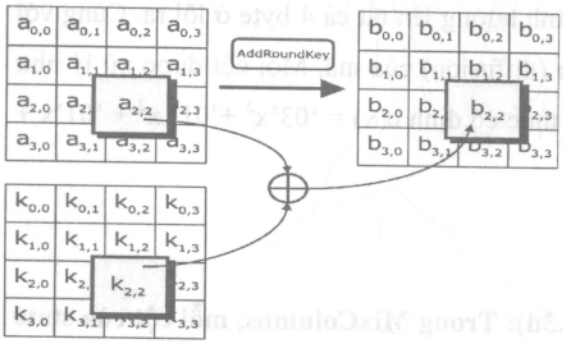
- Nhà thiết kế: Vincent Rijmen và Joan Daemen.
- Năm công bố đầu tiên: 1988.
- Kích thước khối dữ liệu: *128 bit*.
- Kích thước khóa: *128, 192*, hoặc *256 bit*.
- Cấu trúc thiết kế: mạng hoán vị - thay thế (SPN – Substitution-Permutation Network).
- Số vòng: 10, 12, hoặc 14 (tương ứng với các kích thước của khóa).

Nói một cách chính xác AES không hoàn toàn là Rijndael nguyên bản (dù rằng trong thực tế chúng được sử dụng như nhau), bởi vì Rijndael có thể thực hiện trên các khối dữ liệu và khóa có kích thước khác nhau (là bội số của 32 bit, nhỏ nhất là 128 bit và lớn nhất là 256 bit), còn AES chỉ làm việc trên khối dữ liệu kích thước 128 bit và các khóa 128, 192 hoặc 256 bit. Khối dữ liệu 128 bit được phân thành 16 mảng, mỗi mảng là 1byte (128 = 16×8)

Hầu hết các tính toán của AES được thực hiện trên trường hữu hạn. AES hoạt động trên các mảng của byte kích thước 4×4, được gọi là *state* (trạng thái).

Đối với quá trình mã hóa, mỗi vòng của thuật toán (ngoại trừ vòng cuối cùng) bao gồm 4 giai đoạn - stage (hoặc bước).

2.2a). AddRoundKey – mỗi byte của state được kết hợp với *khóa vòng*, mỗi khóa vòng được sinh ra từ *khóa bí mật*, khi sử dụng thời gian biểu của khóa (key schedule), mỗi khóa vòng có cùng kích thước với state.

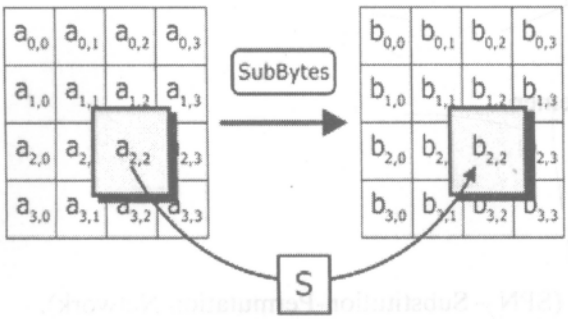


Hình. 4.3a). Trong bước AddRoundKey, mỗi byte của state được kết hợp tương ứng với 1 byte của khóa vòng bằng sử dụng toán tử XOR.

Vòng cuối cùng, thay thế bước Mixcolumns bằng một giá trị AddRoundKey riêng biệt.

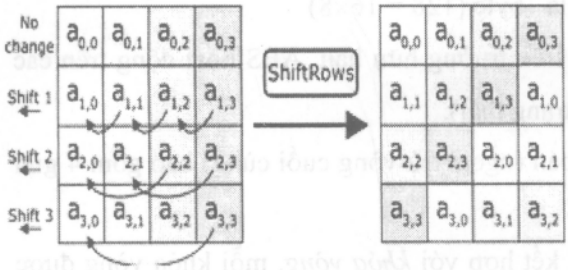
2.2b). SubstituteBytes – bước thay thế có tính phi tuyến, theo đó mỗi byte được thay thế bởi một byte có giá trị khác theo bảng tìm kiếm (lookup table) – gọi là hộp S. Như vậy, mỗi byte trong bảng được thay thế bởi một byte (phần tử 8bit) của hộp S (S-box). Hộp S được sinh ra từ hàm ngược (inverse function) trên GF(2^8) để có được tính chất phi tuyến cao nhất. Để ngăn ngừa cuộc tấn công dựa trên các tính chất đại số đơn

giản, hộp S được thiết kế là sự biến đổi của hàm ngược nhờ một ma trận được chọn với các hàng là độc lập tuyến tính. Hộp S cũng được thiết kế để bảo đảm tính xáo trộn tốt nhất.



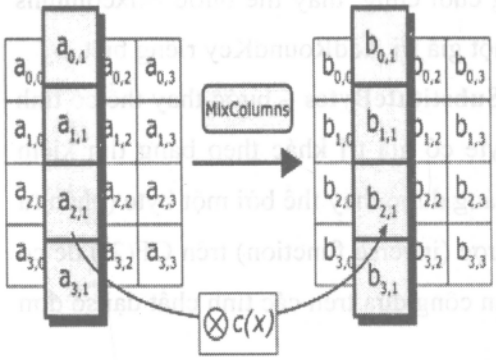
Hình 4.3b): Trong bước SubstituteBytes, mỗi byte trong *state* được thay thế bởi một byte (phần tử 8bit) của hộp S đóng vai trò là bảng tìm kiếm.

2.2c). ShiftRows – bước chuyển dịch, theo đó mỗi hàng của state được dịch vòng một số bước nào đó. Đối với AES hàng đầu tiên không thay đổi. Các byte của hàng thứ 2 được dịch vòng đi 1byte về bên trái. Tượng tự, các byte của hàng thứ 3 và 4 được dịch vòng đi 2 và 3 byte tương ứng về phía trái. Bằng cách này *mỗi cột* của state ra của ShiftRows là sự kết hợp lại của các byte từ mỗi cột của state vào.



Hình.4.3c): Trong bước ShiftRows, các byte trong mỗi hàng của state được dịch vòng về phía trái. Số lượng các bước dịch vòng là khác nhau ở mỗi hàng.

2.2d). MixColumns – hoạt động trên các cột của state, thực hiện một biến đổi tuyến tính. Trong bước này 4 byte của mỗi cột của state được tổ hợp sử dụng một biến đổi tuyến tính ngược. Trong hàm MixColumns mỗi byte của lối vào sẽ ảnh hưởng lên tất cả 4 byte ở lối ra. Cùng với ShiftRows, MixColumns sẽ tạo ra *tính khuếch tán* (diffusion) của mã. Mỗi cột được xử lý như một đa thức trên GF(2^8) và được nhân với một đa thức cố định c(x) = '03'x^3 +'01' x^2 + '01'x + '02' theo modul (x^4 + 1).



Hình 4.3d): Trong MixColumns, mỗi cột của state được nhân với một đa thức cố định c(x) theo mod(x^4 + 1).

Hàm Mixcolums(*state*) được thực hiện trên mỗi cột của ma trận state. Như vậy, đối với 4 cột của ma trận state, hàm Mixcolums(*state*) được thực hiện 4 lần.

Xem xét việc thực hiện hàm Mixcolums(*state*) trên 1 cột của ma trận. Kết quả của phép biến đổi cũng sinh ra cột của ma trận tương ứng.

Với véctơ (xét trường hợp chung, không cần đánh số của cột): $\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix}$

Cột này có thể được biểu diễn theo đa thức bậc 3 như sau:

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \tag{4.1}$$

Nhận thấy rằng, bởi vì *hệ số* của $a(x)$ – là các *byte*, nghĩa là các phần tử của trường GF(2^8), hàm $a(x)$ là đa thức trên trường GF(2^8) (nhưng không phải là phần tử của trường AES).

Đa thức $a(x)$ được rút gọn, nhờ việc nhân đa thức này với 1 đa thức cố định bậc 3 - $c(x)$ (tính theo mod $(x^4 + 1)$):

$$c(x) \,.\, a(x) \bmod (x^4 + 1) \tag{4.2}$$

ở đây $c(x) = c_3x^3 + c_2x^2 + c_1x + c_0 =$ '03'x^3 + '01'x^2 + '01'x + '02'.

Chú ý rằng, các phép toán trong (4.2) không là các phép toán trong trường AES (bởi vì các đa thức $a(x)$ và $c(x)$ không thuộc trường AES). Ngoài ra, đa thức $x^4 + 1$ là đa thức không tối giản trên trường GF(2) ($x^4 + 1 = (x + 1)^4$). Nguyên nhân duy nhất để (4.2) cần phải được thực hiện theo mod của đa thức bậc 4 là kết quả của nó cần phải là đa thức bậc 3.

Trong trường GF(2) thực hiện phương trình sau: $x^i(\text{mod } x^4 + 1) = x^{i \bmod 4}$

Trong (4.2), các hệ số của x^i (i = 0 ,1, 2, 3) là tổng của các tích c_ja_k với các chỉ số tương ứng, thỏa mãn điều kiện $j + k = i$ (mod 4), ở đây j, k = 0 ,1, 2, 3. Ví dụ hệ số của x^2 trong (4.2) có dạng như sau:

$$c_2a_0 + c_1a_1 + c_0a_2 + c_3a_3.$$

Các phép nhân và cộng đều được tính trong trường GF(2^8). Khi đó (4.2) có thể thực hiện đầy đủ như sau:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} c_0 & c_3 & c_2 & c_1 \\ c_1 & c_0 & c_3 & c_2 \\ c_2 & c_1 & c_0 & c_3 \\ c_3 & c_2 & c_1 & c_0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \tag{4.3}$$

Chú ý rằng đa thức $c(x)$ là đa thức nguyên tố tương hỗ với $x^4 + 1$ trên trường GF(2). Trong trường GF(2) tồn tại nghịch đảo $c(x)^{-1}(\text{mod } x^4 + 1)$. Nói một cách khác, ma trận và biến đổi (4.3) là thuận nghịch.

2.2e). Hộp S trong AES

Bảng tìm kiếm để thay thế (cũng gọi là hộp S) có tính đảo ngược (invertible) và được xây dựng dựa trên 2 biến đổi sau:

- Thực hiện phép biến đổi ngược trong trường GF(2^8), với '00' được ánh xạ lên chính bản thân nó.

- Thực hiện biến đổi affine (trên GF(2)), được mô tả như sau:

$$y = Ax^{-1} + b, \tag{4.4}$$

Ở đây x^{-1} (với giá trị $[x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ – là véc tơ đảo của x) $\in$ GF(2^8). Với các byte x bất kỳ khác không, chúng sẽ được thay thế với biến đổi (4.4). Còn khi x là byte *0* thì kết quả của phép biến đổi được $y = b$. (b có giá trị '63')

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Nhận thấy rằng phép biến đổi phi tuyến (4.4) được đảm bảo duy nhất bởi phép đảo phi tuyến x^{-1}. Nếu biến đổi (4.4) được thực hiện trực tiếp trên x, thì phương trình affine (4.4) sẽ tuyệt đối tuyến tính.

Bởi vì ma trận A có kích thước 8×8 là thuận nghịch. (Hơn nữa, các hàng trong A được chọn sao cho là độc lập tuyến tính trong trường GF(2^8)). Vì thế biến đổi (4.4) cũng có tính thuận nghịch. Và SubBytes (*state*) có tính thuận nghịch.

Bảng 4.1. Bảng thay thế (hộp S) được trình bày (trong dạng hexa).

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2

80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Trong bảng, các *cột* được xác định qua *4bit* thuộc *nửa giá trị thấp*, còn các *hàng* được xác định theo *4bit* thuộc *nửa giá trị cao* của byte. Ví dụ byte có giá trị '9a' – hệ 2 là (0101_1001) sẽ được biến đổi thành byte có giá trị 'b8' – hệ 2 là (0001_1101) qua bảng thay thế (hộp S) của AES. Giá trị ngược của '00' là chính nó. (Trong (4.4) sau khi thực hiện phép nhân ma trận, kết quả sẽ được XOR với 99 (hệ 10) hoặc '63' (hệ 16) là giá trị của *b:* '63' = "1100_0110" = 99).

Bảng 4.2. Bảng thay thế (hộp S) đảo.

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
10	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
20	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
30	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
40	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
50	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
60	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
70	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
80	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
90	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a0	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b0	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c0	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d0	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e0	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f0	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Bảng S đảo (sử dụng khi giải mã) được sinh ra từ bảng S bằng cách lấy nghịch đảo.Ví dụ, từ bảng S đảo, giá trị 'db' được đảo thành '9f' (cột 'b', hàng 'd').

2.2f). Danh mục khóa

Các *khóa vòng* được định nghĩa từ *khóa mật* theo bảng danh mục khóa. Bảng này bao gồm hai phần: bảng khóa mở rộng và bảng lựa chọn khóa vòng. Nguyên tắc cơ bản phù hợp theo:

- Toàn bộ số bit của khóa vòng bằng độ dài của khối nhân với số vòng cộng 1 (chẳng hạn, khối dài 128bit và có 10 vòng, thì số bit của khóa vòng là: 128×(10+1) =1408);
- Khóa mật được mở rộng thành *khóa mở rộng;*
- Khóa vòng được lấy từ khóa mở rộng này theo cách sau: vòng thứ nhất bao gồm Nb từ đầu tiên, vòng thứ hai là Nb từ tiếp theo và cứ thế.

♣). Khóa mở rộng

Khóa mở rộng là một mảng tuyến tính của các từ 4byte và được định nghĩa bởi *W[Nb*(Nr + 1)]*. Nk từ đầu tiên bao gồm chỉ các bit của khóa mật. Toàn bộ từ còn lại được định nghĩa theo thuật ngữ từ với chỉ số nhỏ. Hàm khóa mở rộng phụ thuộc vào số Nk (kích thước khối khóa - Nk = 4).

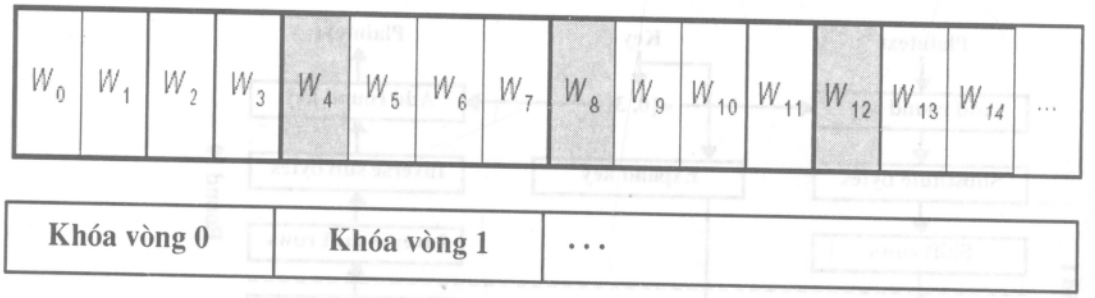
Với $Nk \leq 6$, chúng ta có:

```
KeyExpansion(byte Key[4*Nk] word W[Nb*(Nr+1)])
{
      for(i = 0; i < Nk; i++)
            W[i] = (key[4*i],key[4*i+1],key[4*i+2],key[4*i+3]);

      for(i = Nk; i < Nb * (Nr + 1); i++)
      {
            temp = W[i - 1];
            if (i % Nk == 0)
                  temp = SubByte(RotByte(temp)) ^ Rcon[i / Nk];
            else if (i % Nk == 4)
                  temp = SubByte(temp);
            W[i] = W[i - Nk] ^ temp;
      }
}
```

♣). Lựa chọn khóa vòng

Khóa vòng thứ i được cho bởi các từ đệm khoá vòng W[Nb*i] đến W[Nb*(i+1)], (Nb - kích thước khối dữ liệu Nb = 4), được biểu diễn trên hình 4.4. Mỗi từ đệm khóa vòng có kích thức 4 byte.



Hình 4.4. Mở rộng khóa và lựa chọn khóa vòng đối với Nb = 4 và Nk = 4.

2.3. Sơ đồ khối AES

Quá trình mã hóa (cũng như giải mã), gồm 10 vòng (xem hình 4.5.).

Mỗi vòng mã hóa gồm 4 phép biến đổi: Subbytes → ShiftRows → MixColumns → AddRoundkey. Riêng ở vòng 10 không có phép MixColumns. Ở đầu vào mã hóa (cũng như giải mã) được bắt đầu bằng phép AddRoundkey.

Các toán tử giải mã là các phép biến đổi ngược (Inverse) của các toán tử mã hóa.

2.4. Cơ sở toán học trong AES

Nhiều phép toán trong AES được định nghĩa ở *mức byte*, với các byte được biểu diễn qua các phần tử trong trường hữu hạn GF(2^8). Một số phép toán khác được định nghĩa bằng thuật ngữ của *từ mã 4 byte*.

a).Trường GF(2^8)

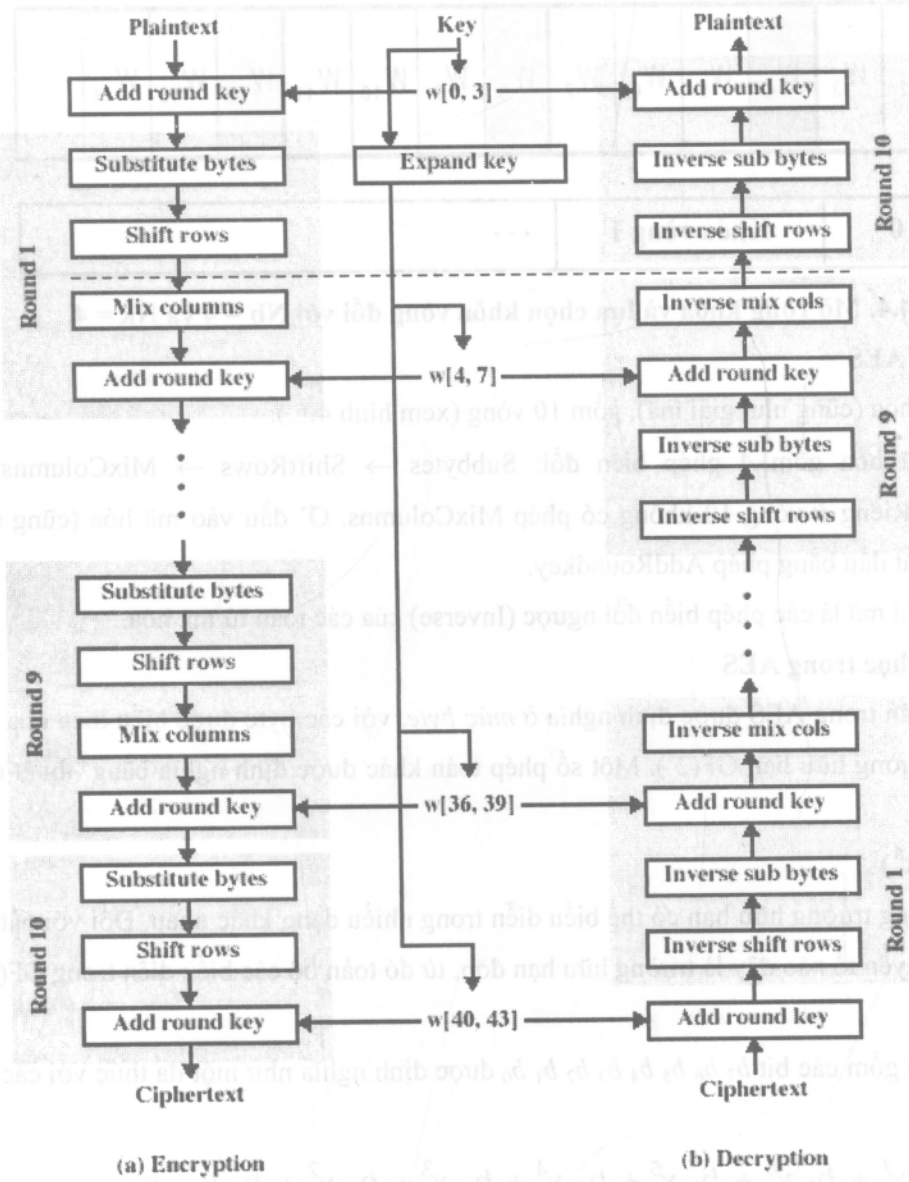
Các phần tử trong trường hữu hạn có thể biểu diễn trong nhiều dạng khác nhau. Đối với bất kỳ một luỹ thừa nguyên tố nào đây là trường hữu hạn đơn, từ đó toàn bộ các biểu diễn trong GF(2^8) là cùng cấu trúc.

Một byte b bao gồm các bit $b_7\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$ được định nghĩa như một đa thức với các hệ số trong {0, 1}:

$$b_7 x^7 + b_6 x^6 + b_5 x^5 + b_4 x^4 + b_3 x^3 + b_2 x^2 + b_1 x + b_0$$

Ví dụ byte với giá trị trong hệ 16 (hexadecimal) là '57' (hệ 2 là 01010111- viết theo trình tự giảm dần) tương ứng với đa thức:

$$x^6 + x^4 + x^2 + x + 1$$



Hình 4.5. Sơ đồ khối AES (bên trái – mã hóa, bên phải – giải mã).

b). Phép cộng

Trong biểu diễn đa thức, tổng của hai phần tử là một đa thức với các hệ số được cho bởi tổng theo mod2 của các hệ số của hai thành phần.

Ví dụ: '57' + '83' = 'D4', hoặc với biểu diễn đa thức:

$$(x^6 + x^4 + x^2 + x + 1) + (x^7 + x + 1) = x^7 + x^6 + x^4 + x^2 .$$

Trong dạng nhị phân chúng ta có: "01010111" + "10000011" = "11010100". Rõ ràng phép cộng phù hợp với phép XOR ở mức byte. Phép cộng là đóng, giao hoán và kết hợp.

c). Phép nhân

Trong biểu diễn đa thức, phép nhân trong GF(2^8) thực hiện theo modulo với một đa thức nhị phân tối giản bậc 8 (đa thức tối giản là đa thức không có ước số ngoài 1 và chính bản thân nó). Trong AES đa thức đó được gọi là *m(x)* và có dạng:

$$m(x) = x^8 + x^4 + x^3 + x + 1$$

hoặc '11B' trong hệ 16, còn trong hệ 2 là 100011011

Ví dụ:

'57' • '83' = 'C1',

$$(x^6 + x^4 + x^2 + x + 1)(x^7 + x + 1) = x^{13} + x^{11} + x^9 + x^8 + x^7 + x^7 + x^5 + x^3 + x^2 + x + x^6 + x^4 + x^2 + x + 1 = x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$$

$$x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \text{ modulo } x^8 + x^4 + x^3 + x + 1 = x^7 + x^6 + 1$$

Rõ ràng, kết quả là đa thức nhị phân có bậc nhỏ hơn 8. Không giống với phép cộng, đây là phép toán không đơn giản ở mức byte.

Phép nhân được định nghĩa ở trên có tính kết hợp và đó là phần tử trung hòa ('01'). Đối với đa thức nhị phân bất kỳ b(x) có bậc nhỏ hơn 8, thuật toán Euclid mở rộng có thể được sử dụng để tính đa thức a(x), c(x) như sau:

$$b(x)a(x) + m(x)c(x) = 1$$

Từ đây a(x)•b(x) mod m(x) = 1, hoặc

$$b^{-1}(x) = a(x) \bmod m(x)$$

Hơn nữa, dẫn đến a(x)•(b(x) + c(x)) = a(x)•b(x) + a(x)•c(x)

Phù hợp theo điều đó, tập hợp các giá trị có thể gồm 256 byte, với phép cộng là toán tử XOR, còn với phép nhân được định nghĩa ở trên có cấu trúc của trường hữu hạn GF(2^8).

d). Phép nhân với x

Nếu nhân b(x) với đa thức x, chúng ta có:

$$b_7x^8 + b_6x^7 + b_5x^6 + b_4x^5 + b_3x^4 + b_2x^3 + b_1x^2 + b_0x$$

x•b(x) thu được ở trên cần phải rút gọn theo mod m(x). Nếu $b_7 = 0$ thì kết quả được giữ nguyên. Nếu $b_7 = 1$, cần rút gọn theo mod m(x). Phù hợp theo kết quả ở trên, phép nhân với x (hệ 16 là '02'=(10) hệ 2) cần phải thực hiện ở mức byte như phép dịch vòng trái và sau đó XOR với '1B'. Toán tử này được định nghĩa bởi *b = xtime(a)*. Phép nhân với lũy thừa bậc cao của x, có thể thực hiện bằng lặp lại toán tử xtime(a) số lần phù hợp.

Chẳng hạn: '57'•'13' = 'FE', có thể thực hiện như sau:

$$'57' \bullet '02' = xtime(57) = 'AE'$$

$$'57' \bullet '04' = xtime(AE) = '47'$$

$$'57' \bullet '08' = xtime(47) = '8E'$$

$$'57' \bullet '10' = xtime(8E) = '07'$$

$$'57' \bullet '13' = '57' \bullet ('01' \oplus '02' \oplus '10') = '57' \oplus 'AE' \oplus '07' = 'FE'$$

e). Đa thức với các hệ số trong GF(2^8)

Các đa thức có thể được xác định với các hệ số trong GF(2^8). Bằng cách này véc tơ 4byte có thể biểu diễn bằng đa thức với bậc nhỏ hơn 4. Đa thức có thể được xác định bằng cách bổ sung thêm vào các hệ số phù hợp như là phép XOR của 2 véc tơ.

Giả thử chúng ta có 2 đa thức không thuộc GF(2^8):

$$a(x) = a_3x^3 + a_2x^2 + a_1x + a_0 \text{ và } b(x) = b_3x^3 + b_2x^2 + b_1x + b_0$$

Tích c(x) = a(x).b(x) được cho bởi:

$$c(x) = c_6x^6 + c_5x^5 + c_4x^4 + c_3x^3 + c_2x^2 + c_1x + c_0 \text{ với:}$$

$c_0 = a_0 \bullet b_0$

$c_1 = a_1 \bullet b_0 \oplus a_0 \bullet b_1$

$c_2 = a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2$

$c_3 = a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3$

$c_4 = a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3$

$c_5 = a_3 \bullet b_2 \oplus a_2 \bullet b_3$

$c_6 = a_3 \bullet b_3$

Rõ ràng c(x) cần phải không dài hơn véc tơ 4byte, vì vậy tích c(x) cần phải mod với đa thức bậc 4 nào đó. Trong RijnDael đó là đa thức M(x) = x^4 + 1. Tích c(x) = a(x).b(x) mod(x^4 + 1), được ký hiệu là d(x) = a(x)⊗b(x) và xác định như sau:

$$d(x) = d_3x^3 + d_2x^2 + d_1x + d_0 \text{ với các hệ số:}$$

$d_0 = a_0 \bullet b_0 \oplus a_3 \bullet b_1 \oplus a_2 \bullet b_2 \oplus a_1 \bullet b_3$

$d_1 = a_1 \bullet b_0 \oplus a_0 \bullet b_1 \oplus a_3 \bullet b_2 \oplus a_2 \bullet b_3$

$d_2 = a_2 \bullet b_0 \oplus a_1 \bullet b_1 \oplus a_0 \bullet b_2 \oplus a_3 \bullet b_3$

$d_3 = a_3 \bullet b_0 \oplus a_2 \bullet b_1 \oplus a_1 \bullet b_2 \oplus a_0 \bullet b_3$

Toán tử này được xác định như phép nhân b(x) với đa thức a(x) cố định, có thể biểu diễn như phép nhân ma trận dạng vòng các hệ số của a(x). Chúng ta có:

$$\begin{bmatrix} d_0 \\ d_1 \\ d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} a_0 & a_3 & a_2 & a_1 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_1 & a_0 & a_3 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Bởi vì đa thức $x^4 + 1$ không phải là đa thức tối giản, cho nên phép nhân bởi đa thức cố định không cho giá trị đảo duy nhất. Trong RijnDael chọn đa thức cố định thích hợp để có giá trị đảo duy nhất.

e1). Phép nhân với x

Nếu đem nhân b(x) với x, chúng ta có:

$$b_3x^4 + b_2x^3 + b_1x^2 + b_0x$$

[x ⊗ b(x)] thu được bằng cách lấy mod($x^4 + 1$) và được:

$$b_2x^3 + b_1x^2 + b_0x + b_3$$

Từ đây thấy rằng phép nhân b(x) với x tương đương với phép nhân ma trận ở trên, với mọi phần tử a_i = '00', ngoại trừ a_1 = '01'. Chúng ta có:

$$\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 00 & 00 & 00 & 01 \\ 01 & 00 & 00 & 00 \\ 00 & 01 & 00 & 00 \\ 00 & 00 & 01 & 00 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Như vậy, phép nhân với x, hoặc x bậc cao hơn, tương đương với phép dịch vòng véc tơ b(x).

3. THUẬT TOÁN RSA

Diffie và Hellman là những người đầu tiên xây dựng nguyên lí mật mã khoá công khai vào năm 1976. Tuy nhiên, trước đó vào những năm 1960 những tác giả ở NSA (National Security Agency - cơ quan an ninh quốc gia) đã xây dựng ý tưởng về xu thế của mật mã với khoá công khai. Vào năm 1977, Ron Rivest, Adi Shamir và Len Adleman làm việc ở MIT đã xây dựng thuật toán mật mã với khoá công khai. Công trình này được công bố vào năm 1978 và nhanh chóng được ứng dụng rộng rãi.

Sơ đồ RSA là sơ đồ mật mã khối, trong đó cả văn bản rõ và văn bản mã là các số nguyên trong dải từ 0 đến n-1 đối với một giá trị n nào đó. Trong mục này, trước hết chúng ta thảo luận diễn giải thuật toán RSA, sau đó sẽ trình bày các tính toán và vấn đề thám mã đối với RSA.

3.1. Miêu tả thuật toán.

Sơ đồ được thực hiện bởi Ron Rivest, Adi Shamir và Len Adleman dựa trên các biểu thức số mũ. Văn bản rõ được mã hoá bởi các khối, mỗi khối bao gồm các giá trị nhị phân, nhỏ hơn một

giá trị n đã cho nào đó. Điều đó có nghĩa rằng độ dài khối cần phải nhỏ hơn hoặc bằng $\log_2(n)$. Trong thực tiễn độ dài khối chọn bằng 2^k bit, ở đây $2^k < n \le 2^{k+1}$. Mã hoá với khối văn bản rõ M và giải mã với khối văn bản mã C, có thể trình bày dưới dạng công thức:

$$C = M^e \bmod n,$$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{e\,d} \bmod n,$$

Cả người gửi, lẫn người nhận đều phải biết giá trị n. Người gửi biết giá trị của e, còn chỉ có người nhận biết giá trị của d. Như vậy, sơ đồ này là thuật toán mật mã với khoá công khai: KU = {e, n } và khoá riêng KR = {d, n}. Để thuật toán này có thể sử dụng như là mật mã với khoá công khai cần phải thực hiện các đòi hỏi sau:

1. Cần phải tồn tại các giá trị: e, d và n để $M^{e\,d} = M \bmod n$, đối với tất cả $M < n$
2. Cần phải tương đối dễ tính toán M^e, C^d đối với tất cả các giá trị $M < n$.
3. Cần phải trên thực tế không có khả năng xác định d khi biết e và n.

Trước hết chúng ta hãy chú ý đến yêu cầu thứ nhất, các yêu cầu khác xem xét sau. Cần tìm một quan hệ dưới dạng: $M^{e\,d} = M \bmod n$

Từ định lí Euler, ta biết với hai số nguyên tố p và q và hai số nguyên m, n với $n = p\times q$ và $0 < m < n$, và tích của hai số nguyên k được thực hiện theo quan hệ sau:

$$m^{k\phi(n)+1} = m^{k(p-1)(q-1)+1} = m \bmod n,$$

Ở đây, $\phi(n)$ là hàm Euler, giá trị của $\phi(n) = (p-1)\times(q-1)$ là số nguyên dương, nhỏ hơn n và có quan hệ nguyên tố tương hỗ với n. Như vậy, đòi hỏi quan hệ nhận được với điều kiện:

$$e\,d = k\,\phi(n) + 1.$$

Điều này tương đương với quan hệ sau:

$$e\,d \equiv 1 \bmod \phi(n),$$

$$d \equiv e^{-1} \bmod \phi(n).$$

Có nghĩa là e và d là hai số có quan hệ nghịch đảo theo modul $\phi(n)$. Khi chú ý tới các qui tắc số học trong lớp số dư, điều đó có nghĩa rằng chỉ khi d (và tiếp tục cả e) là nguyên tố tương hỗ với $\phi(n)$. Viết một cách tương đương có quan hệ $\gcd(d, \phi(n)) = 1$ (gcd = great common divisor - ước số chung lớn nhất). Hai số có gcd =1, được gọi là hai số nguyên tố tương hỗ (nguyên tố cùng nhau).

Bây giờ chúng ta có thể tóm tắt toàn bộ sơ đồ RSA. Các thành phần của sơ đồ là:

p và q là hai số nguyên tố	(được chọn bí mật)
$n = p \times q$	(n được công bố công khai)
$\phi(n) = (p-1)(q-1)$	(được giữ bí mật)
chọn e để $\gcd(e, \phi(n)) = 1$, $1 < e < \phi(n)$	(được công bố công khai)
$d \equiv e^{-1} \bmod \phi(n)$,	(phải giữ bí mật)

Khoá riêng bao gồm KR = {d, n }, còn khoá công khai bao gồm KU = {e, n }. Giả thiết rằng người sử dụng A công bố khoá công khai của mình, còn người sử dụng B có thể sử dụng khoá công khai của A để gửi tin tức M của mình cho A. Khi người sử dụng B tính toán $C = M^e$ modn và gửi C. Khi nhận được văn bản mã này, người sử dụng A giải mã nó và tính $M = C^d$ modn.

Chúng ta dẫn ra cơ sở của thuật toán này. Chúng ta chọn e và d sao cho:

$$d \equiv e^{-1} \bmod \phi(n),$$

Như vậy,

$$e\,d \equiv 1 \bmod \phi(n).$$

Có nghĩa rằng ed có dạng $k\,\phi(n) + 1$. Nhưng theo định lí Euler, thì đối với hai số nguyên tố bất kì p và q và số nguyên $n = p \times q$ và M với $0 < M < n$, được tính theo quan hệ:

$$M^{k\phi(n)+1} = M^{k(p-1)(q-1)+1} \equiv M \bmod n,$$

Bởi vì $M^{e\,d} \equiv M$ modn, cho nên chúng ta có:

$$C = M^e \bmod n$$

$$M = C^d \bmod n \equiv (M^e)^d \bmod n \equiv M^{e\,d} \bmod n \equiv M \bmod n,$$

Trên hình 4.6. trình bày tóm tắt thuật toán RSA, còn trên hình 4.7. trình bày một thí dụ ứng dụng của chúng.

Thí dụ 1: trong thí dụ này, các khoá được tính như sau:

1. Chọn hai số nguyên tố, chẳng hạn p = 11và q = 17;
2. Tính tích của chúng : $n = p \times q = 11 \times 17 = 187$;
3. Tính $\phi(n) = (p-1)(q-1) = 10 \times 16 = 160$;
4. Chọn e là số nguyên tố tương hỗ với $\phi(n) = 160$ và phải nhỏ hơn $\phi(n)$. Trong trường hợp này chọn e =7. [$\phi(n) = 160 = 2^5 \times 5$; chọn e sao cho không là bội số của 2 và 5. Có nhiều giá trị e = 3, 7, 9, 11,13, 17,19...];
5. Xác định d để de = 1mod160 và d < 160. Giá trị phù hợp để chọn là d = 23.

Bởi vì $23 \times 7 = 161 = 1 \times 160 + 1$. [$d = ((k\phi(n)+1)/e)$ = số nguyên. Khi đã chọn e, thì d là duy nhất]. (Nếu chọn e = 11, thì tính được d = 131).

Kết quả của tính toán nhận được khoá công khai **KU** = {7, 187}, còn khoá riêng **KR** = {23, 187}. Trong thí dụ này đã sử dụng hai khoá trên với văn bản rõ M = 88. Khi mã hoá 88, sẽ được $C = 88^7 \bmod 187 = 11$. Bởi vậy, văn bản mã sẽ là 11. Khi giả mã tính $M = 11^{23} \bmod 187$, rõ ràng $11^{23} \equiv 88 \bmod 187$.

Tính toán các khoá

Chọn số p, q với p, q phải là số nguyên tố,

Tính $n = p \times q$

Tính $\phi(n) = (p - 1)(q - 1)$

Chọn số nguyên e: $\gcd(\phi(n), e) = 1,\ 1 < e < \phi(n)$

Tính số nguyên d: $d \equiv e^{-1} \bmod \phi(n)$, (hoặc $ed \bmod \phi(n) = 1$)

Khoá công khai : $KU = \{e, n\}$

Khoá riêng : $KR = \{d, n\}$

Mã hoá

Văn bản rõ : $M_i < n$

Văn bản mã : $C_i = M_i^e \pmod n$

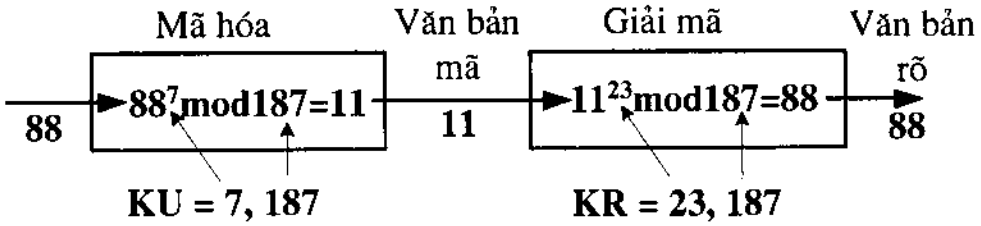
Giải mã

Văn bản mã : C_i

Văn bản rõ : $M_i = C_i^d \pmod n$

Hình 4.6. Thuật toán RSA.

Chú ý: Văn bản rõ M, trước khi mã hoá, được phân chia thành các khối $M_1, M_2, \ldots M_m$. Mỗi khối gồm một số các ký tự. Chẳng hạn hai ký tự. Mỗi ký tự được gán một con số hệ 10, bắt đầu từ chữ cái đầu tiên. Chẳng hạn, a= 01, b = 02,..., z = 26, ký tự trống được gán 00. Các khối M_i phải có giá trị (hệ 10) nhỏ hơn n



Hình 4.7. Thí dụ sử dụng thuật toán RSA.

Thí dụ 2: chọn p = 47, q = 59. Tính $n = p \times q = 47 \times 59 = 2773$ và $\phi(n) = (p-1) \times (q-1) = 2668$. Chọn e để $\gcd(\phi(n), e) = 1$. Chúng ta phân tích $\phi(n) = 2^4.667 = ?$. Các phương án chọn số e = 3, 5, 7, 9, 11, 13, 15, 17,..., 2667. Chúng ta chọn, chẳng hạn, e = 17. Khi đó số d được tính $d = (k.\ \phi(n)+1)/e$ = số nguyên. Từ đây, chúng ta tính được d = 157. Khi đó **KR** = {157, 2773}, còn **KU** = {17, 2773}.

3.2. Xem xét khía cạnh tính toán

Bây giờ chúng ta xem xét vấn đề phức tạp tính toán, được sử dụng trong thuật toán RSA. Chúng ta xem xét hai vấn đề: tính toán khoá và tính toán trong mã hoá/giải mã. Trước hết chúng ta khảo sát vấn đề các tính toán trong mã hoá và giải mã. Sau đó chúng ta quay trở lại vấn đề tính toán khoá.

a) Mã hoá và giải mã

Trong thuật toán RSA cả mã hoá và giải mã đều sử dụng phép tính nâng lên luỹ thừa của một số nguyên theo số mũ nguyên, rồi so sánh theo mod n. Nếu sử dụng phép tính trực tiếp nâng lên luỹ thừa nguyên rồi so sánh theo mod n, thì các giá trị trung gian sẽ rất cồng kềnh. Trong trường hợp riêng có thể sử dụng các đặc tính của số học trong lớp số dư:

$$[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n.$$

Đồng thời, chúng ta có thể tính toán kết quả trung gian theo mod n. Điều này tạo ra các khả năng thực tiễn tính toán.

Một vấn đề khác là hiệu quả của việc thực hiện các phép tính nâng lên luỹ thừa nguyên theo số mũ nguyên, là bởi vì khi sử dụng thuật toán RSA, chúng ta cần phải thực hiện với các số mũ lớn. Chẳng hạn cần tính x^{16}. Ta có thể tính trực tiếp bằng cách nhân x với 16 lần như sau:

$$x^{16} = x \times x \times x \times x \times x \times x \times x \times x \times x \times x \times x \times x \times x \times x \times x \times x.$$

Tuy nhiên, ta có thể thực hiện bằng phép tính bình phương bốn lần của số: x^2, x^4, x^8, để được x^{16}.

Trong trường hợp chung, chúng ta cần phải tính a^m với a, m là các số nguyên dương. Nếu trình bày m dưới dạng số nhị phân $b_k, b_{k-1}, \ldots, b_o$, thì ta có:

$$m = \sum_{bi \neq o} 2^i$$

và bởi vậy:

$$a^m = a^{\left(\sum_{bi \neq 0} 2^i\right)} = \prod_{bi \neq 0} a^{(2^i)}$$

Đồng thời để tính toán a^b modn, có thể sử dụng thuật toán được chỉ ra trên hình 4.8

$$a^m = \left[\prod_{bi \neq 0} a^{(2^i)}\right] \bmod n = \prod_{bi \neq 0} \left[a^{(2^i)} \bmod n\right]$$

```
c ← 0;   d ← 1;
for i ← k downto 0
        do c ← 2 × c
           d ← ( d × d ) modn
           if b_i = 1
           then c ← c + 1
             d ← ( d × a) modn
             return d
```

Hình 4.8: Thuật toán tính a^b modn.

(Thuật toán này có cả một chiều dài lịch sử, còn trình bày cụ thể ở đây trích từ sách của tác giả T. Cormen).

Trên hình 4.9 các bạn quan sát thấy các giá trị liên tiếp, nhận được trong quá trình tính toán của thuật toán này. Cần chú ý rằng biến c nói chung là không cần thiết, nó chỉ tham gia vào thuật toán để bảo đảm tính trực quan. Giá trị c cuối cùng là giá trị của số mũ cần tính toán.

i	9	8	7	6	5	4	3	2	1	0
b_i	1	0	0	0	1	1	0	0	0	0
c	1	2	4	8	17	35	70	140	280	560
d	7	49	157	526	160	241	298	166	67	1

Hình 4.9: Kết quả tính toán của thuật toán để tính nhanh nâng lên luỹ thừa đối với a^b modn, khi a = 7, b = 560 = 1000110000, n = 561.

Phép tính a^b mod n chiếm rất nhiều thời gian, vì vậy mã hoá RSA chạy chậm hơn DES hàng 1000 lần. Thế mà bản thân DES cũng đã rất chậm rồi.

b) Tính toán khoá

Trước hết chúng ta nên nhớ rằng đối với hệ mật mã với khoá công khai, mỗi một người tham gia cần phát sinh một cặp khoá. Có nghĩa là phải thực hiện bài toán sau:

- Xác định hai số nguyên tố p và q;
- Chọn một trong hai số hoặc e hoặc d và tính toán số còn lại.

Trước hết, chúng ta khảo sát thủ tục chọn số p và q. Biết rằng giá trị n = p × q sẽ được công khai cho mọi người, để không có khả năng tìm ra được số p và q nhờ phương án chọn đơn giản, thì các số nguyên p và q phải là các số nguyên tố có giá trị lớn. Đồng thời các phương pháp tìm ra tập hợp các số p và q có giá trị lớn phải có tính hiệu quả thực tiễn.

Ngày nay chưa có phương pháp hiệu quả để tính toán các số nguyên tố giá trị lớn. Bởi vậy, để giải quyết vấn đề này thường dẫn đến các mẹo riêng. Thông thường thủ tục này liên quan đến việc chọn một *số lẻ ngẫu nhiên*, có giá trị gần giá trị mong muốn và làm rõ nó có phải là số nguyên tố hay không? Nếu chứng tỏ rằng đó không phải là số nguyên tố, thì lại chọn một số lẻ tiếp theo, mặc dầu chưa chắc đó sẽ là số nguyên tố.

Để kiểm tra xem có phải là số nguyên tố hay không, thì có rất nhiều các *thử nghiệm*. Hầu hết các thử nghiệm như vậy mang *đặc trưng xác suất*. Điều đó chứng tỏ rằng thử nghiệm chỉ xác định số đã chọn là số nguyên tố theo xác suất. Chúng ta không xem xét sự tin tưởng hoàn toàn rằng các thủ tục ấy có giá trị xác suất gần tới 1 hay không. Để làm thí dụ trong chương 7 [6] sẽ miêu tả thuật toán Miller-Rabin, một trong các thuật toán hiệu quả nhất trong các thuật toán quen thuộc. Trong thuật toán này, cũng như trong phần lớn các thuật toán khác, thủ tục kiểm tra

số n có phải là số nguyên tố hay không, có liên quan đến hàng loạt phép tính, trong đó có sử dụng n và chọn ngẫu nhiên một số nguyên a nào đó. Nếu n không qua được cuộc thử nghiệm, thì n không phải là số nguyên tố. Nếu n chịu đựng được một cuộc thử nghiệm, thì n có thể là số nguyên tố, cũng có thể không là số nguyên tố. Nếu n chịu đựng được liên tiếp các cuộc thử nghiệm như vậy, với việc chọn ngẫu nhiên các giá trị a khác nhau, thì với mức tin tưởng cao rằng n chính là số nguyên tố.

Như vậy, thủ tục chọn số nguyên tố có thể diễn ra như sau:

1. Chọn một số nguyên lẻ n một cách ngẫu nhiên (chẳng hạn sử dụng sự phát sinh các số giả ngẫu nhiên).
2. Chọn một số nguyên a < n theo một cách ngẫu nhiên nào đó.
3. Thực hiện thử nghiệm xác suất trên số nguyên tố, chẳng hạn thử nhiệm Miller-Rabin. Nếu n không chịu đựng được các cuộc thử nghiệm, thì loại bỏ số n này và quay lại bước 1.
4. Nếu n chịu đựng được hàng loạt các thử nghiệm lặp lại, thì coi n xứng đáng là số nguyên tố. Nếu chưa tin tưởng, thì quay lại bước thứ 2.

Thủ tục này thật là vất vả. Tuy nhiên, cũng may là không phải thực hiện thường xuyên, chỉ khi cần phải phát sinh cặp khoá mới (KU, KR).

Đồng thời chúng ta cũng biết khá rõ về những số có biểu hiện chắc là số nguyên tố. Một từ những định lí của lí thuyết số được biết dưới tên gọi là *định lí về số nguyên tố*, tin tưởng vững chắc rằng, số nguyên tố gần số N được phân bố ở khoảng giữa của số nguyên N và ln(N). Như vậy, từ điều này, ta thấy rằng tất cả các số nguyên chẵn có thể bị gạt bỏ được ngay, trình tự chân thực của số lần thử nghiệm được cho bởi số ln(N)/2. Chẳng hạn, nếu cần tìm số nguyên tố trong vùng trị số khoảng 2^{200} thì để tìm được số nguyên tố đó, đòi hỏi gần $\ln(2^{200})/2 = 70$ thử nhiệm.

Sau khi xác định được các số nguyên tố p và q, quá trình tính toán khoá xoay quanh vấn đề chọn giá trị e và tính giá trị d, hoặc ngược lại, chọn giá trị d và tính giá trị e. Trong trường hợp thứ nhất, ban đầu cần thiết chọn giá trị e sao cho gcd[$\phi(n)$, e] = 1 và sau đó tính toán giá trị $d \equiv e^{-1} \bmod \phi(n)$. May mắn có một thuật toán để tính đồng thời *ước số chung lớn nhất* của hai số nguyên và nếu ước số chung lớn nhất bằng 1, thì việc xác định nghịch đảo đối với một từ các số nguyên theo modul của số kia. Đó là thuật toán mà sau này được gọi bằng tên chung Euclide. Thủ tục có liên quan đến sự phát sinh các số ngẫu nhiên và so sánh chúng với $\phi(n)$, cho đến khi nào sẽ không tìm thấy số nguyên tố tương hỗ với $\phi(n)$. Hiện tại một lần nữa, chúng ta có thể hỏi rằng các số ngẫu nhiên nhiều như thế nào để dẫn đến sẽ tìm ra được số thích hợp là số nguyên tố tương hỗ với $\phi(n)$?. Dễ dàng có thể chứng minh rằng: xác suất để hai lần chọn số ngẫu nhiên thì xác suất để chúng là nguyên tố tương hỗ bằng 0,6 - điều đó có nghĩa rằng để tìm một giá trị nguyên thích hợp cần đến tất cả một vài phép thử.

3.3. Độ an toàn của thuật toán RSA

Có ba khả năng dẫn đến thám mã thuật toán RSA như sau:

- **Phép chọn đơn giản**: Trong trường hợp này thực hiện kiểm tra tất cả các khả năng của khoá riêng.
- **Phép phân tích toán học**: Tồn tại một vài xu thế của loại này, nhưng tất cả chúng, theo bản chất, đều tương đương với bài toán xác định thừa số của tích hai số nguyên tố.
- **Phép phân tích thời gian tiêu tốn**: Dựa vào phép phân tích thời gian cần thiết thực hiện thuật toán giải mã.

Độ an toàn chống lại phép chọn đơn giản trong trường hợp thuật toán RSA, cũng như đối với mọi hệ thống mật mã khác, là sử dụng *không gian khoá lớn*. Với quan điểm này số bit trong e và d càng lớn càng tốt. Tuy nhiên, vì sự khá phức tạp khi phát sinh khoá, cũng như khi mã hoá/ giải mã, thì kích thước khoá càng lớn dẫn đến hệ thống làm việc càng chậm.

Trong mục này đối với thuật toán RSA, chúng ta trình bày tổng quan ngắn gọn về khả năng thám mã khi sử dụng xu thế phân tích thành thừa số và xu thế dựa trên sự phân tích thời gian tiêu tốn.

a) Vấn đề phân giải ra thành thừa số:

Có thể phân chia ra thành ba xu thế toán học khác nhau trong thám mã thuật toán RSA:

- Phân giải n ra thành hai thừa số nguyên tố. Điều đó dẫn đến tính được $\phi(n) = (p-1) \times (q-1)$ và trên cơ sở đó có thể xác định được $d = e^{-1}(\bmod \phi(n))$;
- Xác định trực tiếp $\phi(n)$, chứ không phải ngay từ đầu xác định p và q. Điều đó cũng cho phép xác định $d = e^{-1}(\bmod \phi(n))$;
- Xác định trực tiếp d, chứ không phải ngay từ đầu xác định $\phi(n)$.

Trong phần lớn trường hợp thảo luận vấn đề thám mã đối với thuật toán RSA, đều dẫn đến vấn đề phân giải một số nguyên n ra thành hai thừa số nguyên tố. Bài toán xác định $\phi(n)$ theo n, tương đương bài toán phân giải số n thành thừa số nguyên tố. Ngày nay, cũng đã chứng tỏ rằng thuật toán xác định d khi biết e và n, nó cũng đòi hỏi tiêu tốn thời gian như bài toán phân giải n thành hai thừa số nguyên tố. Như vậy, vấn đề thời gian tiêu tốn cho giải bài toán phân tích n ra thành thừa số nguyên tố, được sử dụng làm chỉ tiêu định lượng đánh giá độ an toàn của thuật toán RSA.

Bảng 4.3: Sự tiến bộ trong việc giải quyết vấn đề phân giải ra các thừa số

Số lượng số hệ 10	Gần đúng số lượng bit	Thời điểm giải	Số thời gian MIPS - năm yêu cầu	Thuật toán sử dụng
100	332	Tháng1-1991	7	Sàng bậc hai
110	365	Tháng1- 1992	75	Sàng bậc hai
120	398	Tháng6- 1993	830	Sàng bậc hai
129	428	Tháng1- 1994	5000	Sàng bậc hai
130	431	Tháng1- 1996	500	Sàng trong trường số dạng chung

Đối với số n lớn, để phân tích thành các thừa số nguyên tố lớn là một bài toán khó khăn. Để minh chứng cho điều này, chúng ta dẫn ra một thí dụ lịch sử. Vào năm 1977 khi thuật toán RSA mới ra đời, các độc giả của tờ tạp chí "*Scientific American*" đã nhận được một văn bản mã được trình bày trong mục "trò chơi toán học" của Martin Gardner. Ai giải được văn bản mã đó sẽ được nhận 100 đôla. Theo đánh giá, bài toán này không có thể giải sớm hơn 40^2 năm. Nhưng vào tháng giêng năm 1994 một nhóm sử dụng Internet đã cùng cộng tác với nhau để giải bài toán này trong vòng tám tháng. Bài toán này đã sử dụng khoá công khai dài 129 số trong hệ 10 (độ dài của n) và bằng khoảng 428 bit. Cũng giống như mật mã DES, phòng thí nhiệm RSA đã cấu trúc mật mã với độ dài khoá là: 100, 110, 120 và lớn hơn. Cuối cùng, để giải bài toán RSA-130, trong đó độ dài khoá bằng130 số trong hệ 10. Kết quả được trình bày trong bảng 4.5. Độ phức tạp của bài toán được trình bày trong đơn vị MIPS (million instruction per second) - năm, được hiểu là *khối lượng công việc được thực hiện trong vòng một năm của vi xử lí có thể thi hành một triệu lệnh trong vòng một giây, tương đương với thực hiện 3×10^{13} lệnh*. Như vậy, máy tính xây dựng trên Pentium với tần số 200 MHz có tốc độ gần 50 MIPS.

Từ bảng 4.3, chúng ta rút ra một kết luận quan trọng là: *vấn đề là phương pháp nào được sử dụng*. Đã từ lâu phương pháp được sử dụng để phân tích thành hai thừa số nguyên tố được gọi là *phương pháp sàng bậc hai*. Để thám mã RSA-130 đã sử dụng một thuật toán mới, được gọi là phương pháp *sàng trong trường số dạng chung*; nó cho phép rút ngắn khối lượng tính toán đến 10% so với khối lượng tính toán trước đây, để thám mã RSA-129.

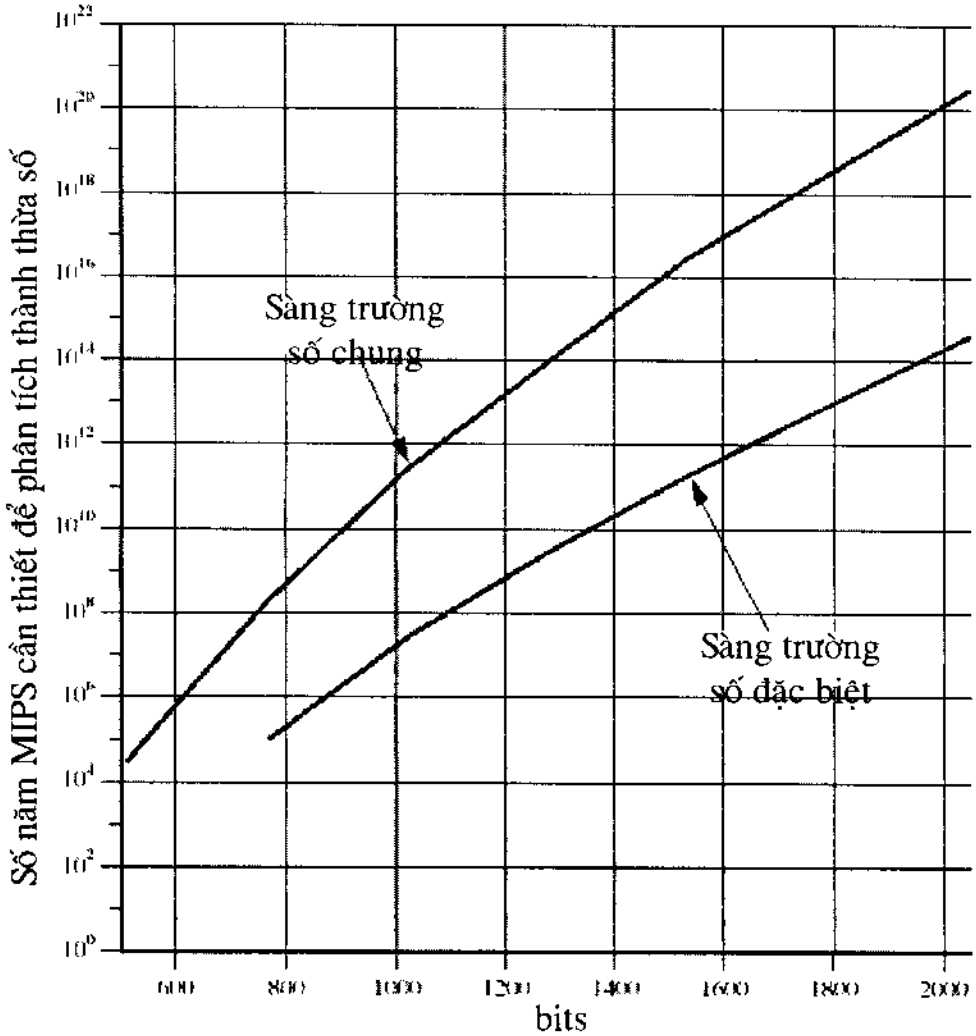
Nguy cơ đối với khoá có độ dài lớn có hai mặt: liên tục tăng khối lượng tính toán của các máy tính hiện đại và liên tục phải hoàn thiện thuật toán phân giải ra thừa số nguyên tố. Như đã thấy, khi ứng dụng phương pháp mới đã rút ngắn thời gian phân giải xuống đáng kể. Có thể chờ đợi sự hoàn thiện thuật toán sàng trong trường số dạng chung, không loại trừ khả năng xuất hiện thuật toán mới, còn hiện đại hơn. Chẳng hạn, ngày nay đã xuất hiện thuật toán sử dụng phương pháp sàng trong trường số của dạng đặc biệt, cho phép phân tích thành thừa số dạng riêng, nhanh hơn rất nhiều so với phương pháp sàng trong trường số dạng chung. Trên hình 4.10 trình bày kết quả so sánh của hai thuật toán này. Về logic mà nói, chúng ta chờ đợi một kết quả dẫn đến tăng khả năng về tốc độ thực hiện phép phân tích ra thừa số, trong trường hợp chung chỉ tiêu sẽ bằng hoặc lớn hơn các chỉ tiêu của phép sàng dưới dạng trường số đặc biệt. Bởi vậy, dẫn đến một điều là cần phải *rất thận trọng* trong việc chọn độ dài khoá trong thuật toán RSA. Trong một viễn cảnh không xa kích thước khoá sẽ được lựa chọn trong dải từ 1024 đến 2048 bit, hoặc dài hơn.

Đối với vấn đề chọn độ dài của n, các nghiên cứu đã chỉ ra hàng loạt các giới hạn khác. Để tránh khỏi việc chọn giá trị độ dài n, trong đó có thể dẫn đến sự phân tích thành thừa số với công sức nhỏ nhất, thì việc chọn p và q phải có những hạn chế sau:

1. Giá trị p và q cần phải khác nhau theo độ dài đến vài bậc. Chẳng hạn, cả p và q cần rơi vào khoảng giá trị từ 10^{75} đến 100^{100};
2. Cả (p - 1), cũng như (q - 1) cần có đặc tính khi phân tích thành thừa số, phải có được thừa số nguyên tố lớn;
3. gcd(p -1, q - 1) cần phải đạt được càng nhỏ càng tốt. Ngoài ra, như đã chứng minh, nếu e < n và d < $n^{1/4}$, thì d có thể xác định được dễ dàng.

b) Phân tích chi phí thời gian.

Nếu một ai đó đòi hỏi một chứng minh khác rằng độ an toàn của một thuật toán mật mã là một bài toán không đơn giản, thì tốt nhất chúng ta miêu tả phương pháp phân tích chi phí thời gian. Như đã được xác nhận bởi chuyên gia hiệu đính về mật mã Paul Kocher: đối phương có thể nhận được khả năng xác định khoá riêng nhờ phân tích chi phí thời gian cần thiết cho máy tính để giải mã tin tức. Phân tích chi phí thời gian có thể sử dụng không chỉ trong trường hợp RSA, mà còn đối với các hệ mật mã với khoá công khai khác. Các cuộc tấn công loại này thể hiện sự nguy hiểm theo hai nguyên nhân: chúng có thể dẫn ra xu hướng không chờ đợi nhất, và đưa ra tiên đoán rằng thám mã chỉ cần có văn bản mã.



Hình 4.10. Số năm MIPS cần thiết để phân tích thành thừa số.

Phương pháp phân tích chi phí thời gian trong chừng mực nào đấy tương tự như xu hướng bẻ khoá, đoán đúng ra tổ hợp ổ khoá của két sắt, sẽ đòi hỏi bao nhiêu thời gian để xoay vặn thiết bị khoá từ số này đến số kia. Có thể dẫn giải bản chất xu thế này qua thí dụ của thuật toán nâng lên luỹ thừa trong số học của lớp số dư, được trình bày trên hình 4.8, nhưng xu thế này có thể mô phỏng đối với sự thực hiện bất kỳ, mà không cần phải giả thiết sự thực hiện của thuật toán trong thời gian nhất định. Trong thuật toán nâng lên lũy thừa được thực hiện bit theo bit, với một phép nhân trong mỗi phép lặp và bổ sung thêm một phép nhân được thực hiện với mỗi bit có giá trị bằng 1.

Cũng như nhận xét của Kocher trong công trình của mình, về bản chất của xu thế này được trình bày trong thí dụ phù hợp với trường hợp nâng lên lũy thừa. Chúng ta giả thiết rằng hệ mật mã sử dụng hàm nhân trong lớp số dư được thực hiện rất nhanh hầu như trong tất cả các trường hợp, ngoài số lượng không lớn giá trị 1 của chúng, khi mà phép nhân chiếm thời gian lớn đáng

kể. Sự phân tích được tiến hành bit theo bit được bắt đầu từ bit b_k bên trái đầu tiên. Chúng ta cũng giả thiết rằng j bit đầu tiên cũng đã được biết. (Để nhận được giá trị của chỉ số mũ cần phải bắt đầu với j = 0 và lặp lại thủ tục phân tích cho đến khi không trở nên biết tất cả các bit của chỉ số). Để có được văn bản mã, đối phương có thể thực hiện j vòng lặp for đầu tiên. Trong bước tiếp theo sự thực hiện toán tử phụ thuộc vào giá trị bit chưa biết của chỉ số mũ. Nếu bit trong đó bằng 1, thì thực hiện phép toán $d \leftarrow (d \times a) \bmod n$. Đối với một vài giá trị của a và d phép nhân được *thực hiện đặc biệt chậm* và đối phương nhận biết điều đó. Bởi vậy, nếu thời gian thực hiện của thuật toán giải mã luôn luôn tăng một các đáng kể khi phép lặp cụ thể này được thực hiện với giá trị bit bằng 1, thì sự giữ chậm bị phát hiện và có thể chỉ ra sự kiện bit bằng 1 này. Nếu trong suốt quá trình quan sát mà thuật toán luôn luôn thực hiện nhanh ,thì các bit đó cần phải bằng 0.

Trong thực tiễn sự thực hiện phép nâng lên luỹ thừa trong lớp số dư không có biểu lộ một cách rõ ràng sự phụ thuộc theo thời gian, để thời gian thực hiện một vòng lặp có thể vượt thời gian trung bình để thực hiện toàn bộ thuật toán. Nhưng thực tiễn chứng tỏ rằng sự phân tích diễn biến thời thực hiện phép toán này là có ích. Những thông tin cụ thể hơn về chủ đề này độc giả có thể tìm trong sách của P.Kocher.

Mặc dù sự phân tích chi phí thời gian đã thể hiện nguy cơ nghiêm trọng, để chống lại nó có một vài cách đơn giản, chẳng hạn như sau:

- **Thực hiện toán tử nâng lên luỹ thừa với thời gian không đổi**: thực hiện thay đổi thuật toán sao cho tất cả các phép nâng lên luỹ thừa chiếm cùng khoảng thời gian như nhau, kể từ khi bắt đầu thực hiện cho đến khi đạt tới kết quả. Cách làm này đơn giản, nhưng khi đó sẽ làm tăng thời gian thực hiện thuật toán nói chung.
- **Các giữ chậm ngẫu nhiên**: có ảnh hưởng nhỏ đến thời gian chung của sự thực hiện gây ra bởi sự bổ sung trong thuật toán nâng luỹ thừa do các giữ chậm ngẫu nhiên, dẫn đến giảm nhỏ tác dụng của thám mã dựa trên phân tích chi phí thời gian. Nhưng đồng thời như nhận xét của Kocher, nếu sự phòng vệ do bổ sung các giữ chậm ngẫu nhiên không hoàn hảo, thì đối phương vẫn còn khả năng phân tích, ngay cả trong trường hợp có ảnh hưởng của các giữ chậm ngẫu nhiên này.
- **Che dấu**: Nhân văn bản mã với một số ngẫu nhiên như là thực hiện nâng lên luỹ thừa. Điều này không cho phép đối phương biết các bit như thế nào của văn bản mã đã được gia công bổ sung và đồng thời không tạo cho đối phương dẫn ra sự phân tích theo loạt, đây là xu thế riêng được thực hiện dựa trên sự phân tích chi phí thời gian.

Trong một vài sản phẩm của công ty *RSA Data security* có trình bày khả năng sử dụng hàm che phủ. Trong chúng toán tử $M = C^d \bmod n$ với khoá riêng được thực hiện như sau:

1. Phát sinh một số ngẫu nhiên bí mật r nằm trong dải từ 0 đến n-1;

2. Tính $C' = C^{r.e} \bmod n$, ở đây e là một giá trị của khóa công khai;

3. Tính $M' = (C')^d \bmod n$ đối với việc thực hiện RSA thông thường;

4. Tính $M = M' . r^{-1} \bmod n$, trong trường hợp này r^{-1} là nghịch đảo của giá trị r theo modn. Có thể chứng minh rằng kết quả sẽ hoàn toàn phù hợp nếu thay thế $r^{ed} = r \bmod n$.

RSA Data Security tin tưởng rằng khi sử dụng hàm che phủ thì hiệu suất chung của công việc thám mã sẽ hạ thấp xuống được theo trị số từ 2 đến 10%.

4. MẬT MÃ VỚI VIỆC SỬ DỤNG CÁC ĐƯỜNG CONG ELLIPTIC

Rất nhiều các sản phẩm và tiêu chuẩn, trong đó để mã hoá và chữ ký số sử dụng phương pháp mật mã với khoá công khai, được xây dựng dựa trên thuật toán RSA. Như chúng ta đã thấy số bit khóa cần thiết để sử dụng thuật toán RSA an toàn, trong những năm trở lại đây ngày càng tăng. Đó là nguyên nhân dẫn đến tăng tải cho hệ thống, gây khó khăn cho việc sử dụng thuật toán RSA. Điều này làm phát sinh hàng loạt các vấn đề, đặc biệt đối với các nút mạng, được thiết lập với máy tính điện tử, ở đây đòi hỏi an toàn của một số lớn các chuyển tiếp. Gần đây xuất hiện một xu hướng có thể cạnh tranh được với thuật toán RSA, đó là mật mã dựa trên các đường cong elliptic (ECC - Eliptic Curve Cryptography). Hiện tại đã xuất hiện việc xây dựng các tiêu chuẩn về xu thế mật mã này. Thí dụ như trong tiêu chuẩn IEEE P1363, hoặc ГОСТ P 34.10-2001 về mật mã với khóa công khai.

Sự hấp dẫn của xu hướng mật mã dựa trên cơ sở của các đường cong elliptic so với thuật toán RSA ở chỗ là: khi sử dụng đường cong elliptic, để bảo đảm độ an toàn tương đương nhau, thì độ dài của các con số có bậc không lớn, do đó làm giảm tải cho các bộ xử lý số. Mặc dù lý thuyết về mật mã sử dụng các đường cong elliptic đã nghe thấy từ lâu, nhưng chờ mãi gần đây mới xuất hiện một sản phẩm đáng chú ý về thám mã của loại mật mã này. Như vậy, độ tin tưởng về phương pháp mật mã sử dụng các đường cong elliptic còn chưa thể so sánh được với độ tin tưởng của mật mã dựa trên thật toán RSA.

Việc dẫn giải phương pháp mật mã sử dụng các đường cong elliptic khá khó khăn so với việc dẫn giải phương pháp RSA hay phương pháp Diffie - Hellman, bởi vậy, dẫn giải toán học một cách đầy đủ chỉ thấy lần đầu ở cuốn sách này. Trong mục này, sẽ trình bày cơ sở sử dụng các đường cong elliptic và mật mã sử dụng chúng.

4.1. Các đường cong elliptic

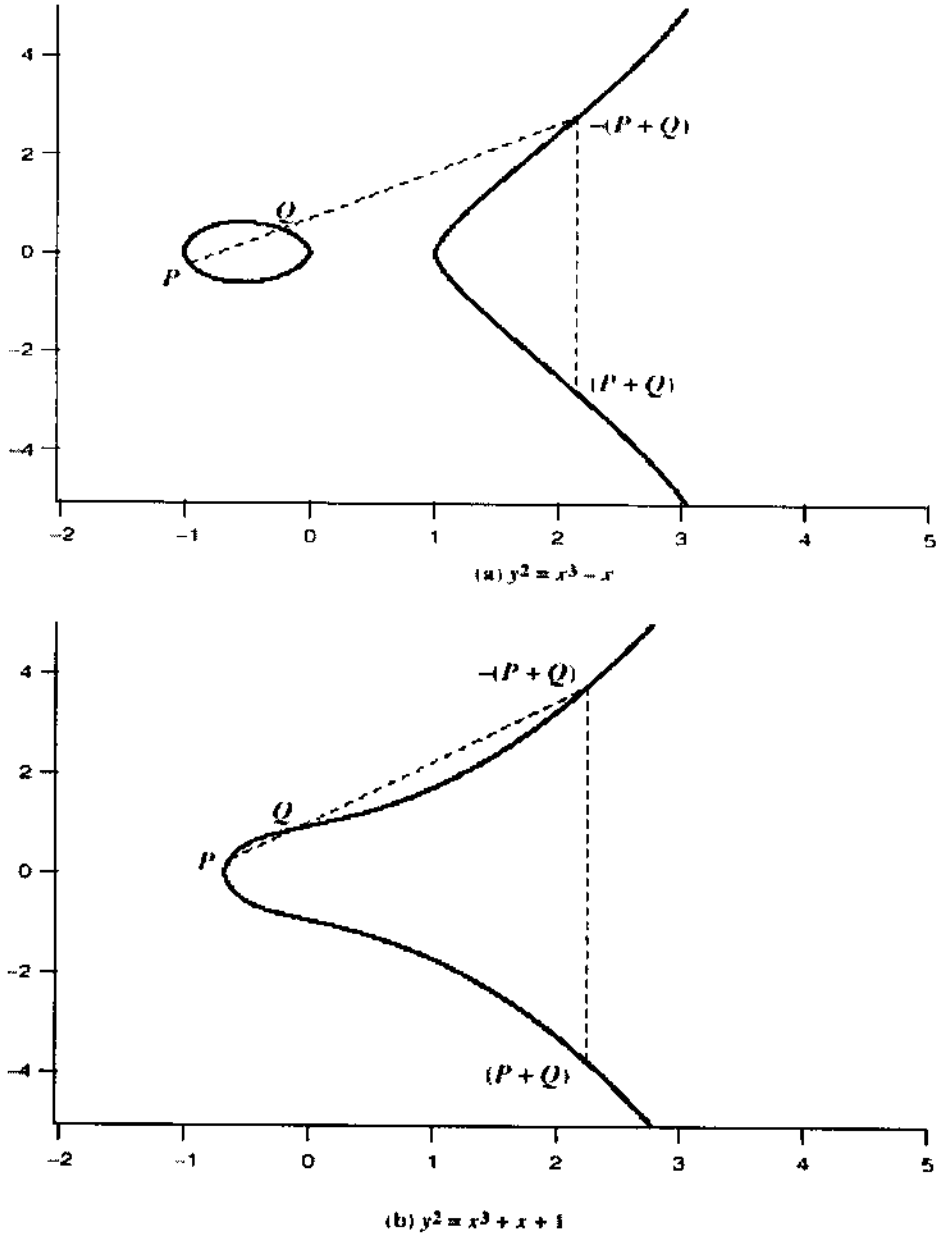
Các đường cong elliptic (EC) đó không phải là các ellip. Chúng được gọi đơn giản như vậy, bởi vì chúng được miêu tả bằng phương trình bậc ba. Tương tự như đã được sử dụng để miêu tả đường cong ellip. Trong trường hợp chung, phương trình bậc ba đối với các đường cong elliptic có dạng:

$$y^2 + axy + by = x^3 + cx^2 + dx + e,$$

ở đây a, b, c, e là các số thực, phù hợp với một vài điều kiện đơn giản nào đó. Việc định nghĩa đường cong elliptic cũng liên quan đến một phần tử được ký hiệu là $\mathcal{O}$ và được gọi là phần tử không riêng biệt (cũng gọi là phần tử vô cùng – point at infinity, hoặc phần tử không), ý nghĩa của nó chúng ta sẽ giải thích sau. Phương trình như trên gọi là phương trình lập phương hay phương trình bậc ba, bởi vì trong phương trình bậc cao nhất bằng ba. Hai thí dụ về các đường cong elliptic, được trình bầy trên hình 4.11. Như vậy, chúng ta đã nhìn thấy, đôi khi những công thức như thế sinh ra dạng đường cong rất kỳ lạ.

Từ định nghĩa của đường cong elliptic có thể dẫn ra như sau: nếu có ba điểm trên đường cong elliptic nằm trên cùng một đường thẳng thì tổng của chúng là $\mathcal{O}$. Từ đó đưa ra một định nghĩa của quy tắc toán tử về phương pháp cộng các điểm trên đường cong elliptic:

1. Đối tượng $\mathcal{O}$ đóng vai trò là phần tử $\mathcal{O}$, có tọa độ $\mathcal{O} = (x, \infty)$, khi cộng, sao cho $\mathcal{O} = -\mathcal{O}$ và đối với điểm P bất kỳ nào trên đường cong elliptic có $P + \mathcal{O} = P$.
2. Một đường thẳng đứng cắt đường cong ở hai điểm với cùng một toạ độ x, chúng ta chỉ ra $P_1 = (x, y)$ và $P_2 = (x, -y)$. Đường thẳng đứng này cũng cắt đường cong ở điểm vô cùng. Bởi vậy $P_1 + P_2 + \mathcal{O} = \mathcal{O}$ và $P_1 = -P_2$. Như vậy, điểm với dấu trừ là điểm có cùng toạ độ x, nhưng ngược dấu toạ độ y. Quy tắc này được phản ánh trên hình 4.11.
3. Để cộng hai điểm Q và R có toạ độ x khác nhau, hãy dẫn qua các điểm này một đường thẳng và hãy tìm điểm thứ ba P_1 là giao điểm của đường thẳng với đường cong elliptic. Dễ dàng chỉ ra rằng chỉ tồn tại một điểm như thế (nếu đường thẳng này không là đường tiếp tuyến với đường cong hoặc tại điểm Q hoặc tại điểm R, trong trường hợp như thế cần thiết đặt $P_1 = Q$ hoặc $P_1 = R$ tương ứng). Sau đó cần nhớ rằng: $Q + R + P_1 = \mathcal{O}$, tiếp tục có $Q + R = -P_1$. Cấu trúc này cũng được trình bày trên hình 4.11.
4. Để tăng gấp đôi điểm Q, hãy dẫn ra một đường tiếp tuyến tại điểm Q và hãy tìm một điểm cắt S khác. Khi đó : $Q + Q = 2Q = -S$.



Hình 4.11. Các thí dụ về đường cong elliptic.

Các quy tắc trình bày trên của phép cộng cũng tuân theo các đặc tính thông thường của phép cộng. Chẳng hạn như quy luật giao hoán và kết hợp. Phép nhân của điểm P trên đường cong elliptic với một số nguyên, dương k, cũng được xác định như là tổng k chập của điểm P, như sau: 2P = P + P; 3P = P + P + P; v.v...

4.2. Các đường cong elliptic trong trường hữu hạn

Trong trường hợp mật mã với sử dụng các đường cong elliptic được diễn ra trong dạng đơn giảm bớt của đường cong elliptic, trong đó được định nghĩa trong một trường hữu hạn.

Điều quan tâm đặc biệt đối với mật mã là đối tượng, được gọi là **nhóm elliptic theo modulp**, ở đây p là một số nguyên tố. Nhóm như vậy, được xác định như sau. Hãy chọn hai số nguyên không âm là a và b (a, b phải nhỏ hơn p) thoả mãn điều kiện sau:

$$4a^3 + 27\,b \pmod p \neq 0$$

Điều kiện này bảo đảm phương trình $f(x) = x^3 + ax + b = 0$, chỉ có một nghiệm.

Khi đó $E_p(a,b)$, ký hiệu nhóm elliptic theo modulp, là tập hợp các điểm mà tọa độ (x, y) là số nguyên không âm nhỏ hơn p và phù hợp với điều kiện sau:

$$y^2 \equiv x^3 + ax + b \pmod p \qquad (4.5)$$

cùng với điểm ở xa vô cùng , O.

Thí dụ: Giả thử, chọn p = 23. Chúng ta khảo sát đường cong elliptic $y^2 = x^3 + x + 1$. Trong trường hợp này a = b = 1 và chúng ta có $4 \times 1^3 + 27 \times 1 \pmod{23} = 8 \neq 0$, nó phù hợp với điều kiện của các nhóm elliptic theo mudul 23.

Cần chú ý rằng phương trình trong thí dụ này, trước đó đã được vẽ trên hình 4.11b). Trên hình vẽ biểu diễn là đường cong liên tục, các điểm sẽ cho giá trị phù hợp với phương trình đã cho. Đối với các nhóm elliptic chỉ được khảo sát các giá trị nguyên từ (0, 0) đến (p, p) trong góc phần tư của các số không âm (góc thứ nhất), phù hợp với phương trình theo modul p. Trong bảng 4.4. trình bày các điểm (khác với O) là các phần tử của trường $E_{23}(1,1)$. Trong trường hợp chung, một bảng kê như thế được tạo ra theo qui tắc sau:

1). Đối với mỗi giá trị nguyên của x, với $0 \leq x < p$, tính: $x^3 + ax + b \pmod p$;

2). Đối với mỗi giá trị tính được ở bước trên, sẽ có hay không giá trị căn bậc hai theo modulp. Nếu không có, thì trong nhóm $E_p(a,b)$ không có điểm với giá trị x đó. Nếu tồn tại phép khai căn, thì có hai giá trị y phù hợp với toán tử khai căn bậc hai (hãy bỏ qua trường hợp duy nhất là khi y = 0). Các giá trị (x, y) này sẽ là các điểm của nhóm $E_p(a,b)$.

Qui tắc đối với phép cộng trong nhóm $E_p(a,b)$ phù hợp với phương pháp hình học trình bày trên hình 4.11. Hình thức của các thu nhận này đối với tất cả các điểm $E_p(a,b)$ có thể được mô tả như sau:

1). $P + O = P$;

2). Nếu P = (x , y) thì P + (x , - y) = O. Điểm (x, -y) là giá trị âm của điểm P và kí hiệu là - P. Cần thấy rằng (x, -y) nằm trên đường cong elliptic, có thể chứng minh bằng đồ thị (xem hình 4.11b) và thuộc về $E_p(a,b)$. Thí dụ trong trường hợp $E_{23}(1, 1)$ đối với điểm P = (13, 7), chúng ta có – P = (13 , -7), nhưng –7 mod23 = 16;

3). Như vậy – P = (13 , 16), như chúng ta đã thấy trong bảng các giá trị của E_{23} (1, 1);

Bảng 4.4. Các điểm $E_{23}(1, 1)$ trên đường cong elliptic.

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

Các điểm này được trình bày trên hình 4.12.

4). Nếu $P = (x_1, y_1)$ và $Q = (x_2, y_2)$, ở đây $P \neq Q$, thì $P + Q = (x_3, y_3)$ được xác định phù hợp với qui tắc sau:

$$x_3 \equiv \lambda^2 - x_1 - x_2 \pmod{p},$$

$$y_3 \equiv \lambda (x_1 - x_3) - y_1 \pmod{p},$$

ở đây:

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & nê'u\ P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, & nê'u\ \ P = Q \end{cases}$$

Chúng ta khảo sát hai thí dụ sau. Giả thử P = (3, 10) và Q = (9, 7). Khi đó :

$$\lambda = \frac{7-10}{9-3} = \frac{-3}{6} = \frac{-1}{2} \equiv 11 \bmod 23$$

$$x_3 = 11^2 - 3 - 9 = 109 \equiv 17 \bmod 23$$

$$y_3 = 11(3 - 9) - 10 = -76 \equiv 16 \bmod 23$$

Bởi vậy P + Q = (17, 16). Để tìm 2P chúng ta xác định:

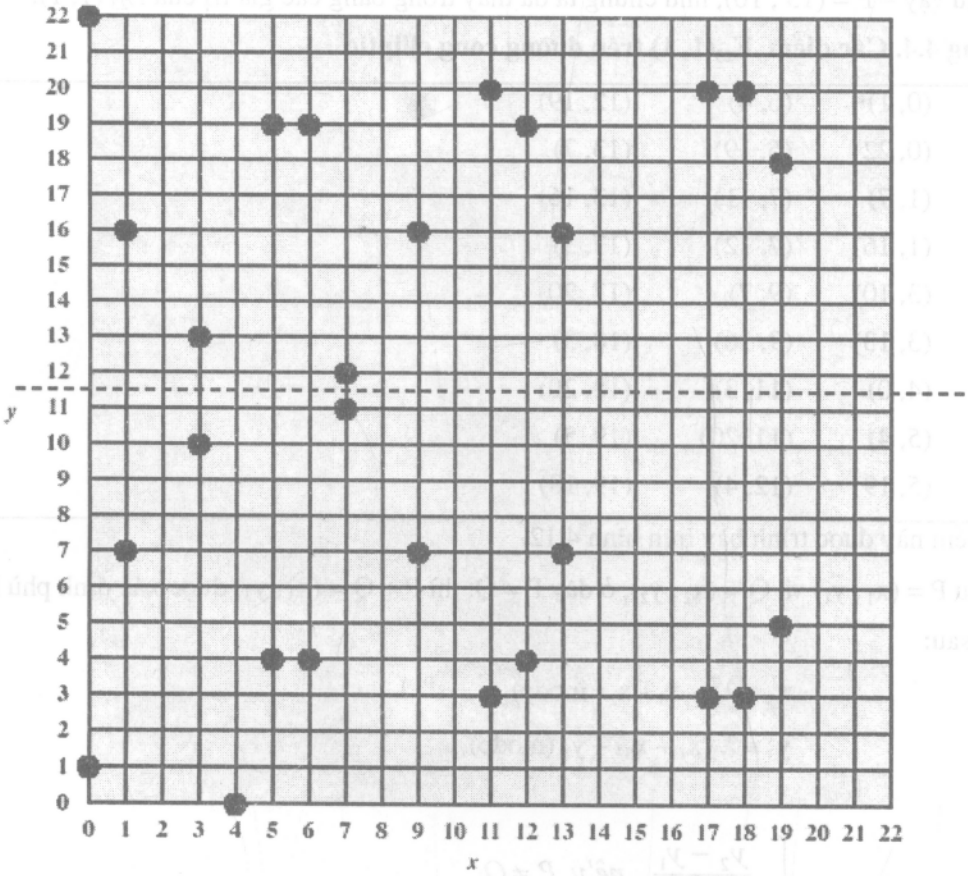
$$\lambda = (3(3^2) + 1)/(2 \times 10) = 28/20 = 7/5 \equiv 6 \bmod 23$$

$$x_3 = 6^2 - 3 - 3 = 30 \equiv 7 \bmod 23$$

$$y_3 = 6(3 - 7) - 10 = -34 \equiv 12 \bmod 23$$

và bởi vậy 2P = (7, 12).

Cũng cần nhớ rằng phép nhân được định nghĩa như là sự lặp lại của sử dụng toán tử cộng: 4P = P + P + P + P.



Hình 4.12. Các điểm $E_{23}(1, 1)$ trong trường elliptic hữu hạn.

4.3. Các đường cong elliptic và mật mã

Toán tử cộng trong mật mã dựa trên cơ sở của các đường cong elliptic là tương tự như toán tử nhân theo modul của số nguyên tố trong RSA, còn lặp lại nhiều lần phép cộng tương tự như phép nâng lên luỹ thừa. Để xây dựng hệ thống mật mã trên cơ sở sử dụng đường cong elliptic, cần phải tìm " vấn đề khó khăn " tương tự như phân tích thành thừa số của tích hai số nguyên tố hoặc là logarit rời rạc.

Chúng ta hãy khảo sát phương trình Q = k P, ở đây Q, P $\in E_p(a,b)$ và k < p. Sẽ dễ dàng tính Q khi biết k và P, nhưng tương đối rất khó khăn xác định k nếu có P và Q.

Trong phần này, chúng ta miêu tả ba xu thế hướng tới sử dụng đường cong elliptic trong mật mã, để các bạn có thể đánh giá việc ứng dụng kĩ thuật này.

a). Thuật toán trao đổi khoá theo sơ đồ Diffie - Hellman

Trao đổi khoá với việc sử dụng đường cong elliptic có thể thực hiện như sau: đầu tiên chọn ra một số nguyên tố $p \approx 2^{180}$ và tham số a và b đối với đường cong elliptic trong phương trình (4.5). Từ đó xác định nhóm điểm elliptic $E_p(a,b)$. Sau đó, trong $E_p(a,b)$ chọn một điểm phát sinh G = (x_1, y_1). Khi chọn G điều quan trọng là cần phải bảo đảm để giá trị q nhỏ nhất, khi đó

q×G = $\mathcal{O}$, mà q phải là số nguyên tố có giá trị rất lớn. Nhóm $E_p(a,b)$ và G trong hệ thống mật mã là các tham số được biết bởi tất cả các người tham gia.

Trao đổi khoá giữa hai người sử dụng A và B có thể diễn ra theo sơ đồ sau:

1). Phía A chọn một số nguyên n_A, với $n_A < q$. Số này là khoá riêng của người tham gia A. Sau đó, phía A phát sinh khoá công khai $P_A = n_A \times G$. Khoá công khai là một điểm nào đó trong $E_p(a,b)$;

2). Cũng tương tự như A, người tham gia B chọn một khoá riêng n_B và tính khoá công khai P_B;

3). Phía A phát sinh khoá mật $K = n_A \times P_B$, tương tự như thế, phía B phát sinh khoá mật $K = n_B \times P_A$.

Cả hai công thức trong bước ba, đều cho chung một kết quả, bởi vì :

$$n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A.$$

Để bẻ khoá sơ đồ này, đối phương cần phải tính giá trị k khi biết G và kG, đó là điều rất khó khăn.

Để minh chứng, chúng ta dẫn ra một thí dụ, được tính toán bởi Ed Shaefer ở đại học tổng hợp Canta - Klara. Chọn p = 211, nhóm $E_p(0, -4)$, nó phù hợp với đường cong $y^2 = x^3 - 4$ và điểm sinh G = (2, 2). Có thể thấy rằng 241× G = $\mathcal{O}$. Khoá riêng của người tham gia A chọn là $n_A = 121$, vì vậy khoá công khai của A là $P_A = 121(2, 2) = (115, 48)$. Khoá riêng của người sử dụng B chọn là $n_B = 203$, bởi vậy khoá công khai của B là $P_B = 203(2, 2) = (130, 203)$. Khoá mật chung của hai người là 121(130, 203) = 203(115, 48) = (161, 169).

Cần chú ý rằng khoá mật chung của hai người là một cặp số. Nếu khoá mật sẽ được sử dụng làm khoá phiên đối với mật mã truyền thống, thì từ cặp số này cần thiết phải phát sinh một giá trị phù hợp. Có thể, chẳng hạn, sử dụng đơn giản toạ độ x, hoặc một hàm đơn giản nào đó của x.

b). Mã hoá/ giải mã với việc sử dụng đường cong elliptic

Trong các tài liệu tham khảo chuyên biệt có thể tìm thấy sự phân tích một vài xu hướng về mã hoá và giải mã sử dụng đường cong elliptic. Trong mục này, chúng ta hãy khảo sát trường hợp đơn giản nhất trong các xu thế đó. Bài toán đầu tiên trong hệ thống khảo sát là mã hoá văn bản rõ M, cần phải gửi đi dưới dạng tọa độ (x, y) đối với điểm P_M. Ở đây điểm P_M sẽ là văn bản mã và sẽ cần giải mã ở phía kia. Cần chú ý rằng chúng ta chưa có thể mã hoá tin tức thành toạ độ đơn giản x và y của một điểm, bởi vì không phải tất cả các điểm như vậy có trong $E_p(a,b)$ (xem thí dụ ở bảng 4.4). Có một vài xu thế liên quan tới việc mã hoá này, nhưng chúng ta sẽ không khảo sát chúng. Đối với mục đích của chúng ta là đạt được một cách đơn giản, chúng ta có nhận xét trực tiếp về phương pháp mà có thể ứng dụng ở đây.

Cũng như trong trường hợp trao đổi khoá, trong hệ thống mã hoá và giải mã để dùng làm tham số cũng khảo sát điểm sinh G và nhóm elliptic $E_p(a,b)$. Người sử dụng A chọn khoá riêng n_A và phát sinh khoá công khai $P_A = n_A \times G$. Để mã hoá và gửi tin tức P_M cho người sử dụng B, người sử dụng A chọn một số dương ngẫu nhiên k (k < q) và tính văn bản mã C_M, bao gồm các cặp điểm:

$$C_M = \{k\,G, P_M + k\,P_B\}.$$

Chú ý rằng phía A sử dụng khoá công khai của B: P_B. Để giải mã văn bản mã đó, B nhân điểm thứ nhất với khoá mật của B và lấy điểm thứ hai trừ đi kết quả, được:

$$P_M + k\,P_B - n_B(k\,G) = P_M + k\,(n_B\,G) - n_B(k\,G) = P_M.$$

Người sử dụng A "che giấu" tin tức P_M nhờ bổ sung tới nó giá trị kP_B. Không ai, ngoài người sử dụng này biết giá trị k. Bởi vậy, mặc dù P_B là khoá công khai, nhưng không ai có thể tách ra được phần mặt nạ kP_B. Tuy nhiên, người sử dụng A đã che phủ tin tức, nhưng cũng "gợi ý" rằng có thể tháo bỏ được mặt nạ đó, nếu ai đó có khoá riêng n_B. Đối phương muốn khôi phục lại tin tức cần phải tính được k khi biết G và kG, đây thực sự là bài toán cực kỳ khó.

Để làm thí dụ minh chứng, chúng ta dẫn ra một kết quả tính toán sau. Khảo sát trường hợp p =751, $E_p(-1, 188)$ (phù hợp với đường cong $y^2 = x^3 - x + 188$) và điểm sinh G(0, 376).

Giả sử rằng người sử dụng A gửi cho người sử dụng B một tin tức, trong đó được mã hoá thành điểm trên đường cong elliptic $P_M(562, 201)$ và người sử dụng A chọn một số ngẫu nhiên k = 386. Khóa công khai của B là P_B = (201, 5). Chúng ta có 386(0, 376) = (676, 558) và (562, 201) + 386(201, 5) = (385, 328). Như vậy, người sử dụng A cần gửi văn bản mã {(676, 558), (385, 328)}.

c). Hình thành và kiểm tra chữ ký số với sử dụng đường cong elliptic

Chúng ta dẫn ra sau đây các bước thực hiện chuẩn chữ ký số (CKS) của Cộng hòa liên bang Nga (ГОСТ Р 34.10-2001). Chuẩn CKS này là mật mã với khóa công khai được xây dựng trên cơ sở sử dụng đường cong elliptic. Quá trình hình thành như sau:

1). Phát sinh số nguyên tố p và EC theo công thức (4.5). Từ đây xác định tập các điểm trên EC, trong hệ tọa độ Đềcác (x, y), ký hiệu $E_p(a,b)$. Trong nhóm $E_p(a,b)$ chọn ra một điểm sinh G, với điều kiện q là giá trị nhỏ nhất để $qG \times O$, q phải là số nguyên tố có giá trị lớn (q < p). Nhóm $E_p(a,b)$ và G phải công khai cho mọi người tham gia;

2). Chọn ngẫu nhiên chuỗi số nguyên $n_1, n_2, \ldots, n_n$ với $1 < n_i < q$, để tạo thành khóa riêng của mỗi người tham gia;

3). Chuỗi khóa công khai $P_1, P_2, \ldots, P_n$, được xác định theo công thức $P_i = n_i \times G$;

4). Chọn hàm băm H, tính bản tóm lược của tin tức M, được h = H(M);

5). Chọn ngẫu nhiên số nguyên t, 0 < t < q; theo t tính điểm $R = t \times G = (x_R, y_R)$ trên EC;

6). CKS hình thành từ cặp (r, s), được xác định theo công thức sau:

$$r = x_R \bmod q \quad \text{và} \quad s = (t.h + r.n_i) \bmod q$$ với i là người tham gia ký.

Quá trình kiểm tra CKS, gồm 2 bước sau:

1). Tính $v = s.h^{-1} \bmod q$ và $w = (q - r.h^{-1}) \bmod q$. Từ đây phát sinh điểm $R' = v.G + w.P_i = (x_{R'}, y_{R'})$. Tính A = $x_{R'} \bmod q$;

2). Hình thành vế thứ 2 của phép kiểm tra bằng cách copy r, tức lấy B = r.

Nếu A ≡ B, CKS được xác minh là đúng chủ.

Dễ dàng chứng minh được R'≡ R. Thực vậy:

- $v.G = s.h^{-1} \bmod q.G = (t.h + r.n_i).h^{-1}.G \bmod q =$

  $= (t.G + r.h^{-1}.P_i) \bmod q = (R + r.h^{-1}.P_i) \bmod q$;

- $w.P_i = (q - r.h^{-1}).P_i \bmod q = (-r.h^{-1}.P_i) \bmod q$.

Cuối cùng, chúng ta có:

$$R' = (R + r.h^{-1}.P_i) \bmod q - (r.h^{-1}.P_i) \bmod q = R$$

Nhược điểm của các chuẩn CKS hiện hành là khi cần n (n ≥ 2) người ký vào cùng một văn bản, thì cần lưu giữ n CKS.. Chúng tôi đã thử nghiệm hình thành và kiểm tra CKS tập thể bảo đảm cùng độ an toàn, nhưng độ dài CKS coi như giảm đi tương đương n lần.

4.4. Sự an toàn mật mã với sử dụng đường cong elliptic

Sự an toàn bảo đảm cho xu hướng mật mã dựa trên việc sử dụng đường cong elliptic phụ thuộc vào sự khó khăn như thế nào, khi phải giải bài toán xác định k khi biết P và kP. Bài toán này thường được gọi là *vấn đề logarit hoá trên đường cong elliptic*. Phương pháp nhanh nhất hiện nay, từ tất cả các phương pháp đã biết, được gọi là "phương pháp ρ" của Pollard. Trong bảng 4.5. so sánh hiệu quả của phương pháp này và phương pháp phân giải ra thừa số nguyên tố với sự giúp đỡ của phép sàng trong trường số dạng chung. Như độc giả đã thấy, theo phép so sánh với RSA, trong trường hợp ứng dụng phương pháp mật mã dựa trên cơ sở sử dụng đường cong elliptic, với cùng mức an toàn thì độ dài khoá giảm đáng kể.

Khi độ dài khoá như nhau, thì yêu cầu công sức tính toán khi sử dụng RSA và mật mã dựa trên cơ sở đường cong elliptic là không khác nhau nhiều. Như vậy, khi bảo đảm cùng độ an toàn thám mã như nhau, nếu đem so sánh mật mã dựa trên đường cong elliptic với thuật toán RSA, thì độ dài khoá ngắn hơn nhiều.

Bảng 4.5. Khối lượng tính toán để thám mã khi sử dụng đường cong elliptic và RSA

a). Kích thước khóa	Số năm MIPS	b). Kích thước khóa	Số năm MIPS
150	$3{,}8\times10^{10}$	512	3×10^{4}
205	$7{,}1\times10^{18}$	768	3×10^{8}
234	$1{,}6\times10^{28}$	1024	3×10^{11}
a) Logarit hóa trên đường cong elliptic theo phương pháp ρ của Pollard.		1280	3×10^{14}
		1536	3×10^{16}
		2048	3×10^{20}

b). Phân giải ra thừa số theo phương pháp sàng trong trường số dạng chung.

5. THUẬT TOÁN MD5 TÍNH PROFIL TIN TỨC

Thuật toán MD5 để tính profil tin tức đã được sáng chế bởi Ron Rivest thuộc Đại học công nghệ Massachuset (MIT) (thuật toán mật mã RSA lần đầu tiên xuất bản mang tên Rivest). Đến thời gian gần đây, khi xuất bản lần cuối, đã tăng cường khả năng chống lại thám mã theo phương pháp vét cạn mọi phương án, cũng như trong cả thám mã, MD5 là thuật toán lưu truyền rộng rãi nhất trong số thật toán hàm băm an toàn.

5.1. Logic của MD5

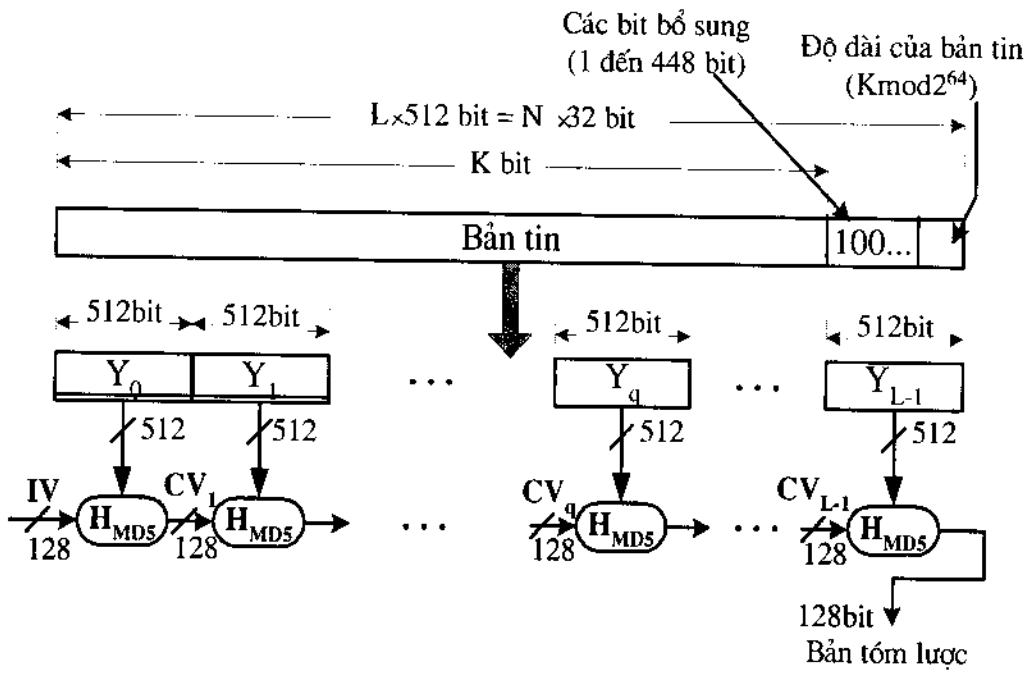
Thuật toán lấy làm số liệu đầu vào các tin tức có độ dài bất kỳ và đưa ở đầu ra 128 bit giá trị profil tin tức. Số liệu đầu vào được gia công thành các khối 512 bit liên tiếp.

Trên hình 4.13 biểu diễn sơ đồ chung gia công tin tức khi tính profil của nó. Sơ đồ này phù hợp với cấu trúc chung hơn, đã được nghiên cứu. Quá trình gia công tin tức bao gồm các giai đoạn sau:

♦**Bước 1: Bổ sung các bit làm đầy**. Tin tức được bổ sung, sao cho độ dài chung của nó có dạng 448 theo modul512 (nghĩa là độ dài ≡ 448 mod512). Độ dài của tin tức bổ sung trong dạng bit cần phải nhỏ hơn hoặc bằng 448bit, để tạo thành (kể cả 64bit để ghi độ dài tin tức) nguyên lần khối 512 bit. Toán tử bổ sung luôn luôn được thực hiện, ngay cả khi tin tức khởi đầu K= nguyên lần khối 512bit. Chẳng hạn, độ dài của tin tức bằng 512bit, thì còn bổ sung thêm 448 bit, kết quả độ dài của tin tức sau bổ sung bằng 960 bit. Bởi vậy, số lượng bit bổ sung phải là từ 1 đến 448 bit. Cấu tạo của các bit bổ sung như sau: bit đầu tiên bằng 1, tất cả các bit còn lại bằng 0.

♦**Bước 2: Bổ sung các bit để ghi giá trị độ dài**. Khối 512 bit cuối cùng dành 64 bit để ghi độ dài của tin tức khởi đầu (tức là độ dài của tin tức trước khi được bổ sung). Nếu độ dài của tin tức

vượt 2^{64}bit, thì chỉ ghi giá trị độ dài phần dư của Kmod2^{64}. Như vậy, trường này ghi độ dài của tin tức khởi đầu theo modul 2^{64}.



Hình 4.13. Tính toán bản tóm lược tin tức của MD5.

♦**Bước 3: Giá trị đệm khởi đầu MD.** Để lưu giữ các kết quả trung gian và cuối cùng của hàm băm, sẽ sử dụng 128 bit đệm. Đệm có thể trình bày dưới dạng bốn từ 32 bit (A, B, C, D). Các từ đệm này với giá trị khởi đầu (trong dạng cơ số 16) như sau:

A = 67452301;

B = EFCDAB89;

C = 98BADCFE;

D = 10325476.

Các giá trị này được lưu giữ trong dạng trình tự trực tiếp ghi các byte, khi đó byte có giá trị nhỏ nhất của từ được đặt ở vị trí với địa chỉ nhỏ nhất. Như là các dòng 32 bit, giá trị khởi đầu (theo cơ số 16) sẽ có dạng như sau:

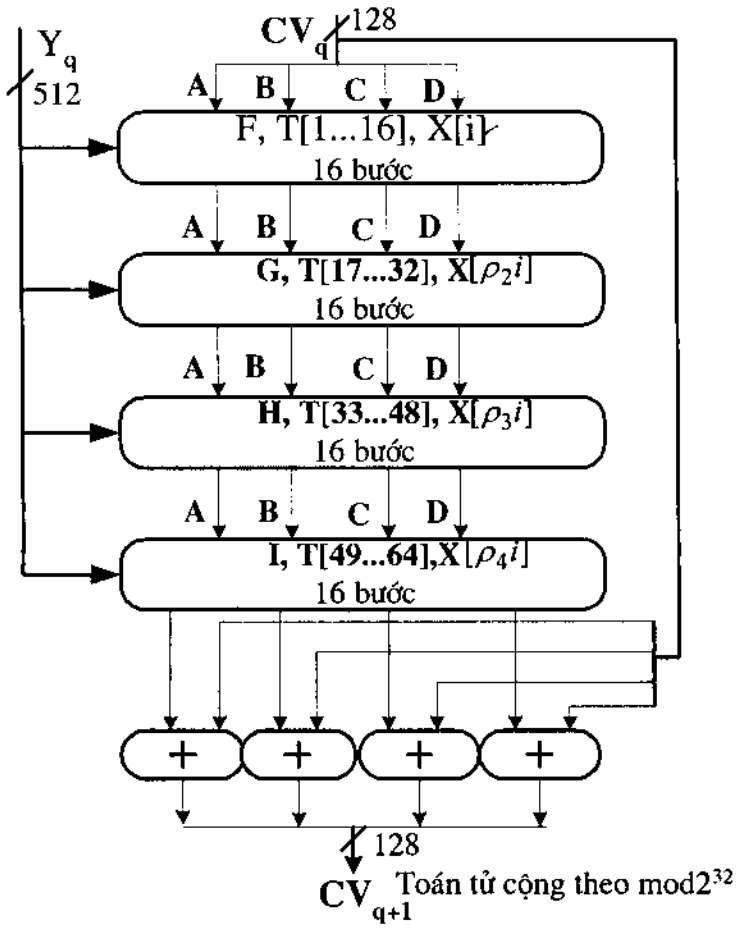
Từ A: 01 23 45 67; (mỗi cặp số trong hệ 16 có giá trị 1byte trong hệ 2)

Từ B: 89 AB CD EF;

Từ C: FE DC BA 89;

Từ D: 76 54 32 10.

♦**Bước 4: Gia công tin tức thành các khối 512bit (mỗi khối theo 16 từ).** Trái tim của thuật toán là hàm nén, bao gồm bốn "công đoạn" gia công. Trên hình 4.13. modul này được ký hiệu là HMD5, còn logic làm việc của modul được chỉ ra trên hình 4.14.



Hình 4.14. Gia công một khối 512 bit trong MD5 (hàm nén MD5).

Tất cả bốn công đoạn đều có cấu trúc tương tự nhau, nhưng trong mỗi công đoạn lại sử dụng hàm logic nguyên thủy của mình (trên hình vẽ phù hợp với hàm được ký hiệu là F, G, H và I).

Trong mỗi công đoạn đưa tới đầu vào 512bit chạy (Y_q) và 128bit đệm ABCD và sau đó khôi phục lại 128bit đệm. Trong mỗi công đoạn cũng sử dụng 1/4 giá trị của bảng 64 phần tử T [1...64], bao gồm từ các giá trị của hàm sin. Trong bảng T, phần tử thứ i, được ký hiệu T[i], là phần nguyên của số $2^{32}\times$ abs(sin(i)), ở đây i là góc tính theo radian. Rõ ràng rằng abs(sin(i)) có giá trị nằm trong dải từ 0 -1, mỗi phần tử T là một số nguyên, mà có thể trình bày trong dạng 32 bit. Bảng bảo đảm"phân đều" tập mã 32 bit, nó có sứ mệnh "đập tan"sự đều dặn bất kỳ của số liệu dẫn vào. Bảng kê giá trị của T trình bày trong bảng 4.6b).

Giá trị đầu ra của công đoạn thứ bốn được bổ sung thêm giá trị đầu vào của công đoạn thứ nhất (CV_q) và kết quả nhận được CV_{q+1}. Phép cộng được thực hiện theo modul 2^{32} không phụ thuộc đối với mỗi một trong bốn từ đệm với các từ phù hợp của CV_q.

♦**Bước 5: Đầu ra.** Sau khi gia công tất cả L khối 512 bit, ở đầu ra của giai đoạn gia công thứ L nhận được 128 bit profil tin tức.

Bảng 4.6. Các phần tử khóa MD5

a) Các giá trị của hàm lôgic

b	c	d	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

b) Bảng T, được tạo ra trên cơ sở các giá trị của hàm sine

T[1] = D76AA478 T[17] = F61E2562 T[33] = FFFA3942 T[49] = F4292244
T[2] = E8C7B756 T[18] = C040B340 T[34] = 8771F681 T[50] = 432AF97
T[3] = 242070DB T[19] = 265E5A51 T[35] = 699D6122 T[51] = AB9423A7
T[4] = C1BDCEE T[20] = E9B6C7AA T[36] = FDE5380C T[52] = FC93A039
T[5] = F57C0AF T[21] = D62F105D T[37] = A4BEEA44 T[53] = 655B59C3
T[6] = 4787C62A T[22] = 02441453 T[38] = 4BDEFCA9 T[54] = 8F0CCC92
T[7] = A8304613 T[23] = D8A1E681 T[39] = F6BB4B60 T[55] = FFEF47D
T[8] = FD469501 T[24] = E7D3FBC8 T[40] = BEBFBC70 T[56] = 85845DD1
T[9] = 698098D8 T[25] = 21E1CDE6 T[41] = 289B7EC6 T[57] = 6FA87E4F
T[10] = 8B44F7AF T[26] = C33707D6 T[42] = EAA127FA T[58] = FE2CE6E0
T[11] = FFFF5BB1 T[27] = F4D50D87 T[43] = D4EF3085 T[59] = A3014314
T[12] = 895CD7BE T[28] = 455A14ED T[44] = 04881D05 T[60] = 4E0811A1
T[13] = 6B901122 T[29] = A9E3E905 T[45] = D9D4D039 T[61] = F7537E82
T[14] = FD987193 T[30] = FCEFA3F8 T[46] = E6DB99E5 T[62] = BD3AF235
T[15] = A679438E T[31] = 676F02D9 T[47] = 1FA27CF8 T[63] = 2AD7D2BB
T[16] = 49B40821 T[32] = 8D2A4C8A T[48] = C4AC5665 T[64] = EB86D391

Theo toàn bộ hoạt động của thuật toán MD5 có thể miêu tả bởi các công thức sau:

$CV_q = IV$

$CV_{q+1} = SUM_{32}(CV_q, RF_I[Y_q, RF_H[Y_q, RF_F[Y_q, CV_q]]])$

$MD = CV_L$.

Ở đây:

IV = giá trị đệm ban đầu ABCD, được cho ở bước 3;

Y_q = khối số liệu 512 bit thứ q;

L = số lượng khối số liệu chung (với sự tính toán các bit bổ sung và trường ghi độ dài tin tức);

RF_X = hàm công đoạn, khi sử dụng hàm lôgic nguyên thủy x;

MD = giá trị đầu ra của profil tin tức;

SUM_{32} = tổng theo modul2^{32}, được thực hiện riêng biệt đối với mỗi một từ của cặp các số liệu dẫn vào.

5.2. Hàm nén MD5

Chúng ta hãy khảo sát chi tiết logic của mỗi một công đoạn từ bốn công đoạn gia công khối 512 bit. Mỗi một công đoạn được hình thành từ sự nối tiếp của 16 toán tử với đệm ABCD đã cho.

Mỗi một toán tử như vậy có dạng:

$a \leftarrow b + ((a + g(b, c, d) + X[k] + T[i] <<< s)$,

ở đây:

a, b, c, d - bốn từ đệm trong sự xác định trình tự, sẽ thay đổi trong quá trình gia công;

g - một từ các hàm nguyên thủy F, G, H, I;

<<< s - dịch vòng trái của kết quả 32 bit đi s bit;

T(i) - từ 32 bit thứ i trong ma trận T;

X(k) - từ 32 bit thứ k trong khối tin tức 512 bit thứ q, nghĩa là $M[q \times 16 + k]$;

+ - ký hiệu trong phép cộng theo mod2^{32}.

Cấu trúc của toán tử thực hiện chỉ ra trên hình 4.15. Trình tự, trong đó sử dụng bốn từ (a, b, c, d) bảo đảm phép dịch vòng trái đi một từ, sau mỗi bước của toán tử.

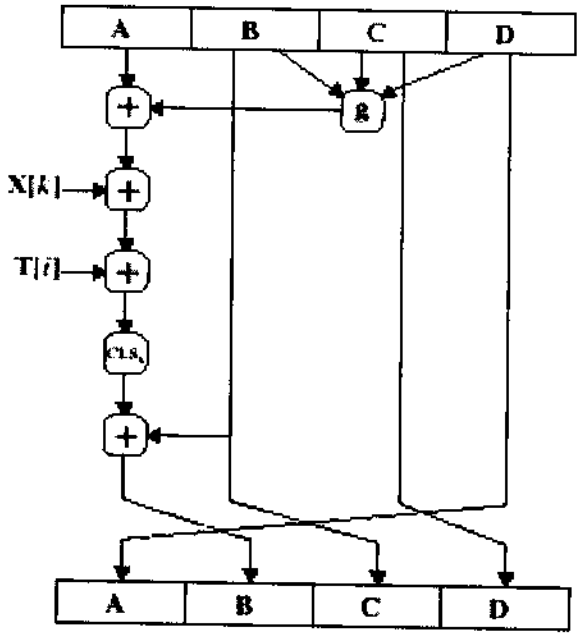
Bảng 4.7. Các hàm nguyên thủy và giá trị của chúng.

Công đoạn	Hàm nguyên thủy g	g(b, c, d)
1	F(b, c, d)	$(b \wedge c) \vee (\bar{b} \wedge d)$
2	G(b, c, d)	$(b \wedge c) \vee (c \wedge \bar{d})$
3	H(b, c, d)	$b \oplus c \oplus d$
4	I(b, c, d)	$c \oplus (b \vee \bar{d})$

Trong mỗi một từ bốn công đoạn của thuật toán sẽ sử dụng hàm logic nguyên thủy của mình. Hàm nguyên thủy như vậy nhận ở đầu vào ba từ 32 bit và đưa ra ở đầu ra một từ 32 bit. Mỗi một hàm như vậy thực hiện hàng loạt toán tử logic theo bit, nghĩa là bit thứ n-i của giá trị đầu ra là hàm của bit thứ n-x của ba giá trị đầu vào. Các hàm nguyên thủy có dạng như bảng 4.7.

Ở đây các toán tử logic AND, OR, NOT, XOR được kí hiệu $\wedge$, $\vee$, - , $\oplus$, phù hợp. Hàm F là biểu thức có điều kiện: nếu b nhân logic c, khác d. Hàm G cũng như vậy, có điều kiện: nếu d

nhân logic b, khác c. Hàm H cho bit chẵn. Giá trị của tất cả bốn hàm được trình bày trong bảng 4.6a).



Hình 4.15. Toán tử cơ sở MD5 (một bước của công đoạn).

Trên hình 4.16. xây dựng trên cơ sở của các số liệu từ RFC 1321, trình bày thuật toán của bước 4. Sẽ gia công các khối của các từ 32 bit X[0…15] bao gồm những giá trị chạy của khối dẫn vào 512 bit. Trong khuôn khổ từng công đoạn, mỗi một trong số 16 từ X[0...15] được ứng dụng chỉ một lần, đối với một bước, còn trình tự trong đó các từ này được sử dụng là thay đổi từ công đoạn này tới công đoạn khác. Trong công đoạn thứ nhất các từ được hình thành trong trình tự khởi đầu của chúng. Trong công đoạn thứ 2 đến thứ 4, người ta định nghĩa các phép hoán vị sau:

$\rho_2(i) = (1 + 5i) \bmod 16;$

$\rho_3(i) = (5 + 3i) \bmod 16;$

$\rho_4(i) = 7i \bmod 16.$

Mỗi một trong số 64 phần tử T có độ dài bằng một từ 32 bit, được sử dụng chỉ một lần đối với một bước của một công đoạn. Cần chú ý rằng trong kết quả thực hiện của một bước của công đoạn được phục hồi lại chỉ 4 byte đệm ABCD. Tiếp tục, trong khoảng của công đoạn, mỗi byte đệm được phục hồi lại bốn lần và còn một lần ở cuối, khi nhận được giá trị của khối đầu ra cuối cùng. Nhận xét cuối cùng, trong mỗi công đoạn người ta sử dụng liên tiếp bốn dịch vòng trái với các giá trị dịch khác nhau. Mục đích của tất cả các phức tạp này là nhằnm đảm bảo không có khả năng thực tiễn tìm ra xung đột (tức là tìm được một cặp của khối 512 bit, mà sẽ có cùng đầu ra).

```
/*Gia công tất cả các khối theo 16từ (512bit)*/
        For q = 0 to (N/16)-1 do
        /* Copy khối q vào X*/
        For j = 0 to 15 do
        Set X[j] to M[q*16+j].
    end /* kết thúc vòng lặp theo j */
/*Lưu A vào AA, B vào BB, C vào CC, D vào
DD */        AA = A; CC = C
             BB = B;  DD = D
        /* Công đoạn 1 */
/* Lấy [abcd k s i] để ký hiệu toán tử
a = b + (( a + F(b,c,d) + X[k] +T[i]<<<s)
 thực hiện theo 16 toán tử sau*/
 [ABCD   0   7   1]
 [DABC   1   12  2]
 [CDAB   2   17  3]
 [BCDA   3   22  4]
 [ABCD   4   7   5]
 [DABC   5   12  6]
 [CDAB   6   17  7]
 [BCDA   7   22  8]
 [ABCD   8   7   9]
 [DABC   9   12  10]
 [CDAB   10  17  11]
 [BCDA   11  22  12]
 [ABCD   12  7   13]
 [DABC   13  12  14]
 [CDAB   14  17  15]
 [BCDA   15  22  16]

        /* Công đoạn 2 */
/* Lấy [abcd k s i] để ký hiệu toán tử
a = b + (( a + G(b,c,d) + X[k] +T[i]<<<s)
 thực hiện theo 16 toán tử sau*/
 [ABCD   1   5   17]
 [DABC   6   9   18]
 [CDAB   11  14  19]
 [BCDA   0   20  20]
 [ABCD   5   5   21]
 [DABC   10  9   22]
 [CDAB   15  14  23]
 [BCDA   4   20  24]
 [ABCD   9   5   25]
 [DABC   14  9   26]
 [CDAB   3   14  27]
 [BCDA   8   20  28]
 [ABCD   13  5   29]
 [DABC   2   9   30]
 [CDAB   7   14  31]
 [BCDA   12  20  32]
        /* Công đoạn 3 */
/* Lấy [abcd k s i] để ký hiệu toán tử
a = b + (( a + H(b,c,d) + X[k] +T[i]<<<s)
 thực hiện theo 16 toán tử sau*/
 [ABCD   5   4   33]
 [DABC   8   11  34]
 [CDAB   11  16  35]
 [BCDA   14  23  36]
 [ABCD   1   4   37]
 [DABC   4   11  38]
 [CDAB   7   16  39]
 [BCDA   10  23  40]
 [ABCD   13  4   41]
 [DABC   0   11  42]
 [CDAB   3   16  43]
 [BCDA   6   23  44]
 [ABCD   9   4   45]
 [DABC   12  11  46]
 [CDAB   15  16  47]
 [BCDA   2   23  48]
        /* Công đoạn 4 */
/* Lấy [abcd k s i] để ký hiệu toán tử
a = b + (( a + I(b,c,d) + X[k] +T[i]<<<s)
 thực hiện theo 16 toán tử sau*/
 [ABCD   0   6   49]
 [DABC   7   10  50]
 [CDAB   14  15  51]
 [BCDA   5   21  52]
 [ABCD   12  6   53]
 [DABC   3   10  54]
 [CDAB   10  15  55]
 [BCDA   1   21  56]
 [ABCD   8   6   57]
 [DABC   15  10  58]
 [CDAB   6   15  59]
 [BCDA   13  21  60]
 [ABCD   4   6   61]
 [DABC   11  10  62]
 [CDAB   2   15  63]
 [BCDA   9   21  64]
/* Bổ sung vào các giá trị chạy của mỗi ghi
dịch các giá trị, mà các ghi dịch đã nhận được
chúng trước khi bắt đầu gia công khối*/
A = A +AA
B = B + BB
C = C + CC
D = D + DD
/*Kết thúc vòng lặp theo q*/
```

Hình 4.16.Thuật toán cơ sở phục hồi MD5 (RFC 1321).

5.3. MD4

Thuật toán MD4 là phiên bản trước của MD5 và cũng được gia công bởi Ron Rivest. Thuật toán bày được xuất bản trong tài liệu RFC vào tháng 10 năm 1990. Không lâu sau, một phiên bản chỉnh sửa đã được xuất bản trong tài liệu RFC 1320 vào tháng hai năm 1992 và cũng thời gian này xuất bản MD5 .

Bây giờ chúng ta khảo sơ qua về MD4, bởi vì khi gia công cả MD5 và MD4 có cùng một mục đích, được hình thành từ bài báo của Ron Rivest. Các mục đích này có liên quan tới các điều sau đây:

♦**An toàn**: Các đòi hỏi thông thường đối với mã hàm băm là không có khả năng tính để tìm ra hai tin tức mà có cùng một profil tin tức. Trong thời gian xuất bản MD4 sự an toàn đạt được đối với các cuộc tấn công kiểu chọn tất cả các phương án, điều này đạt được bằng cách chọn độ dài của profin tin tức. Rivest cũng còn băn khoăn về sự phức tạp của MD4 có được an toàn đối với các cuộc tấn công thám mã.

♦**Tốc độ**: Thuật toán cần phải chấp nhận để bố trí trong những ứng dụng khác nhau, vì vậy cần phải thực hiện đạt tốc độ nhanh. Trong trường hợp riêng, thấy rằng thuật toán cần phải thực hiện nhanh trong các hệ thống với kiến trúc 32 bit. Vì vậy, thuật toán được xây dựng trên những toán tử nguyên thủy đơn giản với các từ 32 bit.

♦**Đơn giản và gọn nhẹ**: Thuật toán cần phải đơn giản theo quan điểm miêu tả nó và khi chương trình hoá cũng không đòi hỏi tạo nên những chương trình lớn và các bảng chuyển vị lớn. Các điều kiện này không chỉ có lợi thế rõ ràng với quan điểm chương trình hóa mà còn mong muốn trong kế hoạch an toàn, bởi vì một thuật toán đơn giản có khả năng bỏ qua phép thám mã về độ bền.

♦ **Kiến trúc ưa chuộng với trình tự trực tiếp của các byte nghiên cứu**: Kiến trúc của một vài vi xử lý (chẳng hạn, họ vi xử lý intel 80xxx và pentium) dự định lưu giữ những byte của từ có giá trị nhỏ nhất vào vị trí địa chỉ sinh sau (cái được gọi là trình tự trực tiếp của byte). Kiến trúc của hai loại vi xử lý (chẳng hạn, SUN Sparcstation) dựa trên sự lưu giữ trong vị trí địa chỉ sinh sau cho các byte của từ giá trị lớn nhất. Sự khác nhau này được thể hiện nếu khảo sát tin tức như là một từ 32 bit liên tiếp, bởi vì đối với trong số kiến trúc trên dẫn đến thay đổi trình tự nghiên cứu của byte trong mỗi từ được gia công. Đối với sự diễn giải tin tức trong dạng từ 32 bit liên tiếp, Rivest đã chọn sơ đồ với trình tự trực tiếp của sự nghiên cứu byte. Sự lựa chọn này của Rivest đã dựa trên sự tin tưởng rằng quá trình xử lý với trình tự trực tiếp của sự nghiên cứu byte, nói chung, là nhanh hơn và tốt hơn xét về mặt giảm tải khi gia công.

Mục đích này cũng đã được theo đuổi khi tạo nên MD5. So sánh với MD4 thì thuật toán MD5 có một số điều phức tạp hơn và bởi vậy nó sẽ chạy chậm hơn. Rivest tính toán rằng khi bổ sung

thêm sự phức tạp nhằm nâng cao thêm mức độ an toàn của thuật toán. Sự khác nhau cơ bản giữa hai thuật toán như sau:

1. Trong MD4 thực hiện gia công theo ba công đoạn, mỗi công đoạn gồm 16 bước, còn MD5 thực hiện bốn công đoạn, mỗi công đoạn 16 bước.
2. Trong MD4 ở công đoạn thứ nhất không sử dụng bất kỳ giá trị bổ sung nào. Ở' công đoạn thứ hai, trong mỗi bước bổ sung thêm một giá trị. Trong công đoạn thứ ba trong mỗi bước lại bổ sung thêm một giá trị khác. Trong MD5 trong tất cả 64 bước bổ sung thêm giá trị T[i] tương ứng.
3. Trong MD5 sử dụng bốn hàm logic nguyên thủy, mỗi hàm trong một công đoạn, còn trong MD4 cũng sử dụng ba hàm này, cũng mỗi hàm trong một công đoạn.
4. Trong MD5 trong mỗi một bước được bổ sung vào kết quả chạy kết quả của bước trước. Thí dụ: kết qủa của bước một được lưu giữ trong từ A. Kết quả của bước hai được lưu giữ trong từ D, được hình thành nhờ bổ sung giá trị A do kết quả của phép dịch vòng trái. Cũng như vậy, kết quả của bước ba được lưu giữ trong từ C, được hình thành nhờ bổ sung giá trị D tới kết quả thực hiện toán tử dịch vòng trái. Trong MD4 phép cộng cuối cũng như thế đã không được thực hiện. Rivest cho rằng việc bổ sung thêm toán tử cộng cuối cùng ở mỗi bước, nhằm tăng thêm hiệu ứng thác lũ trong toàn bộ sơ đồ.

5.4. Độ bền của MD5

Thuật toán MD5 có một đặc tính như sau: mỗi bit của mã hàm băm phụ thuộc vào tất cả các bit đưa đến đầu vào. Sự phức tạp của nhiều lần sử dụng các hàm cơ sở (F, G, H, I), kết quả có một sự trộn lẫn đều tất cả, điều đó có nghĩa rằng không có khả năng thực tiễn để chọn ra hai tin tức khác nhau, có cùng mã hàm băm, ngay cả khi hai tin tức đó có cấu trúc tương tự nhau. Trong tài liệu RFC, Rivest đã chỉ ra rằng MD5 là một thuật toán phức tạp, bài toán tìm ra hai tin tức có cùng profil, có độ phức tạp đánh giá theo 2^{64} ; còn bài toán tìm ra tin tức khi cho trước profil có độ phức tạp đến 2^{128} toán tử.

Trong thời gian đến năm 2000 đã không có kết quả đáng kể nào của cuộc tấn công vào MD5. Tuy nhiên, các cuộc tấn công vào cấu trúc bên trong của MD5 có thể theo các hướng sau:

1. Sử dụng phương pháp thám mã vi sai, Berson đã chỉ ra rằng đối với một công đoạn MD5 có khả năng sau một thời gian hợp lý tìm được hai tin tức, phát sinh cùng một profil. Kết quả này đã được thể hiện đối với một công đoạn của thuật toán. Chúng ta có quyền chờ đợi kết quả tương tự đối với cả bốn công đoạn của thuật toán MD5 ;
2. Boer và Bosslaers đã chỉ ra rằng có thể tìm được một khối tin tức X và hai giá trị liên kết thay đổi phù hợp để phát sinh cùng một trạng thái đầu ra. Điều đó có nghĩa rằng khi thực hiện MD5 với khối 512 bit riêng biệt, cho ở đầu ra có cùng một giá trị, đối với hai giá trị đầu vào khác nhau trong giá trị đệm ABCD. Tình trạng như vậy được gọi là

giả va chạm. Trong thời gian hiện tại chưa có khả năng tấn công thành công MD5 theo kiểu đó;

3. Một chiến lược tấn công MD5 đáng quan tâm hơn cả, được đề xuất bởi Dobbertin. Kỹ thuật này cho phép phát sinh va chạm đối với hàm nén MD5, nghĩa là cách tấn công này được xây dựng trên cơ sở khảo sát hoạt động MD5 đối với từng khối 512 bit riêng biệt và tìm ra một khối khác cho cùng giá trị 128 bit đầu ra. Trong thời gian đến năm 2000 vẫn chưa biết được khả năng tạo ra tin tức đầy đủ cũng như giá trị khởi đầu (IV) của MD5 một cách phù hợp. Tuy nhiên, cách tấn công có vẻ dẫn đến kết quả tốt hơn.

Như vậy, theo quan điểm thám mã MD5 cho đến nay cần phải cho rằng để bị tổn thương. Ngoài ra, với khả năng thử chọn tất cả các phương án để tấn công MD5, đòi hỏi công sức tỉ lệ với 2^{64}. Để tăng thêm độ an toàn của thuật toán hàm băm, có thể chọn giá trị profil dài hơn, có hai thuật toán hàm băm khác là SHA-1 và RIPEMD - 160 đã đi theo phương hướng này.

CHƯƠNG 5

TÍN HIỆU VÀ PHỔ

Chương 5 cung cấp cho chúng ta một công cụ quan trọng để phân tích tín hiệu, đó là khi biết *biểu diễn của tín hiệu trong miền thời gian*, thường ký hiệu là s(t) (hoặc w(t), hoặc x(t),..., tùy hoàn cảnh), có thể xác định được *biểu diễn của tín hiệu trong miền tần số*, thường ký hiệu là S(f) (hoặc W(f), hoặc X(f),...) tương ứng, được gọi là *phổ tần số* của tín hiệu. Những kiến thức thu nhận được từ chương 5 được sử dụng làm cơ sở cho hàng loạt các vấn đề về xử lý tín hiệu trong thực tiễn kỹ thuật. Trong chương này, chúng ta sẽ lần lượt nghiên cứu các vấn đề sau:

- Phân tích tín hiệu thành chuỗi Fourier;
- Biến đổi Fourier cho các tín hiệu tồn tại trong khoảng thời gian hữu hạn. Cùng với chúng, chúng ta sẽ nghiên cứu sơ lược về biến đổi Fourier rời rạc, biến đổi cosine rời rạc ;
- Biến đổi Wierner- Khintchine cho các tín hiệu ngẫu nhiên dừng.

Trước hết chúng ta nghiên cứu sơ lược về phân loại tín hiệu trong các hệ thống truyền tin.

1. PHÂN LOẠI TÍN HIỆU

Tin tức truyền lan trong các hệ thống truyền tin được biểu thị dưới dạng các tín hiệu điện (dòng điện, điện áp), hoặc sóng điện từ.

Định nghĩa: tín hiệu thực là một đại lượng vật lý biến thiên theo thời gian t, chứa đựng thông tin (tin tức). Tín hiệu có chức năng truyền thông tin (tin tức) từ điểm nguồn đến điểm người sử dụng.

Khi nói tín hiệu thực là một đại lượng vật lý, chúng ta có các hàm ý sau:

- Tín hiệu chỉ tồn tại trong khoảng thời gian hữu hạn;
- Tín hiệu có năng lượng hữu hạn, có phổ tần số hữu hạn;
- Dạng sóng của tín hiệu không tồn tại các dạng vô hạn (theo nghĩa hoặc các giá trị đỉnh của tín hiệu, hoặc đạo hàm theo thời gian của dạng sóng tín hiệu phải hữu hạn).

Điều kiện thứ nhất và thứ hai được chấp nhận một cách tự nhiên, vì mọi nguồn sinh tín hiệu thực đều tự nhiên có các hạn chế này. Còn điều kiện cuối cùng là hậu quả của hai điều kiện trên, khi xem xét theo quan điểm toán học.

Các mô hình toán học của tín hiệu mà chúng ta khảo sát sau này thường vi phạm một vài điều qui định ở trên, vì lý do để đơn giản trong phân tích chúng. Vì thế các kết quả nhận được chỉ là gần đúng với thực tiễn.

Khi nghiên cứu tín hiệu thực, hoặc mô hình toán học của tín hiệu thực, chúng ta thường phân loại chúng thành từng cặp sau đây:

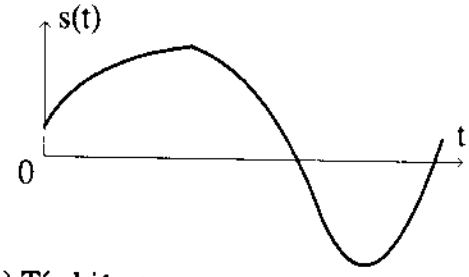
a). Tín hiệu tương tự và tín hiệu số

- *Tín hiệu tương tự (analogue signal)* là tín hiệu mà tập các giá trị xác định chúng là vô hạn không đếm được. Tín hiệu tương tự là một hàm liên tục và hữu hạn theo thời gian, hoặc liên tục từng đoạn. Đồ thị biểu diễn tín hiệu tương tự theo thời gian có dạng chung như hình 5.1a). Các hệ thống truyền tin *chỉ gia công và truyền tín hiệu tương tự* được gọi là hệ thống truyền tin tương tự (*analogue communication system*). Đây là các hệ thống truyền tin được phát triển và sử dụng sớm nhất trong các hệ thống truyền tin điện tử. Ngày nay chúng vẫn tiếp tục được khai thác, sử dụng.

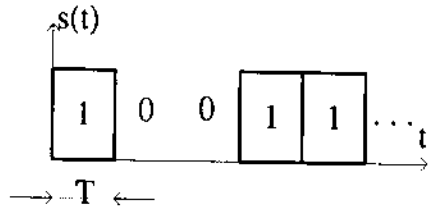
- *Tín hiệu số (digital signal)* là tín hiệu, mà tập các giá trị xác định chúng chỉ là hai, được gọi là mức logic 1 và mức logic 0. Giá trị cụ thể của mức logic 1 và mức logic 0 tùy thuộc vào chuẩn quy định chúng, chẳng hạn, theo chuẩn TTL (Tranzito Tranzito Logic) thì mức logic 1 có giá trị là +5 vôn, còn mức logic 0 có giá trị là 0 vôn, còn ở cổng đầu ra nối tiếp của các máy tính cá nhân thì mức logic 1 có giá trị là −10 vôn, còn mức logic 0 có giá trị là +10 vôn. Đồ thị biểu diễn tín hiệu số (mô hình toán học của tín hiệu số thực) theo thời gian có dạng chung như hình 5.1b). Hơn nữa, tín hiệu số có thể biểu diễn ở dạng đơn cực, như hình 5.1b1), hoặc dạng lưỡng cực, như hình 5.1b2).

Các hệ thống truyền tin *có gia công và truyền tín hiệu số* thì được gọi là hệ thống truyền tin số (digital communication system). Tuy nhiên, bản chất của các nguồn tin trong tự nhiên (như hình ảnh, tiếng nói, âm nhạc) đều ở dạng tương tự, nguồn tin số là do con người tạo ra, và sau phép điều chế mọi tín hiệu đều là tín hiệu tương tự. Vì vậy, mọi hệ thống truyền tin được gọi là hệ thống truyền tin số thì ở các khối đầu vào và đầu ra đều thực hiện gia công, truyền tín hiệu tương tự. Có thể trong một thiết bị đầu cuối như máy vi tính, chúng ta chỉ thấy có tín hiệu số, tuy nhiên, để truyền đi xa, tín hiệu số đều phải điều chế, tức là phép biến đổi đặc biệt từ tín hiệu số thành tín hiệu tương tự.

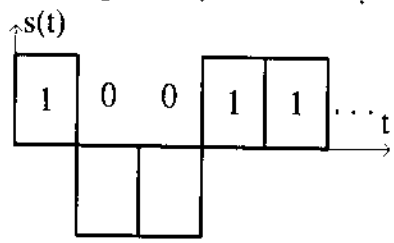
Hệ thống truyền tin số có ưu điểm đặc biệt so với hệ thống truyền tin tương tự, nhờ phép *tái sinh các ký hiệu (symbols)* tại các nút mạng trung gian và đầu cuối. Ở mỗi lần tái sinh các ký hiệu, nhiễu được loại bỏ hoàn toàn (ở các hệ thống truyền tin tương tự năng lượng của nhiễu được tích lũy). Vì vậy, với hệ thống truyền tin số năng lượng của nhiễu không có khả năng tích lũy, cho nên tỉ số (tín hiệu/nhiễu) luôn có điều kiện cải thiện. Hơn nữa, ở hệ thống truyền tin số còn có khả năng sử dụng mã chống nhiễu, và nhiều hình thức gia công đặc thù khác (chẳng hạn, các hình thức điều chế số), cho nên chất lượng tín hiệu luôn vượt trội hơn so với hệ thống truyền tin tương tự.



a).Tín hiệu tương tự



b1). Dạng đơn cực của tín hiệu số



b2). Dạng lưỡng cực của tín hiệu số

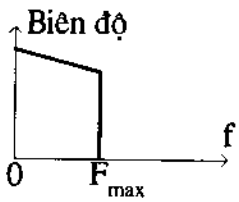
Hình 5.1. Đồ thị biểu diễn tín hiệu tương tự a) và tín hiệu số b).

b). Tín hiệu băng cơ sở và tín hiệu băng thông (dải thông)

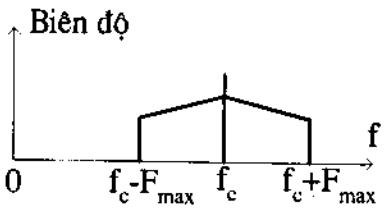
- *Tín hiệu băng cơ sở (base band signal)* là tín hiệu mà phổ tần số của nó chỉ tập trung chủ yếu quanh tần số bằng 0, và không đáng kể tại các tần số còn lại, tức là khoảng phổ tần số của nó có dạng $0 \le f = F_{max}$, trong đó F_{max} là một giá trị hữu hạn nào đó. Chẳng hạn, quy ước khoảng phổ tín hiệu thoại: f = (300 ÷ 3400) Hz; khoảng phổ tín hiệu audio: f =(20÷20.000) Hz ; khoảng phổ tín hiệu hình ảnh: f =(0 ÷ 4,5) MHz . Hình ảnh điển hình về phổ tần số của tín hiệu băng cơ sở được biểu diễn trên hình 5.2a).

Tín hiệu băng cơ sở có thể là tín hiệu tương tự, có thể là tín hiệu số.

- *Tín hiệu dải thông (bandpass signal)* là tín hiệu mà phổ tần số của nó tập trung quanh một tần số sóng mang f_c nào đó, và không đáng kể tại các tần số còn lại,. Dạng phổ tần số điển hình của tín hiệu dải thông được biểu diễn trên hình 5.2b).



a).Phổ biên độ- tần số của tín hiệu băng cơ sở



b).Phổ biên độ- tần số của tín hiệu dải thông

Hình 5.2. Dạng phổ biên độ – a) tần số điển hình của tín hiệu băng cơ sở b) của tín hiệu dải thông.

Tín hiệu băng cơ sở sau khi điều chế sẽ trở thành tín hiệu dải thông (vấn đề điều chế tín hiệu sẽ nghiên cứu trong chương 6). Cần chú ý rằng tín hiệu dải thông luôn luôn là tín hiệu tương tự.

c). Tín hiệu tiền định và tín hiệu ngẫu nhiên

- *Tín hiệu tiền định* là tín hiệu mà quy luật biến thiên của nó theo thời gian được định trước (biết trước). Các mô hình toán học của tín hiệu mà chúng ta xem xét sau này thuộc dạng tín hiệu tiền định.

- *Tín hiệu ngẫu nhiên* là tín hiệu mà quy luật biến thiên của nó theo thời gian có đặc tính ngẫu nhiên, không được định trước (không biết trước).

Khi nghiên cứu tín hiệu ngẫu nhiên, thường ký hiệu là $X_\omega(t)$, hoặc đơn giản X(t), trong đó $\omega \in \Omega$ là tập của các biến thử, còn $t \in T$ là tập của các biến thời gian.

Khi cố định biến thử, tức cho $\omega = \omega_1$ - một biến thử cụ thể nào đó, thì $X_{\omega 1}(t) \to x(t)$, trong đó x(t) gọi là một thể hiện, hoặc một quỹ đạo của X(t).

Khi cố định biến thời gian, tức cho $t = t_1$, thì X(t) → X(t_1), thì X(t_1) là một biến ngẫu nhiên (BNN). Lý thuyết về tín hiệu ngẫu nhiên là phức tạp, độc giả cần nghiên cứu thêm ở các tài liệu chuyên khảo về quá trình ngẫu nhiên (QTNN).

Theo lý thuyết thông tin (xem chương 1) thì thông tin được sinh ra theo bản chất ngẫu nhiên, mà tín hiệu là đại lượng chứa đựng thông tin, vì thế mọi tín hiệu thực (tín hiệu của các hệ thống truyền tin) đều là tín hiệu ngẫu nhiên, nhưng cũng phải nhớ rằng tham số nào của tín hiệu chứa đựng thông tin, thì tham số đó mang bản chất ngẫu nhiên, còn các tham số khác ở dạng tiền định, đặc biệt tần số sóng mang luôn luôn là đại lượng tiền định.

d). Tín hiệu có chu kỳ và tín hiệu không chu kỳ

- *Tín hiệu có chu kỳ* (còn gọi là tín hiệu tuần hoàn) là tín hiệu mà giá trị biên độ của nó lặp lại sau những khoảng thời gian cố định. Khoảng thời gian ngắn nhất để giá trị của biên độ tín hiệu lặp lại được gọi là chu kỳ T. Với tín hiệu có chu kỳ, thường viết: $s(t) = s(t + nT)$, trong đó n = 0, ±1, ±2,...

Tín hiệu có chu kỳ là *mô hình toán học của tín hiệu thực*, nó có năng lượng lớn vô cùng, nhưng công suất trung bình $\overline{P}$ trong toàn khoảng thời gian vô hạn cũng bằng công suất trung bình trong một chu kỳ và là hữu hạn, vì:

$$\overline{P} = \lim_{T_a \to \infty} \frac{1}{T_a} \int_{-\infty}^{+\infty} s^2(t)dt = \frac{1}{T} \int_{-T/2}^{+T/2} s^2(t)dt < \infty \qquad (5.1)$$

Vì thế tín hiệu có chu kỳ còn gọi là tín hiệu công suất.

- *Tín hiệu không chu kỳ*, chúng ta hạn chế chỉ đề cập tới dạng có khoảng thời gian tồn tại hữu hạn, còn gọi là tín hiệu năng lượng. Vì năng lượng của tín hiệu có khoảng thời gian tồn tại hữu hạn là xác định (tất nhiên các giá trị đỉnh của các tín hiệu loại này cũng phải hữu hạn), bởi vì:

$$E = \int_{ta}^{tb} s^2(t)dt < \infty \tag{5.2}$$

trong đó *(ta, tb)* – khoảng thời gian tồn tại của tín hiệu.

Chúng ta thấy rằng một tín hiệu được phân loại là một trong hai kiểu này thì nó không thể là kiểu kia. Nghĩa là, nếu s(t) có năng lượng hữu hạn, thì công suất trung bình trên khoảng thời gian vô hạn phải bằng không, còn nếu có công suất trung bình trên khoảng thời gian vô hạn là hữu hạn, thì năng lượng là vô hạn.

2. PHÂN TÍCH TÍN HIỆU THÀNH CHUỖI FOURIER

Chuỗi Fourier là chuỗi các hàm điều hòa (hàm sine, hàm cosine) và là một trường hợp *cụ thể điển hình* của chuỗi trực giao. Trước khi xem xét các ứng dụng chuỗi Fourier, chúng ta hãy tìm hiểu sơ bộ về các hàm trực giao và các chuỗi trực giao (độc giả muốn tìm hiểu kỹ hơn về các vấn đề này có thể tham khảo các tài liệu khác, chẳng hạn, [11]).

2.1. Các hàm trực giao

Định nghĩa: Các hàm $\psi_n(t)$ và $\psi_m(t)$ được gọi là *các hàm trực giao* với nhau trong khoảng a < t < b, nếu chúng thỏa mãn điều kiện:

$$\int_a^b \psi_n(t)\psi_m^*(t)dt = \begin{Bmatrix} 0 & n \neq m \\ K_n & n = m \end{Bmatrix} = K_n\delta_m \tag{5.3}$$

(dấu * chỉ toán tử liên hợp phức)

trong đó : $\delta_m = \begin{Bmatrix} 0 & n \neq m \\ 1 & n = m \end{Bmatrix}$ - gọi là hàm delta Kronecker

Nếu tất cả các $K_n = 1$ thì các hàm $\psi_n(t)$ và $\psi_m(t)$ được gọi là *trực chuẩn*.

Công thức (5.3) dùng để kiểm tra cặp hàm xem chúng có trực giao với nhau hay không. Sau đây chúng xét các thí dụ thường gặp nhất:

Thí dụ: xét các cặp hàm sinnωt, sinmωt và cặp hàm cosnωt, cosmωt. Chúng ta sẽ kiểm tra điều kiện trực giao của chúng trong khoảng thời gian một chu kỳ, tức là khoảng b = a + T, trong đó T= 2π/ω, với n, m là các số nguyên. Hơn nữa, các hàm này là hàm thực (không phải hàm phức).

a). Chúng ta cần chứng minh khi n ≠ m, thì $\int_a^{a+T} \sin n\omega t \sin m\omega t\, dt = 0$.

Thực vậy, chúng ta có: sinnωt × sinmωt = $\frac{1}{2}$ [cos(n-m)ωt – cos(n+m)ωt],

$$\int_a^{a+T} \frac{1}{2}[\cos(n-m)\omega t - \cos(n+m)\omega t]dt = \frac{1}{2}\left[\frac{\sin(n-m)\omega t}{(n-m)\omega}\Bigg|_a^{a+T} - \frac{\sin(n+m)\omega t}{(n+m)\omega}\Bigg|_a^{a+T}\right]$$

$$= \frac{1}{2}\left\{\left[\frac{\sin(n-m)\omega(a+T) - \sin(n-m)\omega a}{(n-m)\omega}\right] - \left[\frac{\sin(n+m)\omega(a+T) - \sin(n+m)\omega a}{(n+m)\omega}\right]\right\}$$

Biến đổi các biểu thức trong [.], ta được:

$$\sin(n \pm m)\omega(a+T) = [\sin(n \pm m)\omega a \times \cos(n \pm m)2\pi] + [\cos(n \pm m)\omega a \times \sin(n \pm m)2\pi]$$

vì $cos(n\pm m)2\pi = 1$, còn $sin(n\pm m)2\pi = 0$, cho nên các số hạng trong [.] có dạng: $[\sin(n \pm m)\omega a - \sin(n \pm m)\omega a] = 0$. Cuối cùng chúng ta đã chứng minh được điều mong muốn.

Khi $n = m \neq 0$, chúng ta có: sinnωt × sinmωt = $\frac{1}{2}$ [1-cos2nωt]. Tiếp tục chúng ta xác định được:

$$\int_a^{a+T} \frac{1}{2}[1 - \cos 2n\omega t]dt = \frac{1}{2}t\Big|_a^{a+T} - \frac{1}{2}\frac{\sin 2n\omega t}{2n\omega}\Bigg|_a^{a+T} = \frac{1}{2}T - 0 = \frac{1}{2}T$$

Vì

$$\sin 2n\omega t\Bigg|_a^{a+T} = \sin 2n\omega(a+T) - \sin 2n\omega a =$$

$$\sin 2n\omega a \times \cos 2n2\pi - \cos 2n\omega a \times \sin 2n2\pi$$

$$-\sin 2n\omega a = 0 \text{ (trong đó } \cos 2n2\pi = 1, \text{còn } \sin 2n2\pi = 0)$$

Chúng ta xét trường hợp đặc biệt, khi n = 0, lúc đó cần tính giới hạn:

$$\lim_{n\to 0}\left[\frac{1}{2}\frac{\sin 2n\omega t}{2n\omega}\right] = \lim_{n\to 0}\left[\frac{\sin 2n\omega a \times \cos 2n2\pi}{2n\omega} - \frac{\cos 2n\omega a \times \sin 2n2\pi}{2n\omega}\right]$$

$$= \lim_{n\to 0}\left[a\frac{\sin 2n\omega a}{2n\omega a} - T\frac{\sin 2n2\pi}{2n2\pi}\right] = \frac{1}{2}(a - T)$$

tức là $K_0 = (T - \frac{1}{2}a)$, còn phụ thuộc cả vào điểm đầu lấy tích phân. Tuy nhiên, trong thực tiễn ta thường chọn điểm đầu lấy tích phân là gốc tọa độ, tức là a ≡ 0, khi đó $K_0 = T$

Một điểm đầu khác để lấy tích phân cũng thường chọn là a = - T/2, khi ấy điểm cuối lấy tích phân b = +T/2. Chúng ta xác định giá trị tích phân, khi $n = m \neq 0$, như sau:

$$\int_{-T/2}^{T/2}\frac{1}{2}[1-\cos 2n\omega t]dt=\frac{1}{2}t\Big|_{-T/2}^{T/2}-\frac{1}{2}\frac{\sin 2n\omega t}{2n\omega}\Bigg|_{-T/2}^{T/2}=\frac{1}{2}T-\frac{1}{2}\frac{\sin n2\pi+\sin n2\pi}{2n\omega}=\frac{1}{2}T$$

Khi $n = m = 0 \rightarrow$ $\int_{-T/2}^{T/2}\frac{1}{2}[1-\cos 2n\omega t]dt=\frac{1}{2}t\Big|_{-T/2}^{T/2}-\lim_{n\to 0}\frac{1}{2}\frac{\sin 2n\omega t}{2n\omega}\Bigg|_{-T/2}^{T/2}=0$

Kết quả chung, chúng ta có:

$$\int_{a}^{a+T}\sin n\omega t\sin m\omega t\,dt=\begin{Bmatrix}0 & n\neq m & \neq 0\\ T & n=m=0 & a=0\\ \frac{1}{2}T & n=m & \neq 0\end{Bmatrix} \tag{5.4}$$

Sử dụng kết quả (5.4), chúng ta thấy khi n = m ≠ 0, có $K_n=\frac{1}{2}T$. Đối với tất cả giá trị nguyên n = m ≠ 0 có $K_n \neq 1$, cho nên các hàm *sinnωt* không trực chuẩn, nhưng trực giao (trừ n = m = 0, khi chọn điểm đầu lấy tích phân a = T/2). Muốn các hàm sinnωt trở thành các hàm trực chuẩn, cần chuẩn hóa biên độ của chúng theo $\sqrt{K_n}=\sqrt{\frac{1}{2}T}$.

b). Đối với cặp hàm cosnωt, cosmωt độc giả có tự chứng minh để nhận được kết quả tương tự:

$$\int_{a}^{a+T}\cos n\omega t\cos m\omega t\,dt=\begin{Bmatrix}0 & n\neq m & \\ T & n=m=0 & a=0,\frac{T}{2}\\ \frac{1}{2}T & n=m\neq 0 & \end{Bmatrix} \tag{5.4a}$$

2.2. Các chuỗi trực giao

Giả sử s(t) biểu diễn một *dạng sóng thực* nào đó, tuân thủ các điều kiện xét ở tiết 1, hoặc một *dạng sóng là mô hình toán học* của một dạng sóng thực, không cần thiết phải tuân thủ đầy đủ các điều kiện của dạng sóng thực, nhưng để các tích phân mà chúng xét sau này hội tụ, nó vẫn cần một điều kiện là các giá trị đỉnh phải hữu hạn, tức là phải thỏa mãn *điều kiện Direchlet* (*không có các gián đoạn loại 2, số lượng gián đoạn loại 1 là hữu hạn*). Chúng ta muốn biểu diễn chuỗi của s(t) trên khoảng a < t < b. Khi đó chúng ta có thể biểu diễn thành chuỗi các hàm trực giao, nhờ sử dụng định lý sau:

Định lý: dạng sóng s(t) có thể được biểu diễn trong khoảng (a, b) bằng chuỗi các hàm trực giao:

$$s(t) = \sum_n a_n \psi_n(t) \tag{5.5}$$

trong đó các hệ số trực giao được xác định bởi:

$$a_n = \frac{1}{K_n} \int_a^b s(t).\psi_n^*(t)dt \tag{5.6}$$

với chỉ số n lấy các giá trị nguyên.

Chứng minh: Giả thử $\psi_n(t)$ là tập trực giao đầy đủ để biểu diễn dạng sóng. Khi đó để (5.5) đúng, chúng ta chỉ cần chứng minh rằng a_n có thể tính bằng (5.6). Thực vậy, chúng ta nhân hai vế của (5.5) với $\psi_m^*(t)$, rồi lấy tích phân trong khoảng (a,b) được:

$$\int_a^b s(t).\psi_m^*(t)dt = \int_a^b \left[\sum_n a_n \psi_n(t)\right] \psi_m^*(t)dt$$

$$= \sum_n a_n \int_a^b \psi_n(t)\psi_m^*(t)dt = \sum_n a_n K_n \delta_m = a_n K_n$$

Chúng ta giải thích kết quả trên như sau: số hạng đầu tiên trong chuỗi các hàm ở trên chỉ nhận giá trị m = n, cho nên số hạng cuối thứ hai cũng chỉ nhận giá trị m = n, vì thế chúng ta có kết quả cuối cùng. Kết quả trên đã chứng minh công thức (5.6) là đúng.

Tập hàm $\psi_n(t)$ là tập trực giao đầy đủ, nếu nó có thể dùng để biểu diễn dạng sóng s(t) với sai số nhỏ tùy ý. Các hàm mũ phức, các hàm điều hòa, các hàm Bessel, các đa thức Legendre, các hàm sinx/x là các tập hàm trực giao đầy đủ.

Khi các hàm $\psi_n(t)$ là thực, chúng ta có công thức (5.6) với dạng mới như sau:

$$a_n = \frac{1}{K_n} \int_a^b s(t).\psi_n(t)dt \tag{5.6a}$$

2.3. Phân tích tín hiệu tuần hoàn thành chuỗi Fourier

Biểu diễn bằng chuỗi các hàm trực giao theo công thức (5.5) có thể sử dụng cho dạng sóng của tín hiệu thực, hoặc dạng sóng mô hình toán học của tín hiệu thực. Tuy nhiên, hay sử dụng nhất cho các dạng sóng có tính tuần hoàn.

Xét tín hiệu tuần hoàn s(t), *thỏa mãn điều kiện Direchlet*. Biểu diễn chuỗi Fourier của s(t) trong khoảng một chu kỳ T, có dạng:

$$s(t) = \sum_{n=0}^{\infty} \left(a_n \cos n\omega t + b_n \sin n\omega t\right) \tag{5.7}$$

trong đó các hệ số của chuỗi trực giao được xác định theo công thức (5.6a) và hệ số K_n được xác định theo công thức (5.4) và (5.4a), với điểm đầu lấy tích phân bằng 0, có dạng:

$$a_n = \begin{cases} \dfrac{1}{T}\displaystyle\int_0^T s(t)dt & n = 0 \\ \dfrac{2}{T}\displaystyle\int_0^T s(t)\cos n\omega t dt & n \geq 1 \end{cases} \tag{5.8a}$$

$$b_n = \frac{2}{T}\int_0^T s(t)\sin n\omega t dt \quad n \geq 1 \tag{5.8b}$$

Tuy nhiên, theo truyền thống chuỗi Fourier của tín hiệu s(t) thường biểu diễn dưới dạng tách riêng hệ số a_0 ra ngoài như sau:

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty}\left(a_n \cos n\omega t + b_n \sin n\omega t\right) = C_0 + \sum_{n=1}^{\infty} C_n \cos\left(n\omega t + \theta_n\right) \tag{5.7a}$$

khi đó các hệ số của chuỗi được xác định:

$$a_n = \frac{2}{T}\int_0^T s(t)\cos n\omega t dt \qquad C_n = \sqrt{a_n^2 + b_n^2}$$

và

$$b_n = \frac{2}{T}\int_0^T s(t)\sin n\omega t dt \qquad \theta_n = arctg\frac{b_n}{a_n} \tag{5.8c}$$

Vì các hàm điều hòa này tuần hoàn với chu kỳ T, nên chuỗi (5.7) tuần hoàn với chu kỳ T, cho nên s(t) cũng tuần hoàn với chu kỳ T, tức là chuỗi biểu diễn s(t) trên toàn bộ trục thực ($-\infty < t < \infty$).

Chúng ta có một vài nhận xét về các kết quả biểu diễn trong các công thức (5.7a) và (5.8c):

- Có thể xem một tín hiệu s(t) tuần hoàn như tổng của vô số các thành phần điều hòa với tần số khác nhau và biên độ khác nhau. Mỗi thành phần được gọi là *một hài*. Hài bậc $n = 0$, có biên độ C_0, còn tần số $f = 0$, gọi là thành phần một chiều. Hài bậc $n = 1$, có biên độ C_1, còn tần số $f_1 = \frac{1}{T}$, gọi là hài cơ bản. Các hài tương ứng với $n \geq 2$, gọi là hài bậc cao. Nói chung, biên độ của các hài bậc càng cao càng giảm dần. Vì thế, nếu cho trước sai số, toán học đã chứng minh chuỗi Fourier có thể cắt cụt đến thành phần thứ n thích hợp, tức là:

$$s(t) \approx \frac{a_0}{2} + \sum_{n=1}^{n}\left(a_n \cos n\omega t + b_n \sin n\omega t\right)$$

- Sự phụ thuộc của C_n - biên độ các hài (với n = 0, 1, 2,...) vào tần số gọi là phổ biên độ - tần số; còn sự phụ thuộc của θ_n - pha đầu các hài, vào tần số gọi là phổ pha - tần số.

- Để giảm bớt khối lượng tính toán các hệ số theo công thức (5.8c), chúng ta có nhận xét sau:

• Nếu s(t) là hàm chẵn của t (hàm chẵn có đặc điểm: nếu thay t → –t, thì hàm không đổi dấu, tức là s(–t) = s(t); đồ thị của hàm chẵn đối xứng qua trục đứng), thì hệ số $b_n = 0$, vì tích của một hàm chẵn với một hàm lẻ là một hàm lẻ, mà tích phân của một hàm lẻ trong một chu kỳ sẽ bằng 0. Vì thế trong trường hợp này chỉ có các hệ số $a_n \neq 0$.

• Nếu s(t) là hàm lẻ của t (hàm lẻ có đặc điểm: nếu thay t → –t, thì hàm đổi dấu, tức là s(–t) = – s(t); đồ thị của hàm lẻ đối xứng qua gốc tọa độ), thì hệ số $a_n = 0$, vì tích của một hàm lẻ với một hàm chẵn là một hàm lẻ, mà tích phân của một hàm lẻ trong một chu kỳ sẽ bằng 0. Vì thế trong trường hợp này chỉ có các hệ số $b_n \neq 0$.

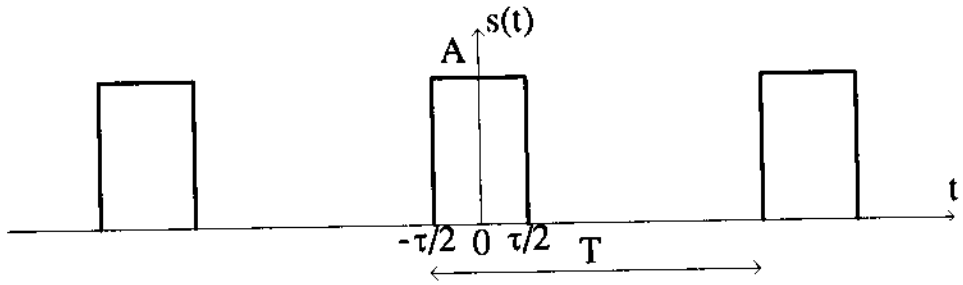
• Đối với các dạng sóng tuần hoàn, trong toán học còn có thêm ba trường hợp nữa, để giảm khối lượng tính các hệ số của chuỗi Fourier, chúng ta sẽ bắt gặp chúng khi giải các bài tập.

Sau đây chúng ta xét một vài thí dụ về phân tích tín hiệu tuần hoàn thành chuỗi Fourier.

Thí dụ : Cho s(t) là chuỗi xung vuông tuần hoàn, có đồ thị như hình 5.3. Các xung có biên độ A (vôn), độ rộng τ (μsec), chu kỳ T (μsec). Xác định chuỗi Fourier của s(t).

Trong chu kỳ đầu tiên, s(t) có dạng:

$$s(t) = \begin{cases} A & -\dfrac{\tau}{2} \le t \le +\dfrac{\tau}{2} \\ 0 & +\dfrac{\tau}{2} < t \le T - \tau \end{cases}$$ (trong trường hợp này, chu kỳ T được bắt đầu từ điểm - τ/2)



Hình 5.3. Đồ thị của chuỗi xung vuông tuần hoàn.

Để xác định chuỗi Fourier, chúng ta sử dụng dạng (5.7a) với các hệ số xác định theo (5.8c) và chú ý s(t) là hàm chẵn nên hệ số $b_n = 0$. Kết quả được:

$$a_0 = 2 \times \frac{2}{T}\int_0^{\tau/2} s(t)dt = \frac{4}{T}\int_0^{\tau/2} Adt = \frac{4At}{T}\Big|_0^{\frac{\tau}{2}} = \frac{2A\tau}{T}$$

$$a_n = 2\times\frac{2}{T}\int_0^{\tau/2} s(t)\cos n\omega t dt = \frac{4}{T}\int_0^{\tau/2} A\cos n\omega t dt = \frac{2AT}{T}\frac{\sin n\omega\tau/2}{n\omega\tau/2}$$

Đưa các hệ số này vào công thức (5.7a) xác định được chuỗi Fourier:

$$s(t) = \frac{A\tau}{T} + \sum_{n=1}^{\infty}(\frac{2A\tau}{T}\frac{\sin n\omega\tau/2}{n\omega\tau/2}\times\cos n\omega t) = C_0 + \sum_{n=1}^{\infty} C_n \times \cos(n\omega t) \qquad (5.9)$$

Khi nghiên cứu các chuỗi Fourier người ta thường khảo sát sự phụ của biên độ các hài vào tần số và vẽ thành đồ thị và gọi là *phổ biên độ – tần số*. Cùng với nó, người ta cũng thường khảo sát sự phụ của pha đầu các hài vào tần số và vẽ thành đồ thị và gọi là *phổ pha – tần số*. Chúng ta khảo sát các sự phụ này, khi căn cứ vào biểu thức (5.9), với các giá trị cụ thể A = 1vôn, τ = 1μsec, T = 5μsec.

Chúng ta đặt: $C_0 = \frac{A\tau}{T}$, gọi là biên độ thành phần một chiều của s(t). Giá trị cho trường hợp cụ thể này là: $C_0 = \frac{1\times 1.10^{-6}}{5.10^{-6}} = \frac{1}{5}$ vôn;

$C_n = \frac{2A\tau}{T}\frac{\sin n\omega\tau/2}{n\omega\tau/2}$, gọi là biên độ hài thứ n của s(t). Với thay thế $\omega = \frac{2\pi}{T}$, chúng ta được: $C_n = \frac{2A\tau}{T}\frac{\sin n\pi/5}{n\pi/5}$:

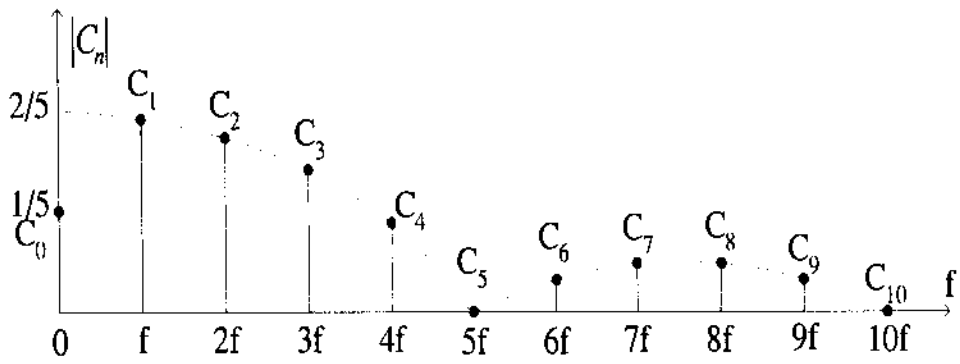
- Với n = 1, chúng ta có: $C_1 = \frac{2}{5}\frac{\sin\pi/5}{\pi/5}$;
- Với n = 2, chúng ta có: $C_2 = \frac{2}{5}\frac{\sin 2\pi/5}{2\pi/5}$;
- Với n = 3, chúng ta có: $C_3 = \frac{2}{5}\frac{\sin 3\pi/5}{3\pi/5}$;
- Với n = 4, chúng ta có: $C_4 = \frac{2}{5}\frac{\sin 4\pi/5}{4\pi/5}$;
- Với n = 5, chúng ta có: $C_5 = \frac{2}{5}\frac{\sin 5\pi/5}{5\pi/5} = 0$.

Tần số tương ứng của các hài được xác định: $f_n = nf_1$, với $f_1 = 1/T = 1/5.10^{-6} = 0{,}2$MHz, sau này chúng ta ký hiệu $f \equiv f_1$ cho tiện.

Từ các kết quả trên, chúng ta nhận thấy với tỉ số $\frac{\tau}{T} = \frac{1}{5}$, biên độ các hài với chỉ số là bội của 5 sẽ bằng 0, tức là: $C_5 = C_{10} = C_{15} = ... = 0$. Điều này tương ứng với các nghiệm của phương

trình hàm *sin(.)* trên tử số của C_n bằng 0, tức là: $\sin n2\pi f\tau / 2 = 0$. Các nghiệm này phù hợp với các tần số $f_k = \frac{k}{\tau}$, với k = 1, 2, ... Chúng ta lại có: $\lim_{f \to 0} \frac{\sin n2\pi f\tau / 2}{n2\pi f\tau / 2} = 1$

Với tất cả các tính toán ở trên, chúng ta có thể vẽ được đồ thị *phổ biên độ – tần số* như hình 5.4. Hơn nữa, đồ thị chỉ vẽ một nửa với f ≥ 0, nửa kia là đối xứng qua trục đứng.

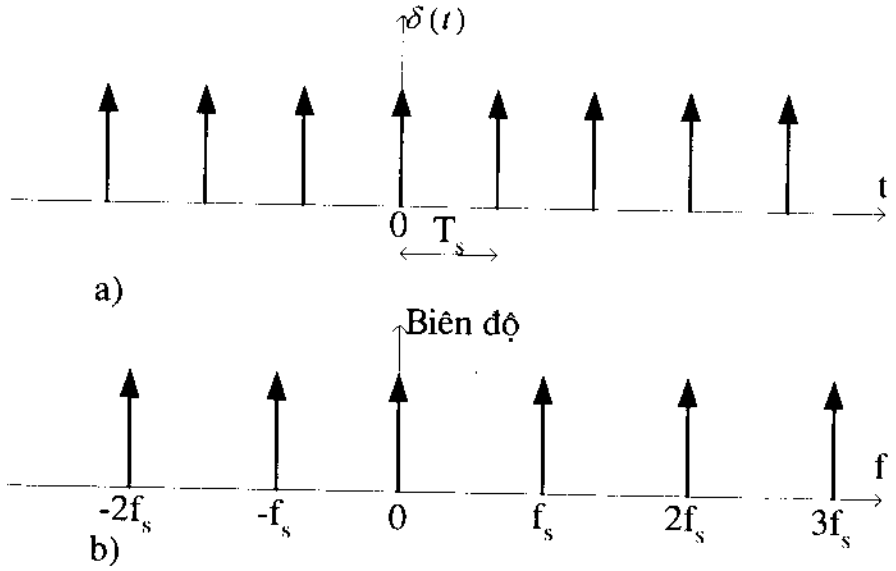


Hình 5.4. Đồ thị phổ biên độ – tần số của chuỗi xung vuông tuần hoàn

(với tỉ số $\frac{\tau}{T} = \frac{1}{5}$).

Từ hình 5.4, chúng ta thấy rằng phổ biên độ – tần số của chuỗi xung vuông tuần hoàn là *phổ rời rạc* (phổ vạch). Hơn nữa, có *vô hạn thành phần tần số*, nhưng *biên độ của chúng giảm dần*. Sự có mặt của vô hạn thành phần tần số được giải thích là do các xung có *sườn dốc đứng* ($\frac{ds(t)}{dt} = \infty$ tại các thời điểm của sườn xung). Với các dạng tín hiệu thực, các sườn xung có độ dốc hữu hạn, nên *biên độ của các hài bậc cao giảm rất nhanh*. Vì thế, có thể cho rằng phổ của chúng là hữu hạn.

♣ Chúng ta xét một *trường hợp đặc biệt*, khi cho các giới hạn: độ rộng các xung τ → 0, còn biên độ của chúng A→ ∞.



Hình 5.5. a) Đồ thị chuỗi xung $\delta(t)$ b) phổ biên độ-tần số của nó.

Khi đó chuỗi xung vuông *s(t)* sẽ tiến đến chuỗi xung *δ(t)*, gọi là chuỗi xung *delta-Dirac*. Toán học định nghĩa xung $\delta(t)$ dưới dạng $\int_{-\infty}^{+\infty}\delta(t)dt = 1$. Đồ thị chuỗi xung *delta-Dirac* và phổ biên độ-tần số nó được vẽ trên hình 5.5. Chúng ta sử dụng các mũi tên để diễn tả các xung $\delta(t)$ và các vạch phổ của nó. Các ký hiệu này ám chỉ chiều cao của chúng là *vô hạn*. Hai đồ thị này rất thuận lợi để giải thích quá trình *lấy mẫu* các tín hiệu tương tự theo điều kiện của *định lý Nyquist*. Trong hình 5.5 chúng ta cũng sử dụng các ký hiệu T_s để chỉ chu kỳ của chuỗi xung lấy mẫu, còn $f_s = \frac{1}{T_s}$ là tần số lấy mẫu.

Từ biểu thức (5.9), toán học cũng định nghĩa *pha đầu của các hài*, ký hiệu $\theta_n(f)$. Ở đây $\theta_n(f) \equiv 0$ với mọi giá trị của chỉ số n.

2.4. Chuỗi Fourier phức

Từ chuỗi Fourier ở dạng thực, xác định theo công thức (5.7), chúng ta có thể dẫn ra chuỗi Fourier phức. Sử dụng các công thức Ơle:

$$\cos n\omega t = \frac{1}{2}\left(e^{in\omega t} + e^{-in\omega t}\right) \text{ và } \sin n\omega t = \frac{1}{2i}\left(e^{in\omega t} - e^{-in\omega t}\right)$$

Đưa tất cả các công thức này vào (5.7), chúng ta có:

$$s(t) = \sum_{n=0}^{\infty}[\frac{a_n}{2}\left(e^{in\omega t} + e^{-in\omega t}\right) - i\frac{b_n}{2}(e^{in\omega t} - e^{-in\omega t})]$$

$$= \sum_{n=0}^{\infty}\left\{\left(\frac{a_n}{2} - i\frac{b_n}{2}\right)\times e^{in\omega t} + \left(\frac{a_n}{2} + i\frac{b_n}{2}\right)\times e^{-in\omega t}\right\}$$

Chúng ta đặt: $\dot{c}_n = \left(\frac{a_n}{2} - i\frac{b_n}{2}\right)$ với $n > 0$

$$\dot{c}_{-n} = \left(\frac{a_{-n}}{2} + i\frac{b_{-n}}{2}\right) \quad \text{với } n < 0 \tag{5.10a}$$

$$\dot{c}_0 = a_0 \quad \text{với } n = 0$$

Đưa các công thức ở (5.10a) vào công thức trên, chúng ta nhận được:

$$s(t) = \sum_{n=-\infty}^{+\infty}\dot{c}_n e^{in\omega t} \tag{5.10b}$$

Công thức (5.10b) gọi là công thức Fourier phức, với các hệ số phức $\dot{c}_n$ được xác định theo công thức:

$$\dot{c}_{\pm n} = \frac{2}{T}\int_0^T s(t).e^{-in\omega t} = \frac{2}{T}\int_{-T/2}^{T/2} s(t)dt \tag{5.10c}$$

Công thức (5.10c) có thể nhận được từ công thức (5.10a) với các thay thế các hệ số của chuỗi Fourier xác định từ công thức (5.8c). Với chỉ số n = 0, hệ số của chuỗi Fourier phức, tính theo (5.10c) phải lấy bằng $\frac{\dot{c}_0}{2}$, điều này suy ra từ công thức (5.8a).

Các hệ số phức có một số tính chất sau:

1. Nếu s(t) là thực thì:

$$\dot{c}_n = \dot{c}^*_{-n};$$

2. Nếu s(t) là thực và chẵn (s(–t) = s(t)) thì:

$$\mathrm{Im}[\dot{c}_{\pm n}] = 0;$$

3. Nếu s(t) là thực và lẻ (s(–t) = –s(t)) thì:

$$\mathrm{Re}[\dot{c}_{\pm n}] = 0.$$

Khi vận dụng chuỗi Fourier ở dạng thực hay phức để xác định phổ của tín hiệu s(t) đều phải cho kết quả như nhau. Để chứng tỏ khẳng định này, chúng ta khảo sát lại thí dụ phân tích chuỗi xung vuông tuần hoàn thành chuỗi Fourier ở trên, nhưng sử dụng các công thức ở dạng phức.

Sử dụng công thức (5.10c), chúng ta có:

$$\dot{c}_n = 2\times\frac{2}{T}\int_{-\tau/2}^{\tau/2} A.e^{-in\omega t}dt = \frac{4A}{T}\frac{e^{-in\omega t}}{-in\omega}\Bigg|_{-\tau/2}^{\tau/2}$$

$$= \frac{4A}{T}\times\frac{\tau}{2}\frac{e^{in\omega\tau/2}-e^{-in\omega\tau/2}}{in\omega\tau/2} = \frac{2A\tau}{T}\frac{\sin n\omega\tau/2}{n\omega\tau/2}$$

Thay kết quả này vào công thức (5.10b), chúng ta được:

$$s(t) = \sum_{n=-\infty}^{\infty}\frac{2A\tau}{T}\frac{\sin n\omega\tau/2}{n\omega\tau/2}\times e^{in\omega t}$$

Để chuyển sang dạng thực, chúng ta thay thế: $e^{in\omega t} = \frac{1}{2}(\cos n\omega t + \sin n\omega t)$

Khi chỉ số n lướt từ $-\infty$ đến $+\infty$ (trừ n = 0), các giá trị theo hàm cosnωt tăng gấp 2, vì cosnωt là hàm chẵn, còn các giá trị theo hàm sinnωt bị triệt tiêu từng đôi một, vì sinnωt là hàm lẻ. Còn khi n = 0, thì $\lim_{n\to 0} e^{in\omega t} = \lim_{n\to 0}\frac{1}{2}\cos n\omega t = \frac{1}{2}$. Cuối cùng, chúng ta được:

$$s(t) = \frac{A\tau}{T} + \sum_{n=-\infty}^{\infty}\frac{2A\tau}{T}\frac{\sin n\omega\tau/2}{n\omega\tau/2}\times\cos n\omega t$$

Kết quả này hoàn toàn phù hợp với việc xác định chuỗi Fourier dưới dạng thực.

3. BIẾN ĐỔI FOURIER

3.1 Biến đổi Fourier

Trong tiết 2, chúng ta đã nghiên cứu vấn đề phân tích dạng sóng s(t) thành chuỗi Fourier và ngụ ý rằng công cụ này sử dụng thuận tiện nhất cho các dạng sóng có chu kỳ. Đối với các dạng sóng không chu kỳ (tín hiệu thực, cũng như các mô hình toán học của tín hiệu thực) sẽ gặp các khó khăn trong thực hành lấy tích phân. Biến đổi Fourier (Fourier Transformation – FT) thích hợp sử dụng cho các dạng sóng có thời gian tồn tại hữu hạn, với điều kiện:

$$\int_{-\infty}^{\infty}|s(t)|dt < \infty \qquad (5.11)$$

Ý nghĩa hình học của điều kiện (5.11) chỉ ra rằng: diện tích của đường cong do hàm dưới dấu tích phân tạo ra với trục thời gian t phải hữu hạn, điều ấy có nghĩa là hàm s(t) phải hữu hạn cả về thời gian, cả về biên độ. Tuy nhiên, điều kiện này là đủ, nhưng không nhất thiết là điều kiện cần, trong một số trường điều kiện (5.11) bị vi phạm, nhưng vẫn xác định được phổ của nó, nhờ các thủ thuật toán học.

Biến đổi Fourier của dạng sóng s(t) được định nghĩa theo công thức sau:

$$S(f) = FT[s(t)] = \int_{-\infty}^{\infty} \left[s(t)e^{-i2\pi f t}\right]dt \qquad (5.12)$$

trong đó FT[.] – ký hiệu phép biến đổi Fourier, tần số f [Hz =1/sec] là tham số của tích phân. *S(f)* – gọi là hàm phổ của dạng sóng s(t). Công thức (5.12) gọi là phép biến đổi Fourier thuận, tức là cho biết s(t) – khả tích, có thể xác định được *S(f)*.

Nói chung, vì $e^{-i2\pi f t}$ là số phức, cho nên *S(f)* là hàm phức của tần số. *S(f)* có thể tách thành phần thực *X(f)*, phần ảo *Y(f)*, sao cho:

$$S(f) = X(f) + i\,Y(f) \qquad (5.13)$$

Cũng có thể biểu diễn *S(f)* dưới dạng:

$$S(f) = |S(f)|e^{i\theta(f)} \qquad (5.14)$$

trong đó: $|S(f)| = \sqrt{X^2(f) + Y^2(f)}$ - là môđun của *S(f)*. Thứ nguyên của $|S(f)|$ được xác định như sau: vì *dt* có thứ nguyên thời gian [sec], còn s(t) giả thiết là dạng sóng thực điện áp, có thứ nguyên [vôn], còn tích *f.t* không có thứ nguyên, cho nên $|S(f)|$ có thứ nguyên [vôn× sec = vôn/Hz]. Vì thế, $|S(f)|$ - gọi là *hàm mật độ phổ biên độ điện áp*, nó đặc trưng cho cho sự phân bố mật độ biên độ điện áp theo tần số.

Còn $\theta(f) = arctg\left(\dfrac{Y(f)}{X(f)}\right)$ - argumen của *S(f)*, gọi là phổ argumen của $|S(f)|$. $\theta(f)$ là hàm không thứ nguyên.

Nếu *S(f)* là hàm khả tích (ít khi gặp), thì sẽ tồn tại biến đổi Fourier ngược, dưới dạng:

$$s(t) = FT^{-1}[S(f)] = \int_{-\infty}^{\infty} [S(f)e^{i2\pi f t}]dt \qquad (5.15)$$

Như vậy, nếu *s(t)* và *S(f)* đều là hàm khả tích, chúng ta có mối quan hệ hai chiều *s(t)* ↔ *S(f)*, gọi là cặp biến đổi Fourier, trong đó s(t) biểu diễn dạng sóng trong miền thời gian, còn *S(f)*- biểu diễn trong miền tần số, gọi là hàm mật độ phổ của s(t).

3.2. Các tính chất của biến đổi Fourier

Các tính toán sẽ thuận tiện, nếu chúng ta xem xét kỹ các tính chất của biến đổi Fourier. Các tính chất của biến đổi Fourier đều được dẫn ra từ công thức định nghĩa (5.12) và công thức thay thế: $e^{-i2\pi f t} = \cos(2\pi ft) - i\sin(2\pi ft)$, như sau:

- Nếu s(t) là hàm thực và chẵn, thì *S(f)* là hàm thực và chẵn. Điều này được giải thích: tích s(t).cos(2πft) là hàm chẵn, còn tích s(t).sin(2πft) là hàm lẻ, mà tích phân của hàm chẵn trong toàn khoảng là hàm chẵn, còn tích phân của hàm lẻ trong toàn khoảng sẽ bằng không;

- Nếu s(t) là hàm thực và lẻ, thì *S(f)* là hàm lẻ và thuần ảo. Điều này cũng giải thích tương tự như trên;

- Nếu s(t) là hàm thực, thì $|S(f)|$ là hàm chẵn, còn $\theta(f)$ là hàm lẻ. Điều này có được, nếu có thể chứng minh:

$$S(-f) = S^*(f) \tag{5.16}$$

trong đó dấu(*) chỉ toán tử liên hợp phức. Thực vậy, từ (5.12) chúng ta có:

$$S(-f) = \int_{-\infty}^{\infty}\left[s(t).e^{i2\pi f t}\right]dt$$

lấy liên hợp phức cả hai vế, chúng ta có:

$$S^*(-f) = \int_{-\infty}^{\infty}\left[s(t).e^{i2\pi f t}\right]^* dt = \int_{-\infty}^{\infty}\left[s^*(t).e^{-i2\pi f t}\right]dt$$

vì s(t) là hàm thực, nên $s^*(t) = s(t)$, cho nên chúng ta có:

$$S^*(-f) = S(f) \text{ hay } S(-f) = S^*(f)$$

Từ đây, hoặc từ chính các định nghĩa của các hàm, chúng ta chứng minh được:

$X(-f) = X(f)$ - hàm chẵn của tần số f;

$Y(-f) = -Y(f)$ - hàm lẻ của tần số f;

$|S(-f)| = |S(f)|$ - hàm chẵn của tần số f;

$\theta(-f) = -\theta(f)$ - hàm lẻ của tần số f.

3.3. Các thí dụ đơn giản

Trước hết, trong mục này chúng ta khảo sát các thí dụ xác định phổ của các dạng sóng (hàm) đơn giản. Kết quả biến đổi Fourier của chúng sẽ được sử dụng trong các mục sau.

a). Xác định phổ của một xung vuông đơn

Dạng sóng *s(t)* là một xung vuông đơn, có đồ thị như hình 5.6a). Biểu thức giải tích của s(t) được xác định:

$$s(t) = \begin{cases} 1 & -\dfrac{T}{2} \le t \le \dfrac{T}{2} \\ 0 & |t| > \dfrac{T}{2} \end{cases} \tag{5.17}$$

Khi nghiên cứu các dạng sóng kiểu xung vuông đơn, có đồ thị như hình 5.6a), người ta thường sử dụng ký hiệu $\Pi\left(\dfrac{t}{T}\right)$ với định nghĩa:

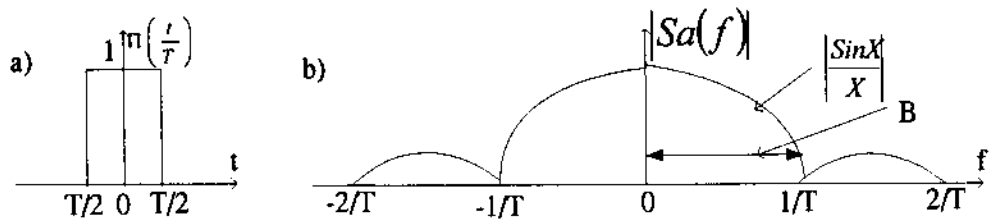
$$\Pi\left(\frac{t}{T}\right)=\begin{cases}1 & |t|\le\frac{T}{2}\\ 0 & |t|>T\end{cases} \tag{5.17a}$$

Sử dụng công thức (5.12) cho biểu thức trên, chúng ta xác định được:

$$S(f)=\int_{-T/2}^{+T/2}\left[\Pi\left(\frac{t}{T}\right).e^{-i2\pi ft}\right]dt=\int_{-T/2}^{+T/2}\left[1.e^{-i2\pi ft}\right]dt$$

$$=1\frac{e^{-i2\pi ft}}{-i2\pi f}\Bigg|_{-T/2}^{+T/2}=T.\frac{\sin(\pi fT)}{\pi fT} \tag{5.18}$$

Đặt $Sa(.)=\frac{Sinx}{x}$ thì $S(f)=T.Sa(\pi fT)$. Đồ thị của hàm mật độ phổ biên độ $|Sa(f)|$ được biểu diễn trên hình 5.6b)



Hình 5.6. Đồ thị dạng sóng xung vuông đơn a) và hàm mật độ phổ biên độ của nó b).

Khi vẽ đồ thị của hàm $|Sa(f)|$ cần chú ý:

$\lim_{f\to 0}|S(f)|=\lim_{f\to 0}\left|T\frac{\sin(\pi fT)}{\pi fT}\right|=T$ và *Sa(f) = 0* tại các tần số được xác định theo:

$\sin(\pi fT)=0\rightarrow f_n=\frac{n}{T}$ với n = ±1, ±2, ±3,...

Với hàm mật độ phổ biên độ như hình 5.6a), người ta cũng thường định nghĩa bề rộng phổ chính của tín hiệu, ký hiệu B [Hz], là khoảng tần số f = (0 ÷ 1/T), tức là B = 1/T. Với khoảng tần số này, tín hiệu chứa khoảng 90% năng lượng. Vì thế, người ta cũng có thể chia hàm phổ của tín hiệu ra hai khoảng: khoảng tần số thấp (Low Frequency – LF), khoảng tần số cao (Higth Frequency – HF). Các xử lý tín hiệu số (phép nén chẳng hạn) phải quan tâm đến sự toàn vẹn hình dạng tín hiệu trong khoảng bề rộng phổ chính.

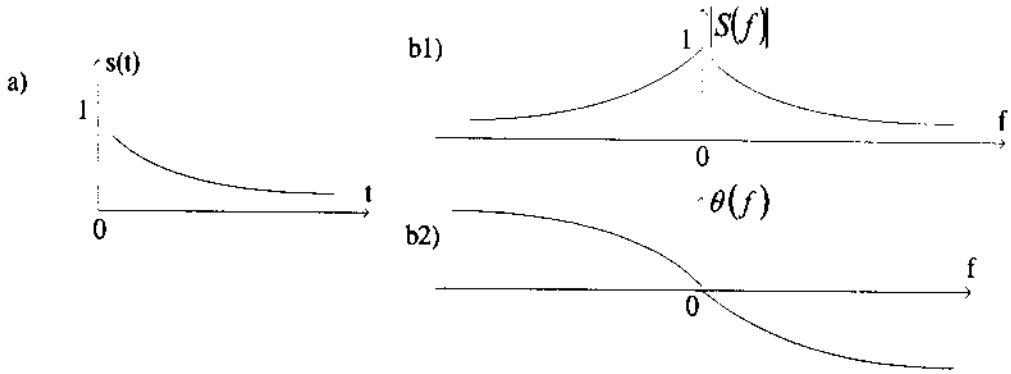
b). Phổ của xung suy giảm theo hàm mũ

Dạng sóng xung suy giảm theo hàm mũ, đồ thị biểu diễn trên hình 5.7a), có biểu thức giải tích như sau:

$$s(t) = \begin{cases} e^{-t} & t \geq 0 \\ 0 & t < 0 \end{cases} \tag{5.19a}$$

Phổ của dạng xung này được xác định theo (5.12):

$$S(f) = \int_0^{\infty} e^{-t}.e^{-i2\pi ft}dt = \frac{1}{1+i2\pi f} \tag{5.19b}$$



Hình 5.7. Đồ thị dạng sóng xung suy giảm theo hàm mũ a) và hàm mật độ phổ biên độ $|S(f)|$ b1), cùng hàm phổ argumen $\theta(f)$ của nó b2).

Biểu diễn *S(f)* dưới dạng: *S(f)* = *X(f)* + *iY(f)*, chúng ta xác định được:

$$X(f) = \frac{1}{1+(2\pi f)^2} \text{ và } Y(f) = \frac{-2\pi f}{1+(2\pi f)^2}$$

Biểu diễn *S(f)* dưới dạng: $S(f) = |S(f)|e^{i\theta(f)}$, chúng ta xác định được:

$$|S(f)| = \sqrt{\frac{1}{1+(2\pi f)^2}} \text{ và } \theta(f) = -arctg(2\pi f)$$

đồ thị của chúng được vẽ trên hình 5.7b1) và hình 5.7b2).

c) Phổ của xung tam giác

Xung tam giác, thường ký hiệu $\Lambda\left(\frac{t}{T}\right)$ và được định nghĩa:

$$\Lambda\left(\frac{t}{T}\right) = \begin{cases} 1-\frac{|t|}{T} & |t| \leq T \\ 0 & |t| > T \end{cases} \tag{5.20}$$

Đồ thị của xung tam giác được biểu diễn trên hình 5.8a)

Phổ của xung tam giác có thể xác định trực tiếp theo (5.12):

$$S(f) = \int_{-T}^{T}\left(1 - \frac{|t|}{T}\right).e^{-i2\pi f t}\,dt$$

Tuy nhiên, xung tam giác có dạng hàm số chẵn, cho nên:

$$S(f) = 2\times\int_{0}^{T}\left(1 - \frac{t}{T}\right).cos(2\pi f t).dt = 2\left[\int_{0}^{T}cos(2\pi f t).dt - \int_{0}^{T}\left(\frac{t}{T}cos(2\pi f t)\right).dt\right] = I_1 - I_2$$

$$I_1 = 2\times\left.\frac{\sin(2\pi f t)}{2\pi f}\right|_{0}^{T} = 0$$

Để xác định I_2, chúng ta sử dụng phương pháp phân đoạn, đặt:

$$u = t \rightarrow du = dt\,;\; dv = cos(2\pi f t) \rightarrow v = \int cos(2\pi f t).dt = -\frac{sin(2\pi f t)}{2\pi f}$$

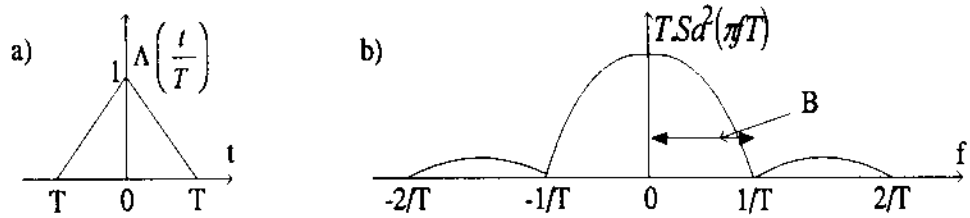
và $\int udv = u.v - \int vdu$; kết quả $I_2 = +2.\frac{t}{T}\times\left.\frac{\sin(2\pi f t)}{2\pi f}\right|_{0}^{T} + 2\frac{1}{T}\left.\frac{\cos(2\pi f t)}{(2\pi f)^2}\right|_{0}^{T}$

chú ý: $1 - \cos(2\pi f T) = 2.\sin^2(\pi f T)$

Kết quả, chúng ta có:

$$S(f) = T.\frac{\sin^2(\pi f T)}{(\pi f T)^2} = T.(Sa)^2 \qquad (5.21)$$

Đồ thị hàm mật độ phổ biên độ của xung tam giác được biểu diễn trên hình 5.8b)



Hình 5.8. Đồ thị xung tam giác a) và hàm mật độ phổ biên độ của nó b).

Bề rộng phổ chính của tín hiệu dạng xung tam giác, ký hiệu B, cũng được định nghĩa là khoảng tần số f = (0 ÷ 1/T), tức là B = 1/T

Với tín hiệu dạng xung tam giác, chúng ta nhận thấy sườn xung của nó không có các đột biến dạng vô cùng, cho nên mật độ biên độ của các thành phần tần số suy giảm rất nhanh (tỉ lệ theo bình phương của *f*). Điều này cho chúng ta xem *tín hiệu thực* có *hàm phổ hữu hạn*.

Kết quả xác định hàm phổ của các dạng sóng khác nhau được trình bày trong mục 3.3 này đều phản ánh đúng các tính chất của FT đã nghiên cứu ở mục 3.2.

3.4. Một số định lý của biến đổi Fourier

Một số định lý của biến đổi Fourier được trình bày trong bảng 5.1. Tất cả các định lý này độc giả đều có thể tự chứng minh dựa vào định nghĩa của FT. Hơn nữa FT là phép biến đổi tuyến tính.

Bảng 5.1. Một số định lý của biến đổi Fourier

Số TT	Phép toán	Hàm	Biến đổi Fourier
1	Tuyến tính	$as_1(t) + bs_2(t)$	$aS_1(f) + bS_2(f)$
2	Trễ thời gian	$s(t-\tau)$	$e^{-i\omega\tau} \times S(f)$
3	Thay đổi tỷ lệ	$s(at),\ a \neq 0$	$\frac{1}{\lvert a\rvert} S(\frac{f}{a})$
4	Dịch tần tín hiệu thực	$s(t).\cos(\omega_c t + \theta)$	$\frac{1}{2}\left[e^{i\theta}S(f-f_c) + e^{-i\theta}S(f+f_c)\right]$
5	Dịch tần tín hiệu phức	$s(t).e^{i\omega t_c}$	$S(f-f_c)$
6	Tín hiệu thông dải	$\mathrm{Re}\left[g(t).e^{i\omega_c t}\right]$	$\frac{1}{2}\left[G(f-f_c) + G^*(-(f+f_c))\right]$
7	Đạo hàm theo t	$\frac{ds(t)}{dt}$	$i2\pi f.S(f)$
8	Tích phân theo t	$\int_{-\infty}^{+\infty} s(\tau)d\tau$	$\frac{1}{i2\pi f}.S(f) + \frac{1}{2}S(0).\delta(f)$
9	Phép nhân	$s_1(t) \times s_2(t)$	$S_1(f) * S_2(f)$
10	Tích chập	$s_1(t) * s_2(t)$	$S_1(f) \times S_2(f)$

Trong bảng 5.1. thuật toán tích chập được định nghĩa:

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau).f_2(t-\tau)d\tau$$

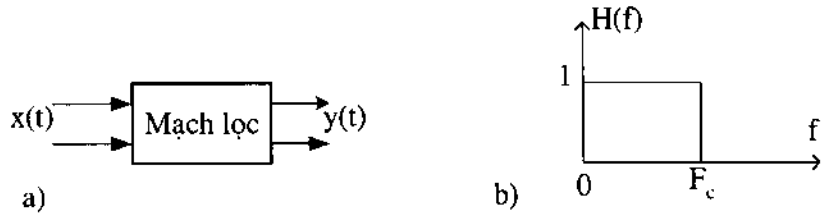
trong đó t – tham số của tích phân.

Hơn nữa, nếu hai hàm trong miền thời gian có quan hệ tích chập (ký hiệu bằng dấu *), thì trong miền tần số hai hàm phổ có quan hệ tích thường, và ngược lại.

Quan hệ tích chập rất thường xuyên được sử dụng để nghiên cứu mạch lọc. Có thể xem mạch lọc như một mạng bốn cực, có tín hiệu đầu vào x(t), tín hiệu đầu ra y(t), được biểu diễn trong hình 5.9a). Theo định lý cơ bản của mạch lọc, chúng ta có:

$$y(t) = x(t) * h(t) \tag{5.22}$$

trong đó h(t) - đáp ứng xung của mạch lọc, nó hoàn toàn xác định các đặc tính của mạch lọc.



Hình 5.9. Mạch lọc như một mạng 4 cực a), hàm truyền đạt H(f) của LPF lý tưởng b).

Theo định lý tích chập, thì FT của (5.22) là:

$$Y(f) = X(f) \times H(f) \tag{5.22a}$$

trong đó hàm truyền đạt theo tần số của mạch lọc, nó quan hệ với h(t) - đáp ứng xung của mạch lọc theo FT, tức là:

$$H(f) = \int_{-\infty}^{\infty} h(t).e^{-i2\pi f t} dt$$

Giả sử một mạch lọc thông thấp (Low Pass Filter - LPF) lý tưởng, có hàm truyền đạt như hình 5.9b), trong đó F_c là tần số cắt của nó. Nếu tín hiệu đầu vào có hàm phổ *X(f)* như hình, chẳng hạn, 5.6b), hoặc 5.8b), thì hàm phổ của tín hiệu đầu ra *Y(f)* có dạng:

$$Y(f) = \begin{cases} X(f) & f \le F_c \\ 0 & f > F_c \end{cases}$$

Nếu *chọn thích hợp* mối quan hệ giữa bề rộng phổ chính B của tìn hiệu đầu vào và tần số cắt F_c, thì hàm phổ của tín hiệu đầu ra *Y(f)* vẫn giữ được nguyên vẹn thành phần phổ chính, khi đó yêu cầu chọn $F_c = B$.

3.5. Các thí dụ khác

Trong mục này, chúng ta dẫn ra vài thí dụ về vận dụng các định lý đã nêu trong bảng 5.1.

a). Phổ của hàm sine suy giảm theo hàm mũ

Xét hàm $\sin\omega t$ suy giảm theo hàm mũ. Biểu thức của nó cho bởi:

$$s(t) = \begin{cases} e^{-\alpha t} \sin \omega_0 t & t \ge 0, \alpha > 0 \\ 0 & t < 0 \end{cases}$$

Khi xác định hàm phổ của s(t) chúng ta có thể *tính trực tiếp* theo định nghĩa của FT, hoặc *gián tiếp* khi sử dụng hai định lý trong bảng 5.1, đó là định lý thay đổi tỷ lệ và định lý dịch tần tín hiệu thực, tức là:

$$\begin{Bmatrix} e^{-\alpha t} & t \geq 0 \\ 0 & t < 0 \end{Bmatrix} \leftrightarrow \frac{1}{|\alpha|} S\left(\frac{f}{\alpha}\right) = \frac{\alpha}{\alpha + i2\pi f}$$

Sử dụng định lý dịch tần tín hiệu thực cho s(t), chúng ta được:

$$S(f) = \frac{1}{2}\left\{e^{-i\pi/2}\frac{\alpha}{\alpha + i2\pi(f - f_0)} + e^{i\pi/2}\frac{\alpha}{\alpha + i2\pi(f + f_0)}\right\}$$

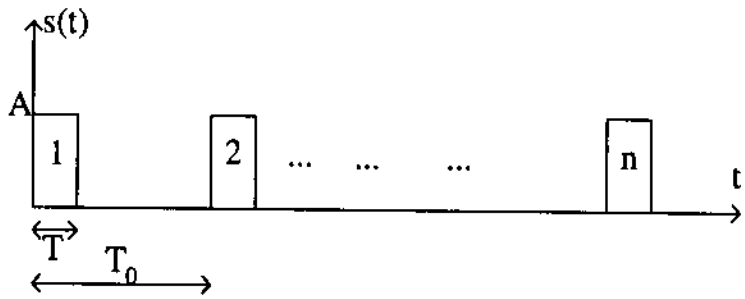
$$= \frac{\alpha}{2i}\left\{\frac{1}{\alpha + i2\pi(f - f_0)} - \frac{1}{\alpha + i2\pi(f + f_0)}\right\}$$

Vì s(t) là hàm thực (không chẵn, hoặc lẻ), cho nên hàm phổ này có đặc điểm:

- Phổ môdun là hàm phổ chẵn. Đồ thị của nó có dáng giống hình 5.7b1), nhưng dịch theo trục f dương (và f âm) một khoảng tần số $f = f_0$ (và $f = -f_0$);
- Phổ argumen là hàm phổ lẻ. Đồ thị của nó có dáng giống hình 5.7b2), nhưng dịch theo trục f dương (và f âm) một khoảng tần số $f = f_0$ (và $f = -f_0$).

b) Phổ của chùm n xung vuông

Giả thử chúng ta có một tín hiệu "giả chu kỳ", được hình thành từ chùm n xung vuông, đồ thị của nó được biểu diễn trong hình 5.10a).



Hình 5.10a). Đồ thị của chùm n xung vuông.

Để xác định hàm phổ của s(t), chúng ta sử dụng định lý *xếp chồng* phổ, tức là: nếu gọi hàm phổ của xung thứ nhất là $S_1(f)$, được xác định theo công thức (5.18), thì hàm phổ của xung thứ 2, thứ 3,..., n, được xác định:

$$S_n(f) = e^{-i2\pi f T_0} \times S_1(f)$$

trong đó T_0 - chu kỳ giả của chùm xung.

Theo định lý *xếp chồng* phổ, chúng ta xác định được:

$$S(f) = S_1(f) + S_2(f) + ... + S_n(f)$$

$$S_1(f)\left[1 + e^{-i2\pi f T_0} + e^{-i2\pi f T_0} + ... + e^{-i2\pi f T_0}\right]$$

Đây là tổng của cấp số nhân, với số hạng thứ nhất $u_1 = 1$, còn công bội $q = e^{-i2\pi f T_0}$. Tổng của cấp số nhân xác định theo công thức:

$$T_n = \frac{u_1\left(q^n - 1\right)}{q - 1}$$

Từ đó, chúng ta xác định được hàm phổ của s(t) như sau:

$$S(f) = S_1(f) \times e^{-i(n-1).\pi.f.T_0} \times \frac{\sin(n\pi.f.T_0)}{\sin(\pi.f.T_0)} \tag{5.23}$$

• Tại các tần số $f = \frac{k}{T_0} = k.f_0$, k = 0, 1, 2, ..., biểu thức $\frac{\sin(n\pi.f.T_0)}{\sin(\pi.f.T_0)}$ có dạng $\frac{0}{0}$. Sử dụng công thức H'opitan, chúng ta xác định được:

$$\lim_{f \to kf_0} \frac{\sin(n\pi.f.T_0)}{\sin(\pi.f.T_0)} = n$$

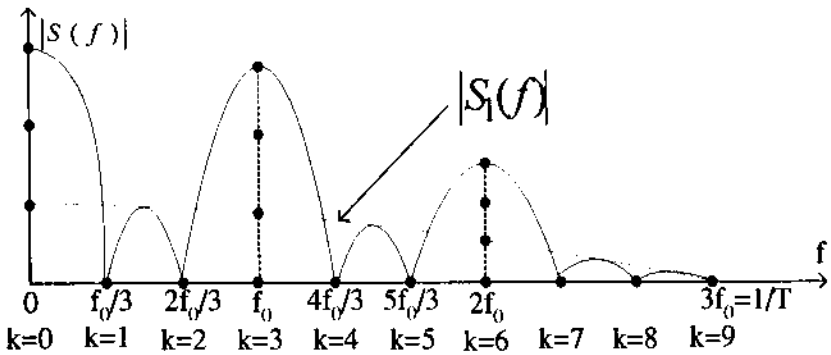
Vậy tại các tần số $f_k = \frac{k}{T_0} = k.f_0$, chúng ta có: $S(f) = n \times S_1(f) \times e^{-i(n-1).\pi.f.T_0}$

Hay
$$|S(f)| = n|S_1(f)| \tag{5.24}$$

• Tại các tần số $f = \frac{k}{n.T_0} = \frac{k}{n} f_0$ → Biểu thức hàm sin(.) trên tử số bằng không, còn biểu thức hàm sin(.) ở mẫu số được xác định bằng $\sin\left(\frac{k\pi}{n}\right)$, nếu tỉ số k/n khác số nguyên, thì mẫu số khác không. Lúc đó *S(f) = 0*. Nếu k/n = số nguyên, thì $|S(f)| \neq 0$, và đạt giá trị max.

Như vậy, tại các tần số $f_{max} = \frac{k}{T_0} \to |S(f)|_{max}$, còn các tần số $f = \frac{k}{n.T_0}$ với k/n ≠ số nguyên $\to |S(f)|_{min}$. Nếu k/n = số nguyên, thì $|S(f)|_{max}$. Căn cứ vào các phân tích ở trên, nếu cho trước tỉ số T/T_0, chúng ta có thể vẽ được đồ thị hàm $|S(f)|$

b)



Hình 5.10b). Đồ thị hàm mật độ phổ $|S(f)|$ của chùm 3 xung vuông.

Trên hình 5.10b), biểu diễn đồ thị hàm mật độ phổ $|S(f)|$ chùm 3 (n = 3) xung vuông, với tỉ số $\frac{T}{T_0} = \frac{1}{3}$. Tại các điểm $f = 0, f_0, 2f_0 \rightarrow |S(f)|_{max}$. Đường cong đứt nét tương ứng với hàm mật độ phổ $|S_1(f)|$ vẽ trong khoảng tần số f = 0 ÷ 1/T.

4. BIẾN ĐỔI FOURIER RỜI RẠC

Biến đổi Fourier rời rạc (Discrete Fourier Transformation - DFT) là một trong các thuật toán được sử dụng rộng rãi nhất trong kỹ thuật xử lý tín hiệu số. Trong tiết này, chúng ta sẽ nghiên cứu sơ bộ về DFT, chi tiết hơn độc giả có thể tham khảo ở các tài liệu khác, chẳng hạn [12].

Xét tín hiệu (dạng sóng) x(t) liên tục trong khoảng thời gian hữu hạn, chẳng hạn, từ 0 đến T_A nào đó. Chia khoảng thời gian (0 ÷ T_A) ra N khoảng bằng nhau, ký hiệu các thời điểm tương ứng là t_k với k = 0, 1, 2,..., N-1. Ứng với mỗi thời điểm t_k, chúng ta xác định được một mẫu của tín hiệu, ký hiệu là x(k). *DFT lập quan hệ về thuật toán giữa các dãy hữu hạn, rời rạc trong miền thời gian, ký hiệu x(k) và các dãy hữu hạn, rời rạc trong miền tần số, ký hiệu X(n).* DFT được suy ra từ *biến đổi Fourier liên tục (CFT)*, với việc thay thế tích phân bởi tổng, còn biến tần số f, thời gian t, được thay thế bằng tần số rời rạc n, thời gian rời rạc k. DFT định nghĩa bởi thuật toán:

$$X(n) = DFT[x(k)] = \sum_{k=0}^{N-1} x(k).e^{-i\frac{2\pi}{N}.n.k} \tag{5.25}$$

trong đó n = 0, 1, 2,..., N-1.

Biến đổi Fourier rời rạc ngược (Inverse Discrete Fourier Transformation - IDFT, hoặc DFT^{-1}) được định nghĩa bởi thuật toán sau:

$$x(k) = DFT^{-1}[X(n)] = \frac{1}{N}\sum_{n=0}^{N-1} X(n).e^{i\frac{2\pi}{N}.k.n} \tag{5.26}$$

trong đó k = 0, 1, 2,..., N-1.

Tổng trong các công thức (5.25) và (5.26) thực chất là tổng kép theo cả hai chỉ số k, n. Khi tính toán, thường biểu diễn chúng dưới dạng ma trận, như sau:

$$[X(n)] = \left[e^{-i\frac{2\pi}{N}.k.n}\right] \times [x(k)] \quad \text{và} \quad [x(k)] = \frac{1}{N}\left[e^{i\frac{2\pi}{N}.k.n}\right] \times [X(n)]$$

hoặc trong dạng cụ thể hơn của các ma trận:

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ . \\ . \\ . \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{-i\frac{2\pi}{N}} & e^{-i\frac{2}{N}.2} & \cdots & e^{-i\frac{2\pi}{N}(N-1)} \\ 1 & e^{-i\frac{2\pi}{N}.2} & e^{-i\frac{2\pi}{N}.4} & \cdots & e^{-i\frac{2\pi}{N}.2(N-1)} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ 1 & e^{-i\frac{2\pi}{N}(N-1)} & e^{-i\frac{2\pi}{N}.2.(N-1)} & \cdots & e^{-i\frac{2\pi}{N}(N-1)^2} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ . \\ . \\ . \\ x(N-1) \end{bmatrix} \quad (5.25a)$$

$$X(n) = DFT[x(k)] = \mathrm{Re}[X(n)] + \mathrm{Im}[X(n)]$$

biến k = 0, 1 ,..., N-1, chạy theo hàng;

còn biến n = 0, 1 ,..., N-1, chạy theo cột;

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ . \\ . \\ . \\ x(N-1) \end{bmatrix} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & e^{i\frac{2\pi}{N}} & e^{i\frac{2}{N}.2} & \cdots & e^{i\frac{2\pi}{N}(N-1)} \\ 1 & e^{i\frac{2\pi}{N}.2} & e^{i\frac{2\pi}{N}.4} & \cdots & e^{i\frac{2\pi}{N}.2(N-1)} \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ . & . & . & \cdots & . \\ 1 & e^{i\frac{2\pi}{N}(N-1)} & e^{i\frac{2\pi}{N}.2.(N-1)} & \cdots & e^{i\frac{2\pi}{N}(N-1)^2} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ . \\ . \\ . \\ X(N-1) \end{bmatrix} \quad (5.26a)$$

$$x(k) = DFT^{-1}[X(n)] = \mathrm{Re}[x(k)] + \mathrm{Im}[x(k)]$$

biến n = 0, 1 ,..., N-1, chạy theo hàng;

còn biến k = 0, 1 ,..., N-1, chạy theo cột.

Cũng như mọi biểu diễn trong miền tần số, *X(0)* - là thành phần *một chiều* của phổ rời rạc, còn *X(N-1)* - là thành phần *cao tần nhất* của phổ rời rạc.

Mỗi ma trận vuông biểu diễn các hệ số có N^2 số hạng, trong dạng:

$$e^{\mp i\frac{2\pi}{N}k.n} = \cos\left(\frac{2\pi}{N}.k.n\right) \mp i\sin\left(\frac{2\pi}{N}.k.n\right)$$

Chúng ta nhận thấy rằng các số hạng đối xứng qua đường chéo chính bằng nhau tương ứng. Trong thực hiện DFT và IDFT, phép tính với các số hạng gặp nhiều sự trùng lặp, do đặc tính *chu kỳ* và *đối xứng* của hàm xác định hệ số, vì vậy có thể tìm cách tính nhanh. Thuật toán thực hiện tính toán nhanh gọi là thuật toán *biến đổi Fourier nhanh* (Fast Fourier Transformation-*FFT*), cùng với biến đổi Fourier nhanh cũng có *biến đổi Fourier nhanh ngược* (Inverse Fast Fourier Transformation - *IFFT*, hoặc FFT^{-1}).

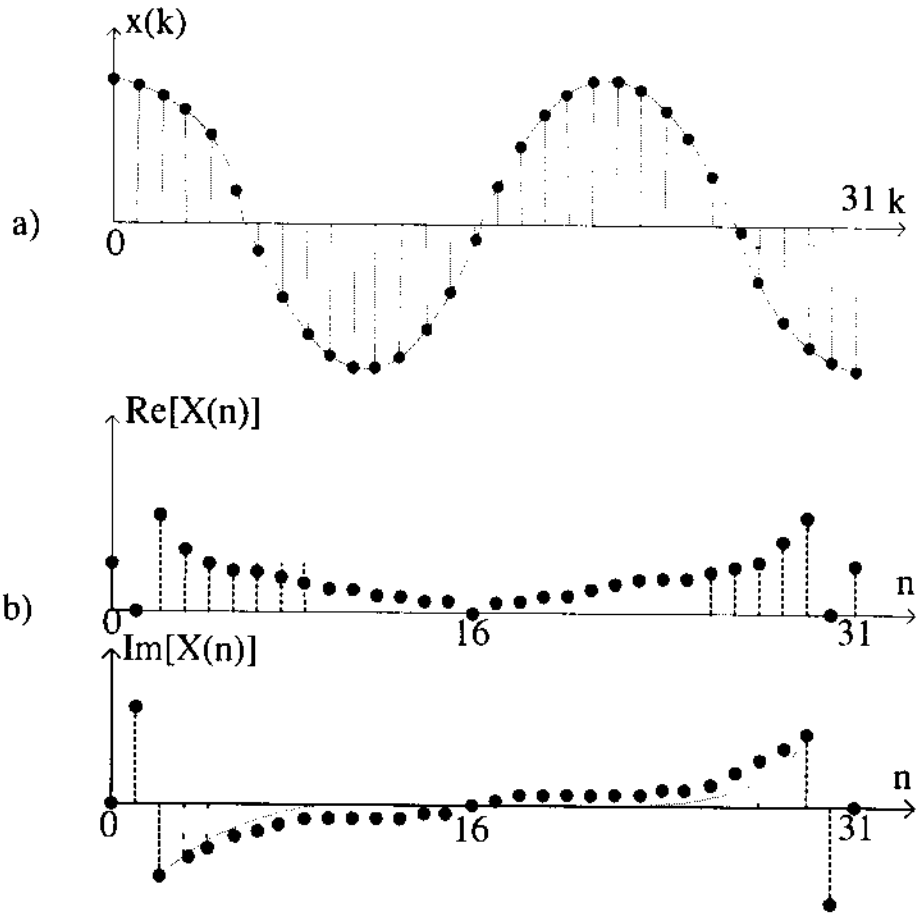
Sau đây, chúng ta xét một thí dụ về DFT của dãy rời rạc xuất phát từ hàm cosαt, với N = 32 điểm, với $\alpha = \frac{3\pi}{31} \approx 0{,}304$. Đồ thị của dãy rời rạc 32 điểm biểu diễn trên hình 5.11a), còn phép biến đổi thực hiện theo công thức:

$$X(n) = DFT[x(k)] = \mathrm{Re}[X(n)] + \mathrm{Im}[X(n)] = \sum_{k=0}^{N-1} \cos\left(\frac{3\pi}{(N-1)}..k\right).e^{-i\frac{2.\pi}{N}.n.k} \qquad (5.27)$$

trong đó n là tham biến lấy giá trị từ 0 đến 31.

Thay thế $\cos\left(\frac{3\pi}{(N-1)}.k\right) = \frac{1}{2}\left(e^{i\frac{3\pi}{(N-1)}.k} + e^{-i\frac{3\pi}{(N-1)}.k}\right)$ vào (5.27), chúng ta có:

$$X(n) = \sum_{k=0}^{N-1} 0{,}5.\left(e^{i\frac{3\pi}{(N-1)}.k} \times e^{-i\frac{2.\pi}{N}.n.k}\right) + \sum_{k=0}^{N-1} 0{,}5\left(e^{-i\frac{3\pi}{(N-1)}.k} \times e^{-i\frac{2.\pi}{N}.n.k}\right) = T_1 + T_2$$



Hình 5.11. Đồ thị dãy *x(k)* a) và *Re[X(n)]* cùng *Im[X(n)]*.

trong đó T_1 , T_2 là tổng của cấp số nhân theo k, có $u_0 = 1$, $q = e^{\pm i\left(\frac{3\pi}{(N-1)} \mp \frac{2\pi}{N}.n\right)}$, dấu ở trên cho T_1, còn dấu ở dưới cho T_2. Sử dụng công thức tính tổng cấp số nhân dạng $T = \frac{\left(q^{(N-1)} - 1\right)}{q-1}$ cho hai tổng trên và thực hiện vài biến đổi, chúng ta nhận được:

$$X(n) = \frac{0,5.\left(1 + e^{i\frac{2\pi}{N}.n}\right)}{1 - e^{i\left(\frac{3\pi}{N-1} - \frac{2\pi}{N}.n\right)}} + \frac{0,5.\left(1 + e^{-i\frac{2\pi}{N}.n}\right)}{1 - e^{-i\left(\frac{3\pi}{N-1} + \frac{2\pi}{N}.n\right)}} \qquad (5.27a)$$

Căn cứ vào công thức (5.27a), với n = 0 ÷ (N-1), chúng ta xác định được Re[X(n)] và Im[X(n)]. Phụ lục 1 biểu diễn chương trình và kết quả tính *DFT[cosak]*.

Đồ thị của Re[X(n)] và Im[X(n)] biểu diễn trên hình 5.11b).

Ngày nay có sẵn *các vi xử lý chuyên dụng* để thực hiện các thuật toán FFT và FFT^{-1} với số điểm rời rạc khác nhau.

5. BIẾN ĐỔI COSINE RỜI RẠC

Trong kỹ thuật nén (tín hiệu ảnh, hoặc tín hiệu audio) rất thường hay sử dụng biến đổi cosine rời rạc (Discrete Cosine Transformation - DCT). DCT có thể thực hiện biến đổi ở dạng tọa độ một chiều, hoặc hai chiều.

5.1. DCT một chiều

Trước hết, chúng ta nghiên cứu DCT một chiều. Giả thử chúng ta có dãy rời rạc, hữu hạn x(k), trong đó k = 0, 1, 2,..., (N-1), còn N - số điểm rời rạc hóa, x(k) - các mẫu biên độ của tín hiệu x(t) hữu hạn.

DCT một chiều thực hiện biến đổi dãy rời rạc, hữu hạn, ký hiệu x(k), trong miền thời gian (hoặc không gian) thành dãy rời rạc, hữu hạn, ký hiệu X(n), trong miền tần số

DCT một chiều được định nghĩa bởi thuật toán sau:

$$X(n) = \sqrt{\frac{2}{N}}.C(n)\sum_{k=0}^{N-1} x(k).\cos\left(\frac{(2k+1).n.\pi}{2N}\right) \qquad (5.28)$$

trong đó: n = 0, 1, 2,..., (N-1)

$$C(n) = \begin{cases} \frac{1}{\sqrt{2}} & n = 0 \\ 1 & n \neq 0 \end{cases}$$

X(n) - các mẫu tần số của tín hiệu x(k), hơn nữa X(0) - mẫu ứng với tần số f = 0, gọi là mẫu một chiều của tín hiệu x(k). Khi chỉ số n tăng dần, chúng ta có các mẫu với tần số tăng dần.

Công thức (5.28) có thể biểu diễn trong dạng ma trận, như sau:

$$[X(n)] = \sqrt{\frac{2}{N}}.C(n).[x(k)].[B(k,n)] \tag{5.29}$$

trong đó: [x(k)] - ma trận hàng, với k = 0, 1, 2,..., (N-1), biểu diễn các giá trị mẫu của biên độ tín hiệu cần thực hiện DCT;

[B(k, n)] - ma trận vuông (k hàng × n cột), với n, k = 0, 1, 2,..., (N-1). [B(k, n)] - gọi là ma trận của hàm cơ sở, được xác định theo:

$$[B(k,n)] = \left[\cos\left(\frac{(2k+1)n.\pi}{2.N}\right)\right] \tag{5.30}$$

và khi N = 8, nó có giá trị như sau:

$$[B(k,n)] = \begin{bmatrix} 1 & ,9807 & ,9238 & ,8314 & ,7071 & ,5555 & ,3826 & ,1950 \\ 1 & ,8314 & ,3826 & -,1950 & -,7071 & -,9807 & -,9238 & -,5555 \\ 1 & ,5555 & -,3826 & -,9807 & -,7071 & ,1950 & ,9238 & ,8314 \\ 1 & ,1950 & -,9238 & -,5555 & ,7071 & ,8314 & -,3826 & -,9807 \\ 1 & -,1950 & -,9238 & ,5555 & ,7071 & -,8314 & -,3826 & ,9807 \\ 1 & -,5555 & -,3826 & ,9807 & -,7071 & -,1950 & ,9328 & -,8314 \\ 1 & -,8314 & ,3826 & ,1950 & -,7071 & ,9807 & -,9328 & ,5555 \\ 1 & -,9807 & ,9238 & -,8314 & ,7071 & -,5555 & ,3826 & -,1950 \end{bmatrix} \tag{5.30a}$$

Chú ý : khi xác định [B(k, n)] tham số n- chỉ số của cột, tham số k- chỉ số của hàng. Trong ma trận này, chỉ biểu diễn các số sau dấu phẩy, hơn nữa, chỉ lấy chính xác đến 4 số lẻ. Chúng ta nhận thấy rằng trong [B(n, k)] có nhiều giá trị giống nhau, vì vậy cũng có thể xây dựng *thuật toán tính nhanh* DCT.

Chúng ta xét một thí dụ: giả thử chúng ta có 8 mẫu biên độ của tín hiệu x(k), như ma trận dòng dưới đây:

[x(k)] = [98, 92, 96, 80, 75, 69, 95, 100].

Sử dụng công thức (5.28), để xác định các mẫu tần số X(n) – *xem phụ lục 2* -, kết quả nhận được như ma trận dòng dưới đây:

[X(n)] = [249; 6; 24; - 15; 0; 6; - 2; 9]

Trong dãy này, thành phần phổ một chiều X(0) = 249; thành phần tần số cao nhất X(7) = 9. Chúng ta thấy rằng biên độ của các thành phần tần phổ, nói chung, giảm dần theo sự tăng của tần số.

DCT một chiều ngược, ký hiệu IDCT, hoặc DCT^{-1}, biểu diễn bằng thuật toán:

$$x(k) = \sqrt{\frac{2}{N}}.C(k).\sum_{n=0}^{N-1} X(n).\cos\left[\frac{(2n+1)k.\pi}{2N}\right] \quad (5.31)$$

trong đó: k = 0, 1, 2,..., (N-1)

$$C(k) = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & k \neq 0 \end{cases}$$

Công thức (5.31) cũng có thể biểu diễn dưới dạng ma trận, như sau:

$$[x(k)] = \sqrt{\frac{2}{N}}.C(k).[X(n)][B(n,k))] \quad (5.31)$$

trong đó: [X(n)] - ma trận hàng, n = 0, 1, 2,..., (N-1);

[B(n, k)] - ma trận vuông (n hàng × k cột), với k, n = 0, 1, 2,..., (N-1).

[B(n, k)] - gọi là ma trận hàm cơ sở, được xác định theo:

$$[B(n,k)] = \left[\cos\left(\frac{(2n+1)k.\pi}{2.N}\right)\right]$$

Như vậy: $[B(n,k)] = [B(k,n)]^T$, hoặc ngược lại; ở đây ký hiệu T- chỉ toán tử hoán vị. Vì thế, khi đã biết [B(k, n)] bằng cách hoán vị chúng ta xác định ngay được [B(n, k)].

Thuật toán DCT một chiều có thể được sử dụng trong kỹ thuật nén các tín hiệu chỉ phụ thuộc vào một biến số không gian.

5.2. DCT hai chiều

DCT hai chiều thực hiện biến đổi các mẫu rời rạc, hữu hạn trong không gian hai chiều của biên độ tín hiệu, ký hiệu x(i, j), thành các mẫu rời rạc, hữu hạn trong không gian hai chiều của tần số, ký hiệu X(u, v).

DCT hai chiều biểu diễn bằng thuật toán:

$$X(u,v) = \frac{2}{N}C(u).C(v).\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(i,j).\cos\left[\frac{(2i+1)u\pi}{2N}\right].\cos\left[\frac{(2j+1)v\pi}{2N}\right] \quad (5.32)$$

trong đó:

$$C(u), C(v) = \begin{cases} \frac{1}{\sqrt{2}} & u, v = 0 \\ 1 & u, v \neq 0 \end{cases}$$

Các thành phần hàm cosine gọi là hàm cơ sở B(i, j). Hàm cơ sở B(i, j) tạo thành ma trận $[B(i,j)]$ - gọi là ma trận của hàm cơ sở, (i cột × j hàng), được xác định theo công thức:

$$[B(i,j)] = \left[\cos\left(\frac{(2i+1).u.\pi}{2N}\right) \times \cos\left(\frac{(2j+1).v.\pi}{2N}\right)\right] \tag{5.33}$$

trong đó (u, v) - các tham biến, với u, v = 0, 1, 2,..., (N-1). Tương ứng với mỗi cặp giá trị của tham biến (u, v), chúng ta xác định được một ma trận $[B(i,j)]$, với N = 8 và cặp (u, v) = (6,4), chúng ta xác định được:

$$[B] = \begin{bmatrix} ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \\ ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ -,2706 & ,6533 & -,6533 & ,2706 & ,2706 & -,6533 & ,6533 & -,2706 \\ ,2706 & -,6533 & ,6533 & -,2706 & -,2706 & ,6533 & -,6533 & ,2706 \end{bmatrix}$$

trong ma trận này, chỉ biểu diễn các số sau dấu phẩy, với độ chính xác đến 4 chữ số.

Thuật toán DCT ngược, ký hiệu IDCT, hoặc DCT^{-1} xác định mối quan hệ giữa dãy rời rạc, hữu hạn trong không gian hai chiều của tần số, ký hiệu X(u, v), với dãy rời rạc, hữu hạn trong không gian hai chiều biểu diễn biên độ tín hiệu, ký hiệu x(i, j). DCT^{-1} biểu diễn bởi thuật toán:

$$x(i,j) = \frac{2}{N} C(i).C(j).\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} X(u,v).\cos\left(\frac{(2u+1)\pi.i}{2N}\right).\cos\left(\frac{(2v+1)\pi.j}{2N}\right) \tag{5.34}$$

trong đó C(i), C(j) có giá trị như C(u), C(v). N - số điểm lấy mẫu, khi sử dụng trong kỹ thuật nén ảnh, chọn N = 8.

Các thành phần hàm cosine gọi là hàm cơ sở B(u,v). Hàm cơ sở B(u,v) tạo thành ma trận [B(u,v)] - ma trận của hàm cơ sở theo biến (u,v), được xác định:

$$[B(u,v)] = \left[\cos\left(\frac{(2u+1)\pi.i}{2N}\right).\cos\left(\frac{(2v+1)\pi.j}{2N}\right)\right] \tag{5.35}$$

trong công thức này cặp (i, j) - là cặp tham biến, với khoảng giá trị 0 ≤ (i,j) ≤ N. Cặp biến số (u,v) cũng có giá trị trong khoảng 0 ≤ (u,v) ≤ N. Giữa ma trận [B(i, j] và ma trận [B(u, v)] có cũng dạng. Nếu thực hiện hoán đổi vị trí: (i, j) ↔ (u, v) tương ứng giữa hai ma trận, thì chúng có cùng giá trị.

6. BIẾN ĐỔI WIERNER - KHINTCHINE

Khi nghiên cứu các quá trình ngẫu nhiên - QTNN, toán học phân loại chúng thành hai nhóm lớn:

- QTNN dừng;
- QTNN không dừng.

Đối với các QTNN dừng, hàm *mật độ phổ công suất* của chúng được xác định qua hàm tự tương quan, được biểu diễn bằng công thức sau:

$$S(f) = \int_{-\infty}^{+\infty} R(\tau) e^{-i2\pi.f.\tau} d\tau \tag{5.36}$$

Công thức (5.36) gọi là biến đổi Wierner-Khintchine thuận, đó là sự mở rộng của FT cho các QTNN dừng, trong đó R(τ)-là hàm tự tương quan của QTNN dừng X(t), định nghĩa theo công thức:

$$R(\tau) = E[X(t) \times X(t-\tau)] \tag{5.37}$$

trong công thức (5.37), ký tự E chỉ toán tử kỳ vọng. Hàm R(τ) đo mức tương quan thống kê giữa X(t) với chính nó, sau khi đã dịch theo trục thời gian một khoảng bằng ±τ. Nếu R(τ) ≠ 0, thì ta nói rằng giữa X(t) và X(t−τ) phụ thuộc thống kê lẫn nhau, tức là chúng có sự "giống nhau" nào đó. Ngược lại, nếu R(τ) = 0, thì nói rằng giữa X(t) và X(t−τ) không phụ thuộc thống kê lẫn nhau, tức là "hành vi" của chúng hoàn toàn "xa lạ" với nhau. Rõ ràng, khi τ = 0, chúng là một, vì thế hàm R(0) đạt cực đại, còn khi khoảng τ tăng dần về cả hai phía, hàm R(τ) giảm dần, cũng có nghĩa là hàm R(τ) là hàm thực và chẵn. Từ đó, công thức (5.36) được cụ thể hơn như sau:

$$S(f) = \int_{-\infty}^{+\infty} R(\tau) \cos(2\pi.f.\tau).d\tau \tag{5.36a}$$

Cũng tương tự như FT, hàm *S(f)* là thực và chẵn.

Giả thử hàm X(t) có thứ nguyên [vôn], thì hàm R(τ) có thứ nguyên [watt]. Từ công thức (5.36), suy ra thứ nguyên của S(f) là $\left[\frac{W}{Hz}\right]$. Hàm S(f) - gọi là hàm mật độ phổ công suất (Power Spectrum Density - PSD), như chúng ta đã gọi từ đầu. Biến đổi Wierner-Khintchine ngược được định nghĩa bằng công thức:

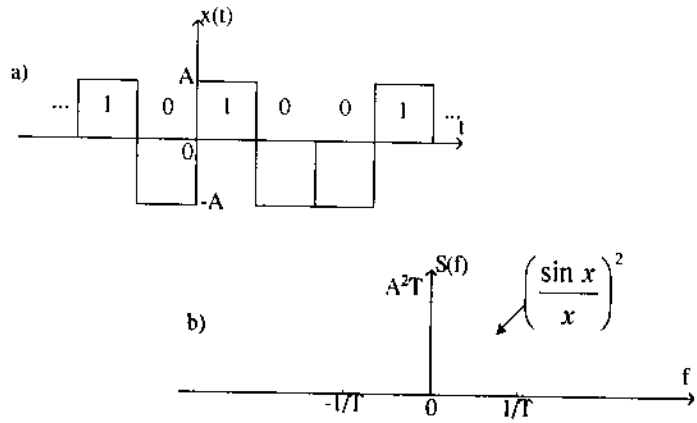
$$R(\tau) = \int_{-\infty}^{+\infty} S(f).e^{i2\pi.f.\tau} df = \int_{-\infty}^{+\infty} S(f).\cos(2\pi.f.\tau).df \tag{5.38}$$

Rõ ràng, không giống như FT^{-1}, từ hàm S(f) không có thể khôi phục được bất kỳ một thể hiện nào của X(t), vì nó không mang thông tin về pha của các thể hiện.

Sau đây, chúng ta xem xét một thí dụ điển hình, mà sau này (xem chương 6) rất thường xuyên được sử dụng để giải thích sự hình thành hàm PSD của các tín hiệu điều chế số.

Thí dụ: Xét tín hiệu ngẫu nhiên X(t), là chuỗi bít đầu ra, kết quả của một phép mã hóa nguồn nào đó, mà một thể hiện của nó được biểu diễn trên hình 5.12a).

Với các thể hiện x(t), chúng ta giả thiết rằng: sự xuất hiện của các bit trên trục thời gian là ngẫu nhiên, nhưng số lượng các bit có mức logic 1, bằng số lượng các bit có mức logic 0, tức là xác suất xuất hiện các bit 1 và các bit 0 bằng 1/2. Độ rộng các bit là T [sec], biên độ của chúng là ±A [vôn]. Một tín hiệu ngẫu nhiên, mà các thể hiện của nó phù hợp với giả thiết trên, chúng ta sẽ chứng minh rằng: X(t) - tín hiệu ngẫu nhiên dừng.



Hình 5.12. Một thể hiện x(t) của tín hiệu ngẫu nhiên X(t) a), và hàm mật độ phổ công suất của nó b).

Quá trình chứng minh như sau:

- QTNN X(t) là QTNN rời rạc, với tập bao gồm chỉ hai giá trị $x_1 = +A$, $x_2 = -A$, hơn nữa $p(x_1) = p(x_2) = 1/2$.

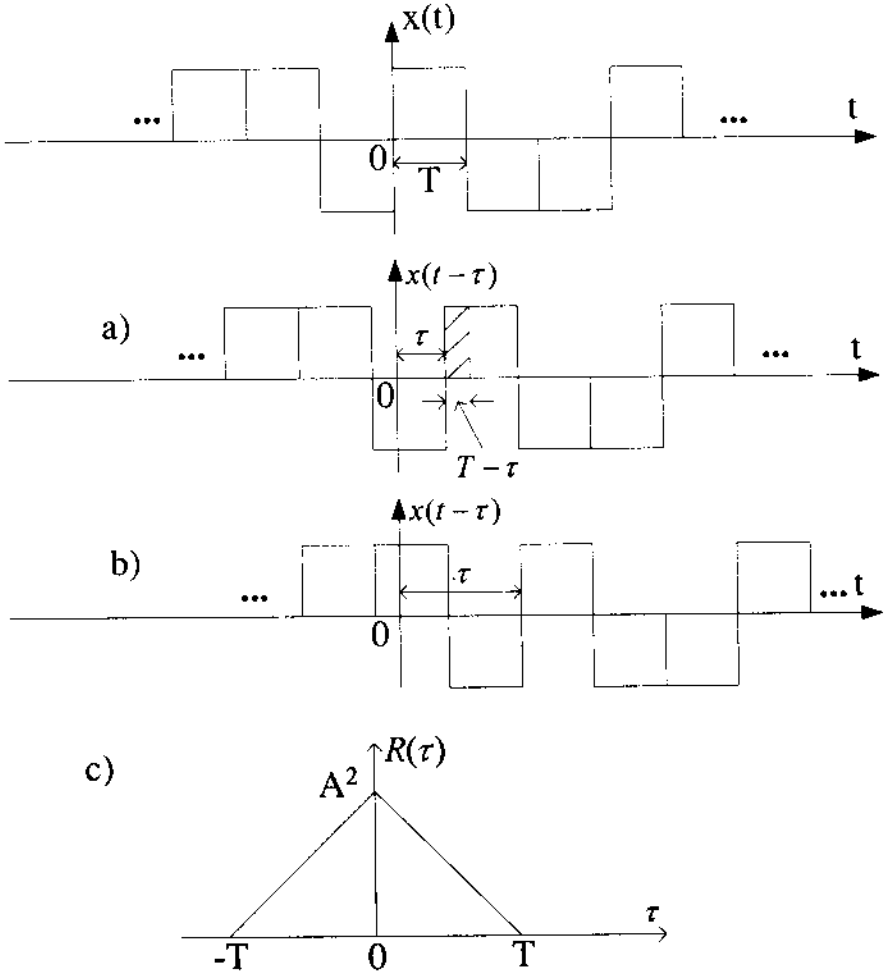
- Hàm kỳ vọng: $m_X(t) = E[X(t)] = \sum_{i=1}^{2} x_i.p(x_i) = A.\frac{1}{2} - A.\frac{1}{2} = 0$;

- Hàm phương sai: $D_X(t) = E\{[X(t) - m_X(t)]^2\} = \sum_{i=1}^{2} x_i^2 p(x_i) = A^2.\frac{1}{2} + (-A)^2.\frac{1}{2} = A^2$;

- Hàm kỳ vọng:

$$R_X(t_1,t_2) = E[X(t_1).X(t_2)] = \sum_{i=1}^{2}\sum_{j=1}^{2} x_i(t_1).x_j(t_2).p[x_i(t_1).x_j(t_2)],$$

trong đó $p[x_i(t_1).x_j(t_2)]$ - xác suất giao của hai sự kiện $x_i(t_1)$ với $x_j(t_2)$. Chúng ta xác định $R(t_1, t_2)$ trong hai trường hợp sau:



Hình 5.13. Diễn giải quá trình tính hàm tự tương quan: khi $|T-\tau| \le 0$ a), khi $|T-\tau| > 0$ b) và kết quả c).

• Khi $\tau = |t_2 - t_1| > T$, chúng ta có:

$$x_i(t_1).x_j(t_2) = \begin{cases} A^2 & a) \\ -A^2 & b) \end{cases}$$

trường hợp a) xảy ra khi t_1 và t_2 rơi vào khoảng để $x_i(t_1)$ và $x_j(t_2)$ nhận cùng giá trị (hoặc A, hoặc -A). Còn trường hợp b) xảy ra khi t_1 và t_2 rơi vào khoảng để $x_i(t_1)$ và $x_j(t_2)$ nhận các giá trị khác nhau. Theo giả thiết $p(x_1) = p(x_2) = 1/2$, cho nên:

$$R_X(t_1,t_2) = E\left[x_i(t_1).x_j(t_2)\right] = A^2\frac{1}{2} - A^2.\frac{1}{2} = 0.$$

• Khi $\tau = |t_2 - t_1| < T$. Trong khoảng này $x_i(t_1)$ và $x_j(t_2)$ lại phụ thuộc lẫn nhau. Bởi vì, trong khoảng như thế cả $x_i(t_1)$ và $x_j(t_2)$ nhận cùng giá trị (hoặc A, hoặc -A), cũng tức là giá trị của $x_j(t_2)$ phụ thuộc vào giá trị $x_i(t_1)$ đã xuất hiện trước đó . Bài toán dẫn đến xác định xác suất giao của hai sự kiện $x_i(t_1)$ và $x_j(t_2)$. Xác suất này được xác định bằng tỉ số giữa đoạn thẳng giao nhau của $x_i(t_1)$ và $x_j(t_2)$ với độ dài T, tức là:

$$p[x_i(t_1).x_j(t_2)] = \frac{T - |\tau|}{T}$$

Từ đó chúng ta có:

$$R_X(t_1,t_2) = \sum_{i=1}^{2}\sum_{j=1}^{2} x_i(t_1).x_j(t_2).p\left[x_i(t_1).x_j(t_2)\right] = A^2\left(1 - \frac{|\tau|}{T}\right)$$

Tóm lại, chúng ta xác định được hàm tự tương quan R(τ), dưới dạng:

$$R(t_1,t_2) = R(\tau) = \begin{cases} A^2\left(1 - \dfrac{|\tau|}{T}\right) & |\tau| \le T \\ 0 & |\tau| > T \end{cases} \qquad (5.39)$$

Hàm tự tương quan R(τ) không phụ thuộc vào t_1, t_2, chỉ phụ thuộc vào $\tau = |t_2 - t_1|$. Đồ thị của hàm tự tương quan R(τ) được biểu diễn trên hình 5.13c).

Cuối cùng, chúng ta đã chứng minh được X(t) hội đủ các tiêu chuẩn là QTNN dừng.

Phù hợp theo biến đổi Wierner - Khintchine thuận, được biểu diễn bằng công thức (5.36a), chúng ta xác định được hàm mật độ phổ công suất của X(t), như sau:

$$\begin{aligned} S(f) &= \int_{-\infty}^{+\infty} S(f).\cos(2\pi.f.\tau).df = 2\int_0^T A^2\left(1 - \frac{\tau}{T}\right).\cos(2\pi.f.\tau)df \\ &= A^2.T\frac{\sin^2(\pi.f.T)}{(\pi.f.T)^2} \end{aligned} \qquad (5.40)$$

Để dẫn ra được kết quả (5.40), chúng ta đã sử dụng các kết quả của mục *phổ của xung tam giác* trình bày ở trên.

Đồ thị của hàm mật độ phổ công suất S(f) được biểu diễn trên hình 5.13b). Độ rộng phổ chính, ký hiệu B [Hz], của tín hiệu băng cơ sở này là khoảng tần số F = 0 đến F_{max}= 1/T; tức là B = 1/T. Người ta cũng gọi là *khoảng phổ Nyquist*, vì để truyền nó đòi hỏi tối thiểu một *dải thông Nyquist*, xác định bằng vận tốc bit của tín hiệu - R, với R = 1/T, (định nghĩa này xuất phát từ *định lý lấy mẫu*).

Cần chú ý rằng kết quả xác định hàm mật độ phổ công suất S(f) của mục này rất đáng ghi nhớ, để diễn giải các hàm mật độ phổ công suất - PSD của các tín hiệu điều chế số, trình bày ở chương 6.

CHƯƠNG 6

CÁC TÍN HIỆU ĐIỀU CHẾ

Trong chương này, chúng ta sẽ nghiên cứu các *mô hình toán học* của các tín hiệu điều chế và hàm phổ của chúng. Các sơ đồ điều chế chỉ trình bày trong dạng *nguyên tắc chung*. Còn các vấn đề khác như giải pháp kỹ thuật, cũng như sơ đồ mạch điện cụ thể, không thuộc nội dung của môn học này. Hơn nữa, chúng ta cũng chỉ quan tâm tới vấn đề *điều chế*, còn vấn đề *giải điều chế*, nói chung, chưa đủ điều kiện để trình bày ở đây. Người học, thông thường phải nghiên cứu tới *ba lần*, ở ba năm học khác nhau, vấn đề điều chế và giải điều chế, khi đó mới có đủ kiến thức về vấn đề quan trọng này.

Chương này, chúng ta sẽ nghiên cứu các tín hiệu điều chế tương tự, cũng như các tín hiệu điều chế số. Hầu hết, các tín hiệu điều chế hiện tại đang được sử dụng trong các hệ thống truyền tin (viễn thông, quảng bá, mạng máy tính,...) đều được đề cập ở đây, mặc dù mức độ nông, sâu có thể khác nhau.

1. KHÁI NIỆM CHUNG VỀ ĐIỀU CHẾ

Các tín hiệu băng cơ sở, dù ở dạng tương tự, hay ở dạng số, nói chung, không có khả năng thích hợp với các *dạng kênh truyền khác nhau*. Tuy nhiên, trong một chừng mực nào đấy, đường dây đôi, cáp đồng trục, hay nói chung là các dạng đường truyền có *hai mối liên hệ*, có thể truyền tín hiệu băng cơ sở với một khoảng cách ngắn. Năng lượng của chúng sẽ bị suy giảm nhanh theo khoảng cách. Vì thế, để tín hiệu băng cơ sở thích hợp được với các dạng kênh truyền khác nhau và có thể truyền đi xa, phải thực hiện *điều chế* chúng. Điều chế cũng cho phép truyền *tín hiệu từ nhiều nguồn tin khác nhau* lên cùng một kênh truyền (trước đó có phép *dồn kênh*).

Để điều chế, bắt buộc phải có *hai thành phần*:

- *Tín hiệu băng cơ sở* đem điều chế, thường được ký hiệu là s(t) (hay x(t), w(t), m(t),..., tùy hoàn cảnh). Tín hiệu đem điều chế chứa đựng thông tin (tin tức). Vì thế, nó là quá trình ngẫu nhiên.
- *Dao động mang*, hay còn gọi là dao động tải tin. Đó là một dao động tần số cao, chúng ta thường trình bày dưới dạng điện áp (hay dòng điện), như sau:

$$u_c(t) = U_0 \cos(2\pi f_c t + \theta) \tag{6.1}$$

trong đó U_0 - biên độ của dao động mang, khi chưa điều chế là hằng số đối với thời gian; f_c - tần số của dao động mang, [Hz]; θ - gọi là pha đầu, [rad]; còn $2\pi f_c t + \theta = \phi(t)$ - gọi là pha tức thời của dao động mang, [rad]. Ký hiệu c- carrier- mang. Từ các khái niệm trên, chúng ta thấy rằng mối quan hệ giữa tần số và pha tức thời của dao động, được xác định:

$$f(t) = \frac{1}{2\pi}\frac{\partial\phi(t)}{\partial t} \tag{6.2}$$

ở đây chúng ta đề cập đến quan hệ tần số nói chung, chứ không chỉ riêng cho tần số sóng mang. Còn ngược lại, mối quan hệ giữa pha tức thời của dao động với tần số của nó, được xác định:

$$\phi(t) = 2\pi \int f(t).dt + \theta \qquad (6.3)$$

Như vậy, dao động mang được đặc trưng bởi *ba tham số* - biên độ, tần số, pha tức thời.

Điều chế là cách thức ghi tín hiệu đem điều chế vào một trong các tham số của dao động mang (hoặc ghi đồng thời vào hai tham số), làm cho các tham số này của dao động mang thay đổi theo quy luật của tin tức.

Nếu đem *ghi* tín hiệu đem điều chế vào *biên độ* của dao động mang, thì biên độ của dao động mang, lúc này, sẽ thay đổi theo quy luật của tin tức. Chúng ta có *tín hiệu điều chế biên độ*. Tùy theo tín hiệu đem điều chế là tương tự hay số và tùy theo các xử lý tiếp theo vào phổ tần số của chúng, mà chúng ta sẽ có *nhiều dạng khác nhau* của tín hiệu điều chế biên độ.

Nếu đem *ghi* tín hiệu đem điều chế vào *tần số* của dao động mang, thì tần số của dao động mang, lúc này, sẽ thay đổi theo quy luật của tin tức - gọi là *tần số tức thời*. Chúng ta có *tín hiệu điều chế tần số*. Tùy theo tín hiệu đem điều chế là tương tự hay số và tùy theo các xử lý khác vào s(t) trước điều chế, mà chúng ta sẽ có *nhiều dạng khác nhau* của tín hiệu điều chế tần số.

Nếu đem *ghi* tín hiệu đem điều chế vào *pha tức thời* của dao động mang, thì pha tức thời của dao động mang, lúc này, bên cạnh sự thay đổi đều đặn theo thời gian, còn bị thay đổi theo quy luật của tin tức. Chúng ta có *tín hiệu điều chế pha*. Tùy theo tín hiệu đem điều chế là tương tự hay số và tùy theo các xử lý khác vào s(t) trước điều chế (gọi chung là *xử lý tín hiệu băng cơ sở*), mà chúng ta sẽ có *nhiều dạng khác nhau* của tín hiệu điều chế pha.

Nếu đem *ghi* tín hiệu đem điều chế *đồng thời* vào hai tham số (biên độ và pha) của dao động mang, chúng ta có tín hiệu điều chế hỗn hợp.

Các phương pháp *xử lý tín hiệu băng cơ sở như: song song*, hoặc *trải phổ* tín hiệu số trước điều chế, cho phép chúng ta tạo ra các tín hiệu điều chế mới thích hợp hơn cho mục đích sử dụng.

Điều chế tín hiệu cũng có thể hiểu là phép *mã hóa đặc biệt*, hoặc phép *xử lý đặc biệt*, để gửi tin tức vào vật mang, mà nhờ đó tin tức được truyền đến người sử dụng.

2. TÍN HIỆU ĐIỀU CHẾ BIÊN ĐỘ

Chúng ta nhắc lại, điều chế biên độ là đem *ghi* tín hiệu đem điều chế vào biên độ của dao động mang. Hơn nữa, cách thức ghi, ở dạng này theo *phép cộng*, ở dạng khác lại theo *phép nhân*. Chúng ta bắt đầu nghiên cứu tín hiệu điều chế biên độ từ dạng truyền thống rồi tiếp đến các dạng phức tạp hơn.

2.1. Điều biên bằng tín hiệu tương tự (AM)

Trong trường hợp này, tín hiệu đem điều chế là tín hiệu tương tự. Thuật ngữ tiếng Anh gọi dạng điều chế này là *Amplitude Modulation* - AM. Phép điều chế được thực hiện dưới dạng

phép cộng, tức là đem cộng tín hiệu đem điều chế vào biên bộ của sóng mang. Chúng ta bắt đầu nghiên cứu từ đơn giản đến phức tạp đối với tín hiệu đem điều chế.

a). Tín hiệu đem điều chế có dạng đơn âm (a tone)

Trong trường hợp này, s(t) có dạng:

$$s(t) = U_{\Omega}.\cos(2\pi Ft) \tag{6.4}$$

trong đó $\Omega = 2\pi F$ - ký hiệu tần số góc [rad/sec] của tín hiệu đem điều chế, còn F - ký hiệu tần số của tín hiệu đem điều chế [Hz], hoặc [1/sec]. Tín hiệu đem điều chế chỉ có một thành phần tần số, *không phải là tín hiệu thực*, nó chỉ là dạng toán học để đơn giản trong nghiên cứu.

Còn dao động mang biểu diễn dưới dạng:

$$u_c(t) = U_0 \cos(2\pi f_c t + \theta) \tag{6.5}$$

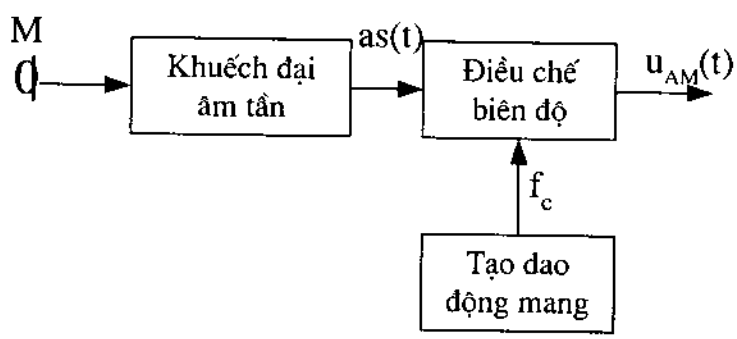
Cách thức điều chế là phép cộng, tức là:

$$u_{AM}(t) = [U_o + a.s(t)].\cos(2\pi f_c t + \theta) \tag{6.6}$$

trong đó a - hệ số tỉ lệ, đặc trưng cho hệ số khuếch đại của khối khuếch đại âm tần, như hình 6.1. Hơn nữa, phải chọn giá trị a thích hợp, để đạt được các chỉ số điều chế khác nhau.

Đem thay thế biểu thức ở (6.4) vào biểu thức (6.6), chúng ta được:

$$\begin{aligned} u_{AM}(t) &= [U_0 + a.U_{\Omega}.\cos(2\pi Ft)].\cos(2\pi f_c t + \theta) \\ &= U_0\left[1 + \frac{aU_{\Omega}}{U_0}.\cos(2\pi Ft)\right].\cos(2\pi f_c t + \theta) \end{aligned} \tag{6.7}$$



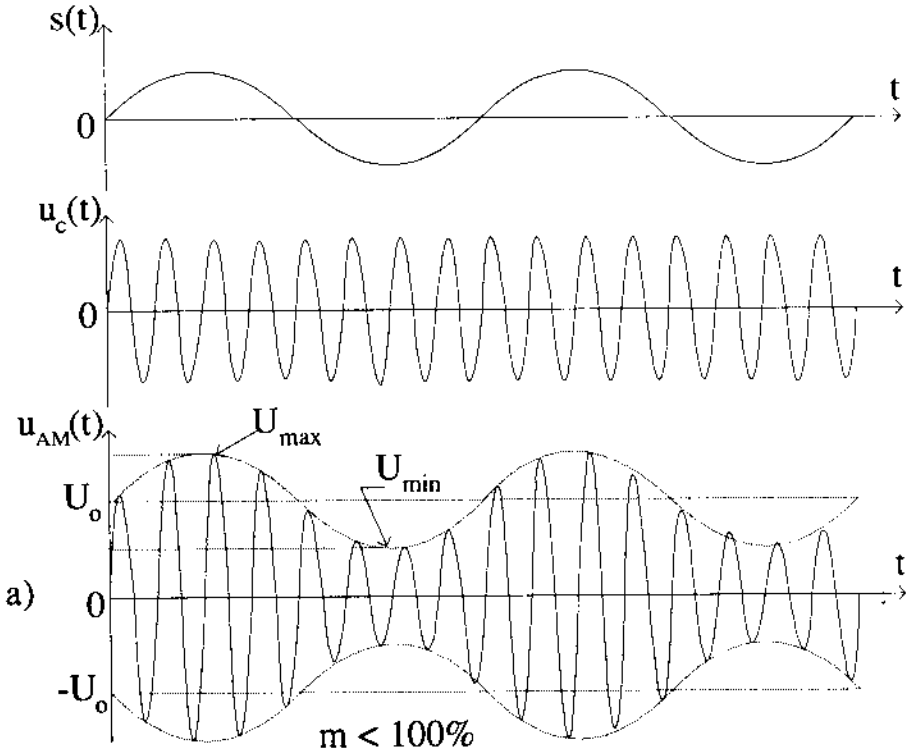
Hình 6.1. Sơ đồ khối điều chế biên độ bằng tín hiệu tiếng nói.

Chúng ta đặt $m = \dfrac{aU_{\Omega}}{U_0}$ (6.7) - gọi là *độ sâu điều chế*, hay còn gọi là *chỉ số điều chế*, thường biểu diễn trong dạng %. Độ sâu điều chế m đặc trưng cho mức độ "nông" hoặc "sâu" của việc ghi tín hiệu đem điều chế vào biên độ của dao động mang. Chúng ta tiếp tục có:

$$\begin{aligned} u_{AM}(t) &= U_0[1 + m.\cos(2\pi Ft)].\cos(2\pi f_c t + \theta) \\ &= [U_0 + mU_0\cos(2\pi Ft)].\cos(2\pi f_c t + \theta) \end{aligned} \tag{6.8}$$

Biểu thức (6.8) gọi là biểu thức của tín hiệu điều biên, trong đó m - là độ sâu điều chế, cũng là độ sâu điều chế toàn bộ, có một giá trị nào đó, hoặc lớn hơn, hoặc bằng không.

Để trực quan khi nghiên cứu tín hiệu điều chế, chúng ta sẽ biểu diễn bằng đồ thị các biểu thức (6.4), (6.5) và (6.8). Riêng biểu thức (6.8) chúng ta biểu diễn trong ba trường hợp, tương ứng với độ sâu điều chế m < 100%, m = 100%, và m > 100%, như hình 6.2.



Hình 6.2. Đồ thị tín hiệu điều biên, khi m < 100% a),

Khi biểu diễn đồ thị theo biểu thức (6.4), chúng ta vẽ theo dạng:

$$s(t) = U_\Omega \sin(2\pi Ft) \tag{6.4a}$$

còn đồ thị theo biểu thức (6.5) theo dạng:

$$u_c(t) = U_0 \sin(2\pi f_c t) \tag{6.5a}$$

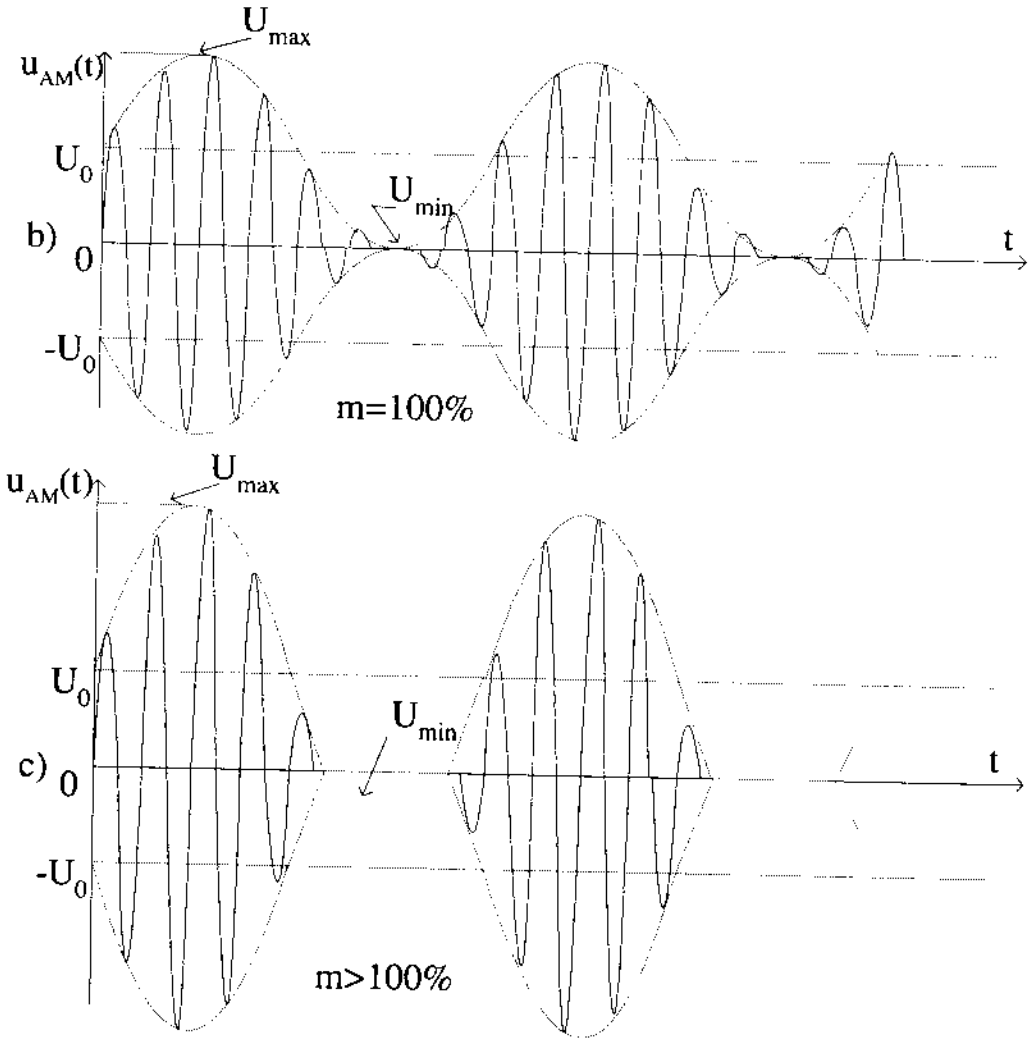
với góc pha đầu $\theta = 0$.

Ký hiệu biên độ cực đại của tín hiệu điều biên là U_{max}, biên độ cực tiểu của nó là U_{min}, còn U_0 là mức đường bao tín hiệu AM trong điều kiện không điều chế, hay biên độ của sóng mang (xem hình 6.2a)), độ sâu điều chế m được xác định:

$$m = \frac{U_{max} - U_{min}}{U_{max} + U_{min}}, \text{ hoặc trong dạng phần trăm } m(\%) = \frac{U_{max} - U_{min}}{U_{max} + U_{min}} \times 100\% \tag{6.9}$$

Định nghĩa theo (6.9) chỉ có giá trị đối với dạng sóng điều chế hình sine. Biểu thức (6.9) có thể suy ra từ biểu thức định nghĩa (6.7), bởi vì chúng ta có:

$$aU_{\Omega} = \frac{U_{max} - U_{min}}{2}, \text{ còn } U_0 = \frac{U_{max} + U_{min}}{2}$$



Hình 6.2. Đồ thị tín hiệu điều biên khi m = 100% b), và m > 100% c).

Khi biên độ tín hiệu đem điều chế $aU_{\Omega} < U_0$, chúng ta có đồ thị tín hiệu điều biên như hình 6.2a). Tiếp tục tăng biên độ của tín hiệu đem điều chế cho đến khi $aU_{\Omega} = U_0$, chúng ta có đồ thị tín hiệu điều biên như hình 6.2b). Lúc này $U_{min} = 0$. Tiếp tục tăng biên độ của tín hiệu đem điều chế cho đến khi $aU_{\Omega} > U_0$, chúng ta thấy biên độ của tín hiệu điều biên tiếp tục tăng, nhưng sẽ xuất hiện các khoảng dạng sóng bị phá vỡ, có tính chu kỳ [5]. Trong khoảng dạng sóng bị phá vỡ do nguyên nhân giao thoa của hai sóng có cùng biên độ, nhưng ngược pha nhau, kết quả dẫn đến sự triệt tiêu. Đồ thị tín hiệu điều biên, trong trường hợp này, như hình 6.2c). Lúc này $U_{min} < 0$.

Thông thường, độ sâu điều chế được xác định với điều kiện $U_{min} \geq 0$, (xem hình 6.2a,b) - gọi là điều chế thích hợp, trong trường hợp này, giá trị của m nằm trong giới hạn $0 \leq m \leq 1$, hoặc trong dạng %, giá trị của m nằm trong giới hạn $0\% \leq m \leq 100\%$. Còn khi $U_{min} < 0$, được gọi là *điều chế sâu (overmodulation)*, như hình 6.2c), trong trường hợp này, giá trị của m nằm trong giới hạn $m > 100\%$. Khi đó giá trị U_{min} không được xác định, vì thế người ta cũng thường đưa vào các định nghĩa khác về độ sâu điều chế:

- % điều chế dương:

$$m+(\%) = \frac{U_{max} - U_0}{U_0} \times 100\%$$

- % điều chế âm:

$$m-(\%) = \frac{U_0 - U_{min}}{U_0} \times 100\%$$

- % điều chế toàn bộ:

$$m \pm (\%) = \frac{U_{max} - U_{min}}{2.U_0} \times 100\%$$

Nếu m+ = m− = m±, chúng ta có dạng *điều chế đối xứng*, trong trường hợp này biên độ của dao động mang U_0, không bị thay đổi bởi điều chế. Khi đó, chúng ta cũng có m+ = m− = m± = m. Vì thế, m gọi là độ sâu điều chế toàn bộ. Giá trị của m nằm trong khoảng $0\% \leq m \leq 100\%$.

Dạng điều chế gọi là *không đối xứng*, nếu m+ ≠ m− ≠ m±. Nguyên nhân của tình trạng này là do đặc tuyến điều chế không tuyến tính (khi mức biên độ của tín hiệu đem điều chế quá cao). Trong trường hợp *điều chế sâu* thì độ sâu điều chế m+ > 100%.

Các *độ sâu điều chế m* mà chúng ta xét sau này chỉ hạn chế với dạng điều chế đối xứng.

Để giải thích vấn đề, trong thực tiễn kỹ thuật thường sử dụng giá trị độ sâu điều chế m nào. Chúng ta cần đề cập tới vấn đề *giải điều chế*, tức là vấn đề tách tin tức s(t) ra khỏi tín hiệu điều biên ở thiết bị thu. Hơn nữa, trong trường hợp này, chúng ta giả thiết tín hiệu thu, trong trường hợp lý tưởng, không chịu tác động của nhiễu (tức $n(t) \equiv 0$).

Với tín hiệu điều chế biên độ, có thể giải điều chế theo *hai phương pháp*:

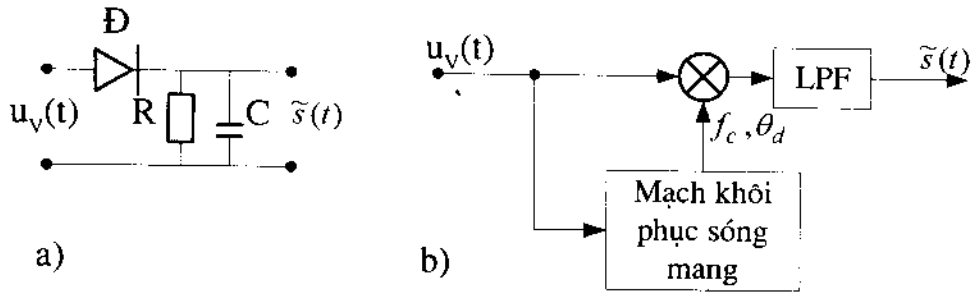
- *Giải điều chế không liên kết*, hay giải điều chế bằng *mạch tách sóng đường bao biên độ (Envelope Detection)*. Mạch tách sóng điển hình cho trường hợp này, trình bày trên hình 6.3a).
- *Giải điều chế liên kết (Coherent Detection)*, hay *giải điều chế tích*. Lúc này cần có mạch khôi phục sóng mang (những hiểu biết sâu hơn về mạch khôi phục sóng mang, độc giả tham khảo ở tài liệu khác, chẳng hạn [11]), để khôi phục chính xác tần số sóng mang f_c và cả trễ pha θ_d của tín hiệu thu. Nguyên tắc của mạch giải điều chế tích, trình bày trên hình 6.3b).

α). Giải điều chế bằng mạch tách sóng đường bao biên độ

Một mạch tách sóng đường bao lý tưởng là mạch điện tạo ra dạng sóng đầu ra tỉ lệ với đường bao của tín hiệu đầu vào. Tín hiệu đầu vào của mạch tách sóng, phù hợp theo quy ước khi vẽ đồ thị của hình 6.2, và với giả thiết tín hiệu thu không nhiễu, có dạng:

$$u_v(t) = [U_0 + mU_0 \sin(2\pi Ft)]\sin(2\pi f_c t + \theta_d) \qquad (6.10)$$

trong đó $mU_0 \sin(2\pi Ft)$ - là đường bao biên độ, còn θ_d - độ trễ pha của tín hiệu thu so với tín hiệu phát.



Hình 6.3. Mạch tách sóng đường bao biên độ a), và mạch tách sóng tích b).

Diễn giải cách thức làm việc của mạch tách sóng đường bao với tín hiệu điều chế biên độ trong hai trường hợp, khi độ sâu điều chế m < 100% và khi độ sâu điều chế m > 100%, được biểu diễn trên hình 6.4.

Trong cả hai trường hợp (m < 100% và m > 100%), do tác dụng của phần tử chỉnh lưu Đ, chỉ có nửa chu kỳ dương của tín hiệu đầu vào truyền được qua điốt, như hình 6.4 a) và b).

Phần tử điện trở R và tụ điện C của mạch tách sóng đường bao biên độ, đóng vai trò là mạch lọc thông thấp (LPF). Hằng số thời gian RC của mạch lọc được chọn sao cho tín hiệu đầu ra sẽ khôi phục lại *đường bao* của tín hiệu đầu vào. Bởi vậy, tần số cắt của LPF phải nhỏ hơn nhiều tần số sóng mang f_c, nhưng phải lớn hơn nhiều dải thông B của tín hiệu đem điều chế (trong trường hợp đơn giản này B ≡ F), tức là:

$$B << \frac{1}{2\pi RC} << f_c$$

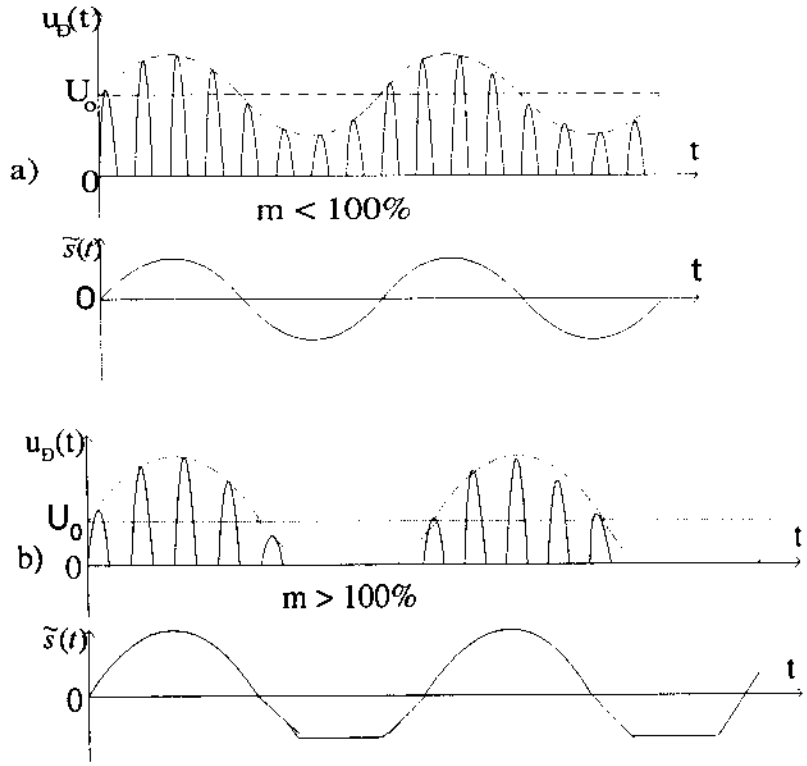
trong đó RC gọi là hằng số thời gian của mạch lọc.

*Với độ sâu điều chế m < 100%, như biểu diễn trên hình 6.4a), mạch tách sóng đường bao biên độ trong trường hợp lý tưởng sẽ cho ra tín hiệu, mà chúng ta gọi là đánh giá của s(t), có dạng:

$$\tilde{s}(t) = K.U_\Omega \sin(2\pi Ft) \qquad (6.10)$$

trong đó K là hằng số tỉ lệ. Rõ ràng, đánh giá của s(t) ở biểu thức (6.10) trùng với s(t) ở biểu thức (6.4a), tức là $\tilde{s}(t) \equiv s(t)$.

*Với độ sâu điều chế m > 100%, như biểu diễn trên hình 6.4b), mạch tách sóng đường bao biên độ trong trường hợp lý tưởng sẽ cho ra tín hiệu, mà chúng ta gọi là đánh giá của s(t), tỉ lệ với biên độ của tín hiệu đầu vào. Trong trường hợp này $\tilde{s}(t) \neq s(t)$.



Hình 6.4. Diễn giải làm việc của mạch tách sóng đường bao với tín hiệu AM khi m < 100% a), và m > 100% b).

Rõ ràng, với việc sử dụng mạch giải điều chế bằng tách sóng đường bao biên độ, việc khôi phục tin tức chỉ chính xác chỉ trong trường hợp độ sâu điều chế m < 100%. Với độ sâu điều chế m > 100%, tin tức khôi phục bị méo dạng và như sau đây sẽ thấy phổ của tín hiệu AM trong trường m > 100% sẽ mở rộng ra nhiều lần.

β). Giải điều chế tích

Sơ đồ nguyên tắc của mạch giải điều chế tích, trình bày trên hình 6.3b). Chúng ta giả thiết rằng tín hiệu đầu vào, trong trường hợp lý tưởng, như biểu thức (6.10), hơn nữa mạch khôi phục sóng mang, khôi phục được chính xác tần số sóng mang f_c, cũng như trễ pha θ_d. Lúc đó, tại mạch nhân diễn ra phép nhân sau:

$$u_v(t) \times U_0 \sin(2\pi f_c t + \theta_d) = U_0[1 + m\sin(2\pi Ft)]\sin(2\pi f_c t + \theta_d) \times U_0 \sin(2\pi f_c t + \theta_d)$$

$$= U_0[1 + m\sin(2\pi Ft)] \times \frac{1}{2}\cos 0 - U_0[1 + m\sin(2\pi Ft] \times \frac{1}{2}\cos(4\pi f_c t + 2\theta_d)$$

Khi truyền qua LPF, thành phần thứ 2 của biểu thức trên- là thành phần cao tần, có tần số bằng $2f_c$, sẽ bị LPF gạt bỏ, chỉ còn lại thành phần thứ nhất. Tức là nhờ mạch nhân và LPF, mà đầu ra của mạch giải điều chế khôi phục được chính xác tin tức ban đầu:

$$\tilde{s}(t) = KU_{\Omega}\sin(2\pi Ft) = s(t)$$

trong đó K- hằng số tỉ lệ.

Nhược điểm căn bản của mạch *giải điều chế tích* là phức tạp, vì thế giá thành sẽ khá cao, không thích hợp cho chức năng xây dựng thiết bị quảng bá.

b). Tín hiệu đem điều chế có dạng đa âm (multipe tone)

Trong trường hợp này, tín hiệu đem điều chế có dạng:

$$s(t) = \sum_{k=1}^{n} U_{\Omega k}\cos(2\pi F_k t) \qquad (6.11)$$

Biểu thức (6.11) biểu diễn một *dạng sóng thực*, bao gồm n thành phần tần số, cũng như trước đây, để đơn giản, chúng ta coi pha đầu của s(t) bằng không.

Tín hiệu điều biên có dạng:

$$u_{AM}(t) = \left[U_0 + \sum_{k=1}^{n} a_k.U_{\Omega k}\cos(2\pi F_k t)\right].\cos(2\pi f_c t + \theta) \qquad (6.12)$$

Cũng như ở mục α), chúng ta ký hiệu $m_k = \dfrac{a_k U_{\Omega k}}{U_0}$ - độ sâu điều chế của thành phần thứ k (k =1, 2,..., n). Để giải điều chế bằng mạch tách sóng đường bao biên độ, khôi phục lại chính xác tin tức ban đầu, chúng ta cũng cần điều kiện các $m_k < 100\%$. Chúng ta cũng định nghĩa độ sâu điều chế toàn bộ, bởi biểu thức:

$$m = \frac{1}{U_0}\sqrt{a_1^2U_{\Omega 1}^2 + a_2^2U_{\Omega 2}^2 + ... + a_n^2U_{\Omega n}^2}$$

hay

$$m = \sqrt{m_1^2 + m_2^2 + ... + m_n^2} \qquad (6.13)$$

Để giải điều chế bằng mạch tách sóng đường bao biên độ, khôi phục lại chính xác tin tức ban đầu, chúng ta cũng cần điều kiện độ sâu điều chế toàn bộ $m < 100\%$

2.2. Phổ biên độ-tần số của tín hiệu AM

Để xây dựng phổ biên độ-tần số của tín hiệu AM, chúng ta cũng bắt đầu từ trường hợp đơn giản nhất, rồi mở rộng cho trường hợp chung.

a). Phổ biên độ - tần số của tín hiệu AM đơn âm

Trong trường hợp này, biểu thức của tín hiệu AM là biểu thức (6.8). Chúng ta có thể triển khai biểu thức (6.8) như sau:

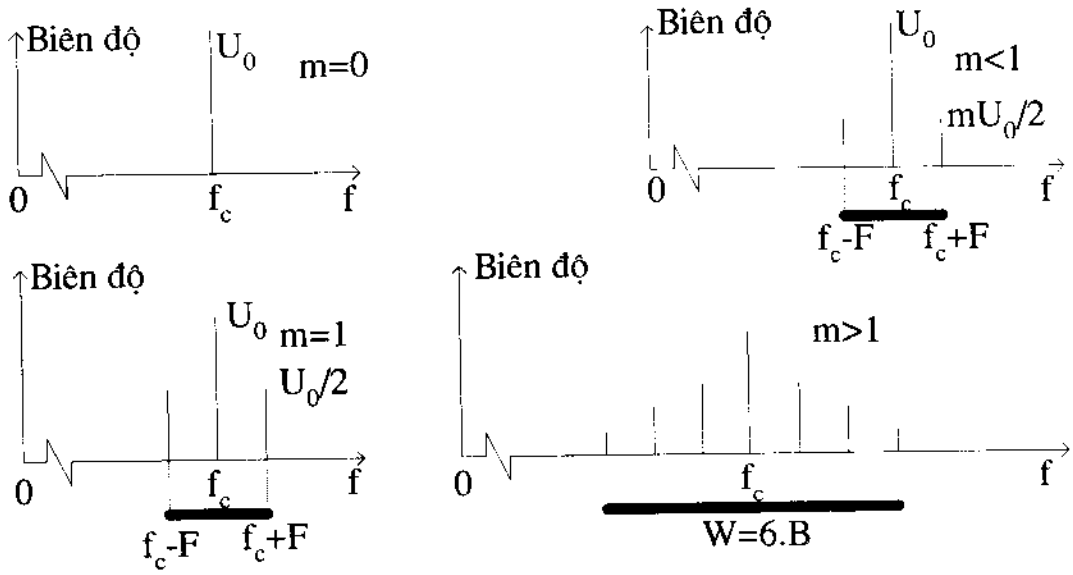
$$u_{AM}(t) = U_0\cos(2\pi f_c t + \theta) + mU_0\cos(2\pi f_c t + \theta)\times\cos(2\pi Ft)$$

$$= U_0\cos(2\pi f_c t+\theta)+\frac{mU_0}{2}\cos[2\pi(f_c+F)t+\theta]+\frac{mU_0}{2}\cos[2\pi(f_c-F)t+\theta] \quad (6.14)$$

Biểu thức (6.14) có ba thành phần:

- Thành phần thứ nhất có biên độ U_0, còn tần số f_c - gọi là thành phần tần số *sóng mang*;
- Thành phần thứ hai có biên độ $mU_0/2$, còn tần số $(f_c + F)$ - gọi là thành phần tần số *biên trên*;
- Thành phần thứ ba có biên độ $mU_0/2$, còn tần số $(f_c - F)$ - gọi là thành phần tần số *biên dưới*.

Căn cứ vào nhận xét trên, chúng ta có thể vẽ được đồ thị phổ biên độ-tần số của tín hiệu AM đơn âm, với các giá trị độ sâu điều chế m khác nhau, như hình 6.5



Hình 6.5. Phổ biên độ-tần số của tín hiệu AM đơn âm, với ba giá trị của m.

Từ hình 6.5, chúng ta thấy rằng biên độ của thành phần sóng mang, U_0, và độ rộng phổ tần số, W, sẽ không đổi, nếu thực hiện điều biên với $m \le 1$. Công suất của các thành phần tần phổ, trong trường hợp này, được xác định như công suất trên một điện trở tải 1Ω, bằng các biểu thức sau:

- Công suất của thành phần sóng mang:

$$P_c = U_0^2$$

- Công suất của thành phần biên:

$$P_b = (\frac{mU_0}{2})^2$$

Công suất toàn bộ của tín hiệu AM là tổng công suất thành phần sóng mang và hai thành phần biên, được xác định như sau:

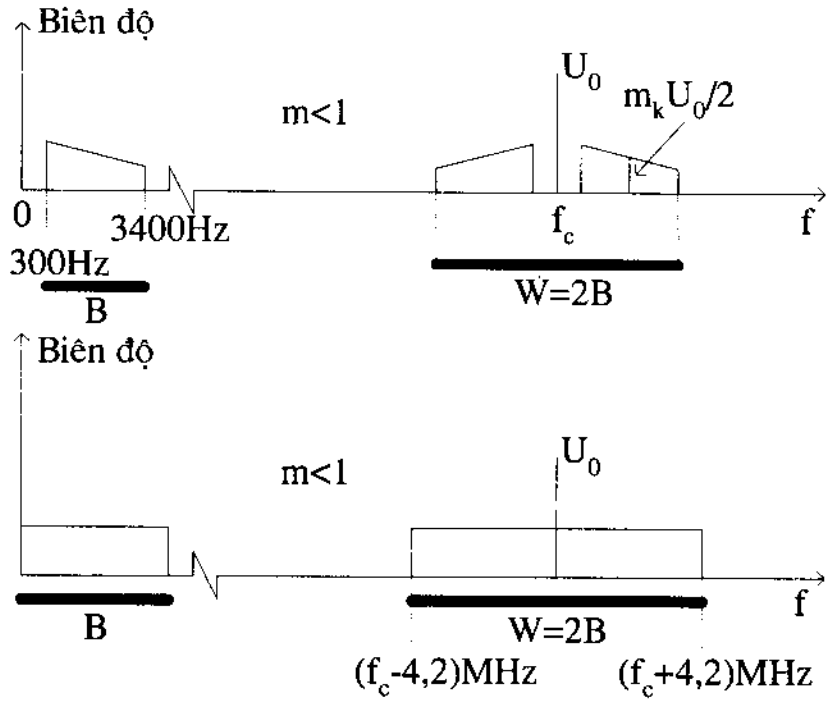
$$P_{AM} = P_c + 2P_b = P_c\left(1+\frac{m^2}{2}\right) \quad (6.15)$$

Từ biểu thức (6.15), chúng ta thấy rằng nếu độ sâu điều chế m = 1, thì hơn 66% công suất của tín hiệu dành cho thành phần sóng mang, chỉ có gần 17% công suất của tín hiệu để tải thông tin (một dải biên). Nếu độ sâu điều chế m < 1, thì việc phân phối công suất để tải thông tin còn giảm hơn nữa, chẳng hạn, m = 1/2, thì gần 90% công suất tín hiệu dành cho thành phần sóng mang, chỉ có hơn 5% công suất của tín hiệu để tải thông tin.

Cũng từ biểu thức (6.15), chúng ta thấy rằng khi điều chế sâu, tức là độ sâu điều chế m > 1, thì công suất của thành phần sóng mang giảm đi. Nhưng độ rộng phổ của tín hiệu AM tăng lên nhiều lần, chẳng hạn, $W_{m>1} = 6B = 3W_{m<1}$, ở đây W[Hz] - ký hiệu độ rộng phổ tần số của tín hiệu AM, còn B[Hz] - ký hiệu độ rộng phổ của tin tức (trong trường hợp đơn giản này, B = F).

b). Phổ biên độ-tần số của tín hiệu AM đa âm

Phổ biên độ-tần số của tín hiệu AM đa âm, có thể suy ra từ phổ biên độ tần số của tín hiệu AM đơn âm vừa nghiên cứu ở trên.



Hình 6.6. Phổ biên độ-tần số của tín hiệu AM tiếng nói và hình ảnh.

Cần chú ý rằng, trong trường hợp này, tín hiệu đem điều chế có n thành phần tần số. Mỗi thành phần tần số có độ sâu điều chế của riêng mình, được ký hiệu là m_k. Vì thế các vạch phổ biên có biên độ xác định theo công thức: $\left(\frac{m_k U_0}{2}\right)^2$. Trên hình 6.6, chúng ta biểu diễn phổ biên độ-tần số của tín hiệu AM, với tin tức đem điều chế là tín hiệu tiếng nói có độ rộng phổ B = (300 ÷ 3400)Hz và tín hiệu hình ảnh có độ rộng phổ B = (0 ÷ 4,2) MHz.

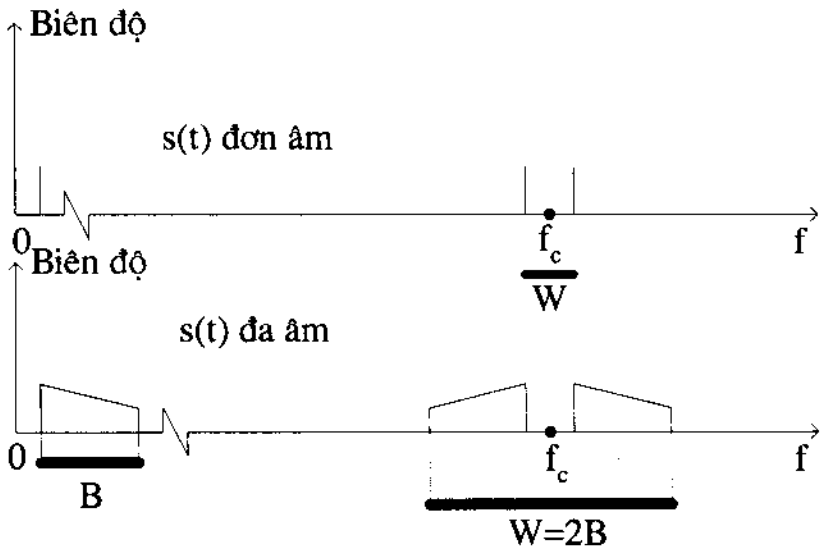
Với tín hiệu tiếng nói, sau khi điều chế biên độ, nhận được tín hiệu AM có khoảng chiến dụng phổ W = 2B = 2×3400 = 6.800Hz. Còn với tín hiệu hình ảnh, sau khi điều chế biên độ, nhận được tín hiệu AM có khoảng chiến dụng phổ W = 2B = 2× 4,2 = 8,4MHz. Như vậy, khoảng chiếm dụng phổ tín hiệu AM hình ảnh quá rộng. Để giảm khoảng chiếm dụng phổ, như dưới đây sẽ nghiên cứu, người ta sử dụng mạch lọc dải thông đặc biệt-*lọc dải biên rớt*- để cắt cụt phần lớn độ rộng phổ của biên tần dưới, sao cho bề rộng phổ vừa khớp với dải thông kênh truyền 6MHz (kể cả phổ của tín hiệu âm thanh).

2.3. Điều chế biên độ với sóng mang bị triệt

Trong một vài ứng dụng thực tiễn, người ta cần tạo ra tín hiệu điều chế biên độ với sóng mang bị triệt. Một tín hiệu AM có hai dải biên tần với sóng mang bị triệt (Double SideBand Supperssed Carrier- DSB-SC), được tạo ra bằng cách nhân tín hiệu đem điều chế s(t) với sóng mang, với điều kiện s(t) không có thành phần một chiều, như sau:

$$u_{DSB-SC}(t) = U_0 s(t)\cos(2\pi f_c t + \theta) \qquad (6.16)$$

Khi đem so sánh tín hiệu DSB-SC với tín hiệu AM, thì phần trăm điều chế là là vô hạn, vì không có thành phần sóng mang. Toàn bộ công suất của tín hiệu dùng để tải thông tin (mỗi dải biên 50%). Tuy nhiên, khi giải điều chế cần sử dụng mạch giải điều chế tích, vì thế giá thành sẽ cao hơn nhiều một mạch tách sóng đường bao biên độ.



Hình 6.7. Phổ biên độ-tần số của tín hiệu DSB – SC.

Phổ biên độ - tần số của tín hiệu DSB - SC chỉ có hai dải biên. Để giải thích điều này, chúng ta lại bắt đầu với trường hợp đơn giản nhất, tín hiệu đem điều chế chỉ có một thành phần tần số (a tone). Lúc đó, biểu thức (6.16) có thể viết đơn giản như sau:

$$u_{DSB-SC}(t) = U_0 U_\Omega \cos(2\pi F t) \times \cos(2\pi f_c t + \theta)$$

Chúng ta triển khai tiếp, như sau:

$$u_{DSB-SC}(t) = A\cos[2\pi(f_c + F)t + \theta]\cos[2\pi(f_c - F)t + \theta] \qquad (6.17)$$

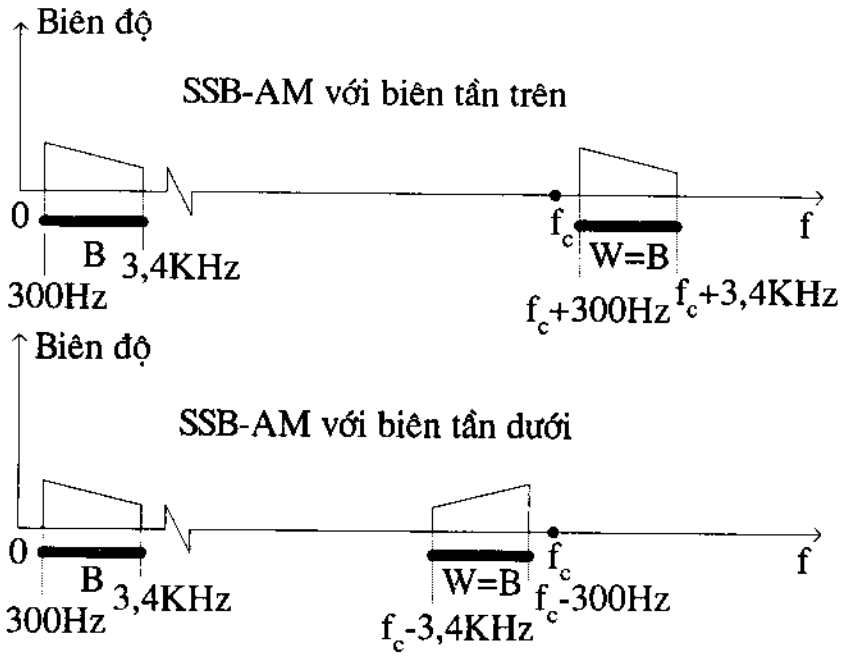
Rõ ràng biểu thức (6.17) chỉ có hai thành phần tần số là $f_c \pm F$. Vì thế, phổ biên độ tần số, khi xây dựng, xuất phát từ biểu thức (6.17), cũng như từ biểu thức (6.16) với s(t) có nhiều thành phần tần số, được biểu diễn trên hình 6.7.

Nếu s(t) là tín hiệu số, thì phép nhân (6.16), kết quả cho tín hiệu điều chế pha hai mức, mà chúng sẽ xét sau này.

2.4. Điều chế biên độ - đơn biên

Khi nghiên cứu phổ biên độ-tần số của tín hiệu AM, chúng ta thấy rằng phổ của nó có hai dải biên (biên trên và biên dưới) cùng chứa một lượng thông tin như nhau, hay nói một cách, chỉ cần một dải biên cũng có thể khôi phục lại tin tức ban đầu, tức là có thể bỏ đi một dải biên.

Chúng ta cũng nhận thấy về mặt phân bố công suất của tín hiệu AM cho các thành phần tần số rất không đều nhau. Vì thế, để toàn bộ công suất của tín hiệu có thể sử dụng cho truyền tải thông tin, trong phổ biên độ-tần số cần loại bỏ đi thành phần số sóng mang.



Hình 6.8. Phổ biên độ- tần số của tín hiệu SSB - AM tiếng nói.

Kết quả, chúng ta có tín hiệu điều chế biên độ đơn biên (Single SideBand Amplitude Modulation - SSB-AM). Tín hiệu SSB - AM với sóng mang bị triệt thường được sử dụng trong thực tiễn nhất.

Tín hiệu SSB - AM có thể thực hiện bằng một trong ba giải pháp kỹ thuật sau:

- Phương pháp lọc (filtering method);
- Phương pháp quay pha (phasing method);
- Phương pháp lai ghép của hai phương pháp trên.

Phổ biên độ tần số của tín hiệu SSB - AM, khi sử dụng để điều chế tín hiệu tiếng nói, như hình 6.8.

Từ hình 6.8. chúng ta thấy rằng SSB - AM cho kết quả như là dịch chuyển, hoặc dịch chuyển kết hợp với đảo của phổ tín hiệu tiếng nói. Biên độ của tín hiệu tỉ lệ với biên độ của tín hiệu tiếng nói. Vì không có thành phần tần số sóng mang, cho nên không định nghĩa độ sâu điều chế. Toàn bộ công suất của tín hiệu dùng để truyền tải thông tin.

So với tín hiệu AM, tín hiệu SSB-AM có chiến dụng phổ chỉ bằng một nửa, tức $W_{SSB\text{-}AM} = B$. Tuy nhiên, giải pháp kỹ thuật ở cả phía phát, cũng như phía thu, đều phức tạp hơn. Vì thế, tín hiệu SSB - AM, chủ yếu chỉ thích hợp sử dụng cho mục đích quân sự.

2.5. Điều chế biên độ, với dải biên rớt

Trong kỹ thuật truyền hình quảng bá, để truyền tải tín hiệu hình ảnh, nếu thực hiện điều biên-AM, hoặc DSB - SC, thì phổ tần số của nó quá rộng, như chúng ta đã thấy trên hình 6.6. Nếu sử dụng SSB-AM, thì giải pháp kỹ thuật quá phức tạp, dẫn đến giá thành sẽ đắt. Vì thế, người ta sử dụng giải pháp *điều biên rớt* (Vestigial SideBand - VSB). Tín hiệu VSB có thể thực hiện từ tín hiệu AM (hoặc DSB - SC), nhờ một mạch lọc dải thông đặc biệt- gọi là *mạch lọc biên rớt*. Mạch lọc biên rớt có tác dụng làm suy giảm (cắt cụt) phần lớn phổ biên tần dưới. Hàm truyền của mạch lọc biên rớt, cũng như diễn giải tác dụng của nó được trình bày trên hình 6.9.

Tín hiệu VSB được biểu diễn bằng biểu thức:

$$s_{VSB}(t) = u_{AM}(t) * h(t) \tag{6.18}$$

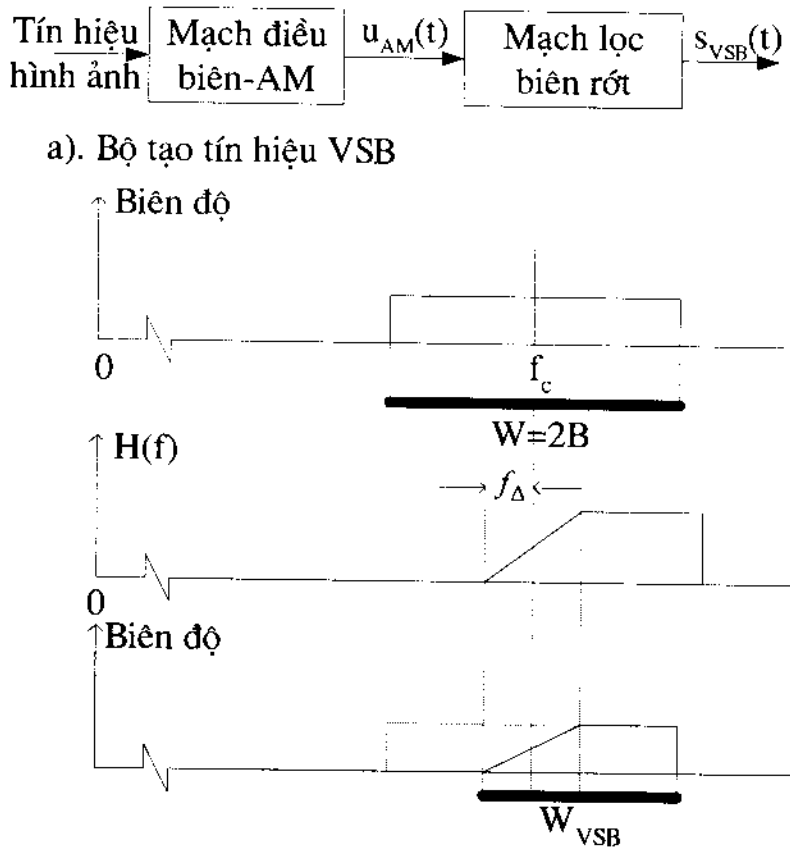
trong đó $u_{AM}(t)$ - tín hiệu AM, h(t)- đáp ứng xung của mạch lọc biên rớt, dấu *- chỉ phép tích chập.

Trong miền tần số, phổ của tín hiệu VSB được xác định bằng biểu thức:

$$S_{VSB}(f) = S_{AM}(f).H(f) \tag{6.19}$$

trong đó $S_{AM}(f)$ - phổ biên độ- tần số của tín hiệu AM, còn $H(f)$ hàm truyền của mạch lọc biên rớt.

Hàm truyền của mạch lọc biên rớt H(f) có đặc tính bằng phẳng trong khoảng tần số: $f_c + f_\Delta \le f \le F_{\max}$, trong đó F_{max}- là giới hạn tần số trên của mạch lọc, $F_{\max} > B$, với B độ rộng tần số của tín hiệu hình ảnh và cắt vát trong khoảng tần số: $f_c - f_\Delta \le f < f_c + f_\Delta$. Tham số f_Δ, trong thực tiễn được chọn bằng 0,75MHz.



Hình 6.9. Bộ tạo tín hiệu VSB a) và quá trình hình thành phổ VSB: b) phổ tín hiệu AM, c) hàm truyền mạch lọc biên rớt, d) phổ tín hiệu VSB.

Để giải điều chế tín hiệu VSB thường sử dụng mạch tách sóng đường bao biên độ (cũng có thể sử dụng mạch tách sóng tích).

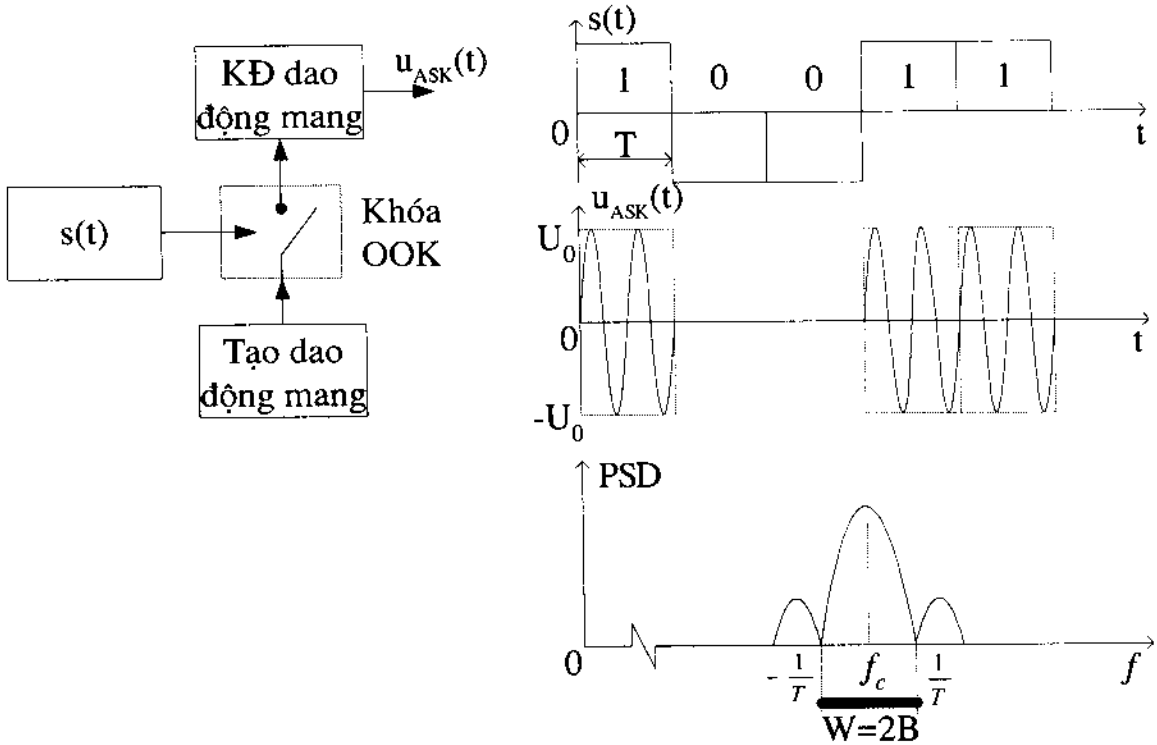
2.6. Điều chế biên độ bằng tín hiệu số (ASK)

Khi thực hiện điều chế biên độ với tín hiệu đem điều chế là tín hiệu số, thuật ngữ tiếng Anh gọi là *Amplitude Shift Keying-ASK-Khóa dịch chuyển biên độ*. Tín hiệu ASK nhận được bằng cách nhân tín hiệu đem điều chế s(t) với sóng mang, nhờ sơ đồ mạch như hình 6.10. Lúc đó, tín hiệu ASK biểu diễn bởi biểu thức:

$$u_{ASK}(t) = U_0 s(t)\cos(2\pi f_c t + \theta) = \begin{cases} U_0\cos(2\pi f_c t + \theta) & s(t) = 1 \\ 0 & s(t) = 0 \end{cases} \quad (6.20)$$

Sơ đồ mạch nhân như hình 6.10, được gọi là sơ đồ *khóa ON- OFF*- khóa Bật-Tắt. Khi s(t) = 1, khóa K ở vị trí đóng- ON, dao động mang được truyền đến khối khuếch đại sóng mang, để khuếch đại hoặc biên độ, hoặc công suất. Khi s(t) = 0, khóa K ở vị trí ngắt- OFF, dao động mang không được nối ra ngoài. Kết quả ở đầu ra không có tín hiệu.

Dạng sóng của tín hiệu ASK được biểu diễn trên hình 6.10.



Hình 6.10. Sơ đồ khối mạch tạo tín hiệu ASK, hình sóng và hàm PSD của nó.

Khi xác định hàm mật độ phổ công suất- PSD của tín hiệu ASK, chúng ta giả thiết rằng s(t) là một thể hiện của QTNN dừng, có các đặc tính như đã xét trong mục 6 chương 5. Vì thế hàm PSD xác định bởi biểu thức:

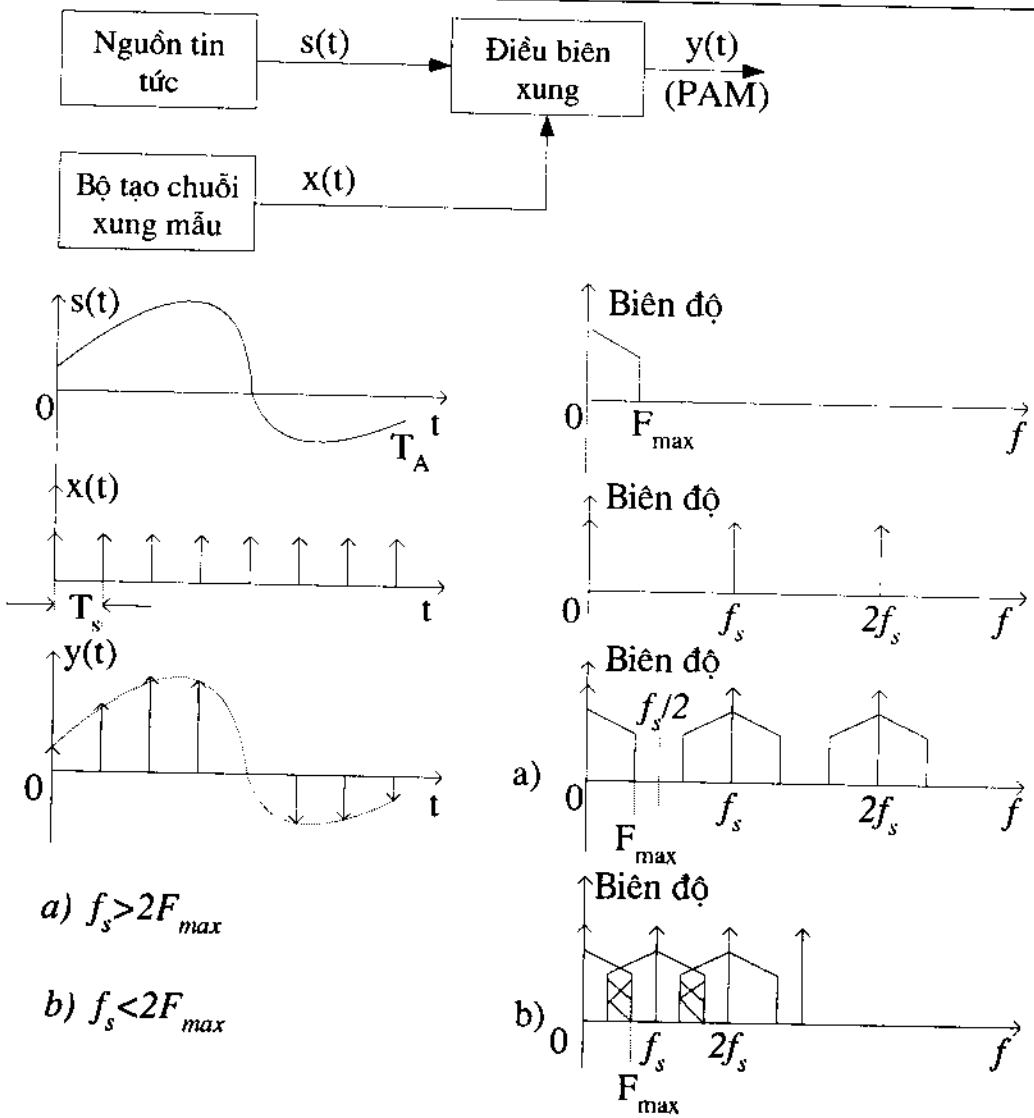
$$S(2\pi f) = U_0^2 T \frac{\sin^2(\pi f_c T)}{(\pi f_c T)^2} \tag{6.21}$$

Đồ thị hàm PSD được biểu diễn trên hình 6.10 (dưới cùng). Khoảng chiếm dụng phổ tần số của tín hiệu ASK bằng 2B.

Sơ đồ khóa ON- OFF được sử dụng điển hình trong thực tiễn là bộ phát *laser* tạo tín hiệu ánh sáng ở mạch *chuyển đổi điện/quang*, trong card mạng thông tin quang. Trong đó, mạch tạo dao động mang là *bộ phát laser bán dẫn*, còn tín hiệu s(t) đóng vai trò *nguồn bơm*. Khi s(t) = 1, *bộ phát laser bán dẫn* đủ điều kiện dao động và phát ra *xung ánh sáng*, còn khi s(t) = 0, *bộ phát laser bán dẫn* không đủ điều kiện dao động, nó "tạo" ra bóng tối.

2.7. Điều chế biên độ xung (PAM)

Trong mục này, chúng ta nghiên cứu vấn đề *điều chế biên độ xung* (Pulse Amplitude Modulation-PAM) bằng tín hiệu tương tự, chẳng hạn, tín hiệu tiếng nói, với mục đích để trình bày *định lý lấy mẫu* (định lý Harry-Nyquist, hay còn gọi là định lý Kachennhicốp).



Hình 6.11. Sơ đồ khối mạch tạo tín hiệu PAM, hình sóng và phổ của nó.

Trên hình 6.11, ở trên cùng, trình bày sơ đồ khối mạch điện điều chế biên độ xung- PAM. Trong đó s(t) là tín hiệu tương tự, có phổ tần số hữu hạn, tức là phổ tần số xác định trong khoảng $f = (0 \div F_{max})$, với các thành phần tần số $f > F_{max}$ sẽ bỏ qua. s(t) được gọi là *tín hiệu đem lấy mẫu*. x(t)-được gọi là chuỗi xung mẫu, trong trường hợp lý tưởng, nó là *chuỗi xung delta-Dirac* - $\delta(t)$. Chu kỳ của chuỗi xung mẫu ký hiệu T_s [sec]- được gọi là chu kỳ lấy mẫu, còn nghịch đảo của nó là tần số lấy mẫu- $f_s = \frac{1}{T_s}$ [Hz]. Trên hình 6.11, ở bên trái, biểu diễn các tín hiệu s(t), x(t) và y(t) *trong miền thời gian*. Cũng trên hình 6.11, ở bên phải, biểu diễn các tín hiệu s(t), x(t) và y(t) *trong miền tần số*, trong hai trường hợp:

a). Khi tần số lấy mẫu f_s lớn hơn (hoặc bằng) hai lần tần số cực đại F_{max} của tín hiệu đem lấy mẫu, tức là $f_s \geq 2F_{max}$ (6.22). Lúc đó, phổ tần số của tín hiệu PAM không xảy ra hiện tượng "*chồng phổ*", vì thế khi khôi phục nhờ một mạch lọc thông thấp có tần số cắt thích hợp ($F_{max} \leq F_c < f_s/2$), hoàn toàn khôi phục được dạng sóng ban đầu.

b). Khi tần số lấy mẫu f_s nhỏ hơn hai lần tần số cực đại F_{max} của tín hiệu lấy mẫu, tức là $f_s < 2F_{max}$ (6.23). Lúc đó, phổ tần số của tín hiệu PAM xảy ra hiện tượng "*chồng phổ*", hay còn gọi là "*canh phổ*"- *aliasing*. Vì thế, với mạch lọc thông thấp không có thể khôi phục lại chính xác dạng sóng ban đầu.

Điều kiện: $f_s \geq 2F_{max}$ (6.22a) được gọi là điều kiện của định lý lấy mẫu.

3. CÁC TÍN HIỆU ĐIỀU CHẾ TẦN SỐ

Chúng ta nhắc lại rằng điều chế tần số là cách thức ghi tín hiệu đem điều chế s(t) vào tần số của dao động mang, làm cho tần số của dao động mang thay đổi theo quy luật của s(t), lúc đó ta có tín hiệu điều chế tần số. Tùy theo, s(t) là tín hiệu tương tự, hay đã số hóa, chúng ta có tín hiệu điều chế tần số khác nhau.

3.1. Điều tần bằng tín hiệu tương tự (FM)

a). Biểu thức của tín hiệu FM

Trước hết, chúng ta nghiên cứu điều chế tần số bằng tín hiệu tương tự, thuật ngữ tiếng Anh gọi là *Frequency Modulation-FM*. Sơ đồ mạch điện như hình 6.12, ở trên cùng. Phần tử cơ bản của mạch điện là VCO - Voltage Controlled Oscillator, được hiểu là: *mạch tạo dao động, mà tần số của nó được điều khiển bởi điện áp ngoài*. Khi đó, tần số tức thời do VCO tạo ra, được xác định bằng biểu thức:

$$f(t) = f_c + as(t) \qquad (6.24)$$

trong đó a - hệ số tỉ lệ. Từ biểu thức (6.24), chúng ta thấy rằng quy luật biến thiên của tần số tức thời $f(t)$ hoàn toàn phù hợp theo quy luật biến thiên của của tín hiệu đem điều chế s(t). Tuy nhiên, do có thêm thành phần f_c, cho nên giá trị trung bình của $f(t)$ chính là f_c, chứ không phải bằng không. Biểu thức pha tức thời của tín hiệu được xác định theo (6.3), có dạng:

$$\Phi(t) = 2\pi \int f(t)dt = 2\pi f_c t + 2\pi a \int s(t)dt + \theta \qquad (6.25)$$

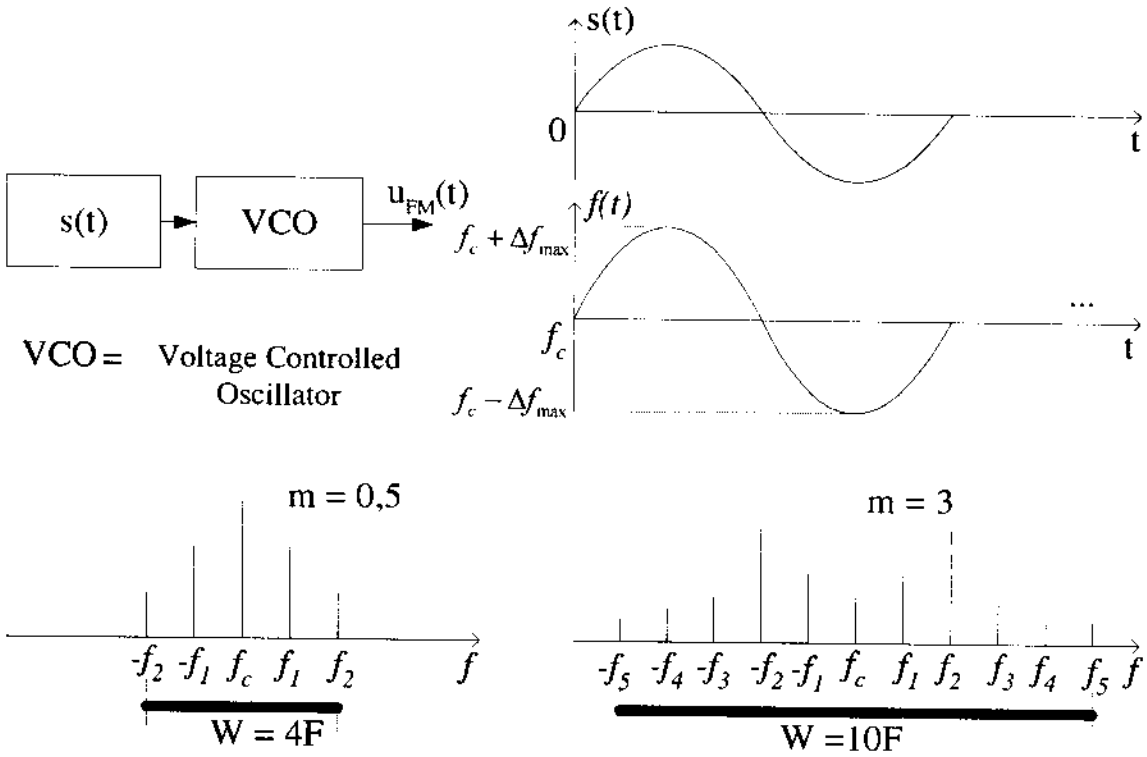
Để đơn giản, chúng ta trước hết, xét tín hiệu FM với s(t) là tín hiệu đơn âm, tức là:

$$s(t) = U_\Omega \cos(2\pi Ft) \qquad (6.26)$$

Khi đó biểu thức pha tức thời của tín hiệu FM có dạng:

$$\Phi(t) = 2\pi f_c t + \frac{aU_\Omega}{F}\sin(2\pi Ft) + \theta \qquad (6.25a)$$

Không mất đi tính tổng quát, khi chúng ta giả thiết rằng góc pha đầu $\theta \equiv 0$.



Hình 6.12. Sơ đồ khối mạch tạo tín hiệu FM, hình sóng và phổ của nó.

Biểu thức của tín hiệu FM, trong trường hợp đơn giản nhất, có dạng:

$$u_{FM}(t) = U_0 \cos\left(2\pi f_c t + \frac{aU_\Omega}{F}\sin(2\pi F t)\right) \qquad (6.27)$$

Chúng ta đặt $m = \dfrac{aU_\Omega}{F} = \dfrac{\Delta f_{max}}{F}$ (6.28),- gọi là *chỉ số điều tần*, trong đó Δf_{max} - gọi là *độ di tần cực đại*, còn F- tần số của tín hiệu đem điều chế. Biểu thức (6.27) viết lại như sau:

$$u_{FM}(t) = U_0 \cos\left(2\pi f_c t + m\sin(2\pi F t)\right) \qquad (6.27a)$$

Dạng sóng của tín hiệu FM, với dạng sóng của tín hiệu đem điều chế s(t) cụ thể, được biểu diễn trên hình 6.15 ở dưới đây.

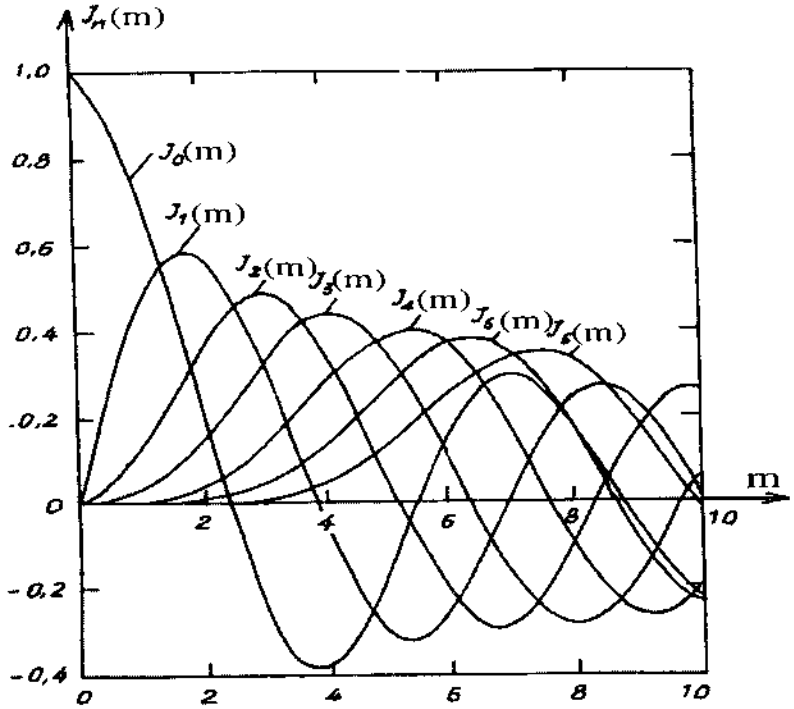
b). Phổ biên độ-tần số của tín hiệu FM

Để xác định phổ biên độ-tần số của tín hiệu FM, người ta thường phân tích biểu thức (6.27a) thành chuỗi các hàm Bessel, như sau:

$$u_{FM}(t) = U_0 \sum_{n=-\infty}^{+\infty} J_n(m)\cos\left[2\pi\left(f_c + nF\right)t\right] \qquad (6.27b)$$

trong đó $J_n(m)$ - *hàm Bessel loại 1, bậc n, đối số m.* Giá trị của các hàm $J_n(m)$ được trình bày dưới dạng bảng, hoặc đồ thị như hình 6.13. Với bậc n nguyên, các hàm Bessel loại 1 có đặc tính:

$$J_{-n}(m) = (-1)^n J_n(m) \tag{6.28}$$



Hình 6.13. Đồ thị các hàm Bessel loại 1 bậc n, đối số m.

Nhờ biểu thức (6.27b), biểu thức (6.28) có dạng mới như sau:

$$u_{FM}(t) = U_0\{J_0(m)\cos(2\pi f_c t) + J_1(m)\cos(2\pi(f_c + F)t - J_1(m)\cos(2\pi(f_c - F)t + J_2(m)\cos(2\pi(f_c + 2F)t) + J_2(m)\cos(2\pi(f_c - 2F)t) + \dots\} \tag{6.27c}$$

Từ biểu thức (6.27c), chúng ta nhận thấy tín hiệu FM có các thành phần sau:

- Thành phần thứ nhất, có biên độ: $U_0 J_0(m)$, tần số f_c - gọi là thành phần *tần số sóng mang*;
- Thành phần thứ hai, có biên độ: $U_0 J_1(m)$, tần số $f_c \pm F$ - gọi là thành phần *hài bậc 1* (có hai hài bậc một);
- Thành phần thứ ba, có biên độ: $U_0 J_2(m)$, tần số $f_c \pm 2F$ - gọi là thành phần *hài bậc 2* (có hai hài bậc hai);
- v.v...

Chúng ta lại nhận thấy rằng biên độ của các thành phần tần phổ *phụ thuộc vào chỉ số điều chế m*. Tùy thuộc vào m, mà biên độ của các thành phần tần phổ có giá trị khác nhau, thậm chí bằng không. Hơn nữa, với chỉ số n càng lớn, nói chung biên độ của các thành phần hài càng nhỏ, có

thể bỏ qua. Trên hình 6.12, phía dưới cùng, biểu diễn phổ biên độ-tần số của tín hiệu FM, với hai giá trị của m là 0,5 và 3. Trong hình vẽ này, chúng ta sử dụng ký hiệu $f_i = f_c + iF$, còn $-f_i = f_c - iF$, với i = 1 , 2, 3,...

Rõ ràng, *khoảng chiếm dụng của phổ tần số* trong hai trường hợp là khác nhau, giá trị của m càng lớn, thì W càng rộng. Tồn tại công thức Carson để xác định khoảng chiếm dụng phổ tần số của tín hiệu FM, như sau:

$$W_{FM} = 2(m+2)B \tag{6.29}$$

trong đó B - bề rộng phổ của tín hiệu băng cơ sở đem điều chế.

Cần chú ý rằng khi tín hiệu đem điều chế s(t) có nhiều thành phần tần số, thì mỗi thành phần tần số cho một chỉ số điều chế m tương ứng, cùng một hình ảnh phổ tương ứng. Phổ chung là *xếp chồng* của các phổ thành phần.

3.2. Điều tần bằng tín hiệu số (FSK)

Sơ đồ tạo tín hiệu khóa dịch chuyển tần số FSK (*Frequency Shift Keying*), cũng tương tư như sơ đồ tạo tín hiệu FM, chỉ có điều khác biệt, ở đây s(t) là tín hiệu nhị phân.

Mạch VCO tạo ra tín hiệu với tần số tức thời có hai giá trị như sau:

$$f(t) = f_c + as(t) = \begin{cases} f_1 = f_c + \Delta f & s(t) \equiv 1 \\ f_0 = f_c - \Delta f & s(t) \equiv 0 \end{cases} \tag{6.30}$$

Biểu thức của tín hiệu FSK với pha liên tục, được xác định như sau:

$$u_{FSK}(t) = \begin{cases} U_0 \cos(2\pi f_1 t + \theta_1) & s(t) \equiv 1 \\ & \theta_1 = \theta_2 \\ U_0 \cos(2\pi f_0 t + \theta_2) & s(t) \equiv 0 \end{cases} \tag{6.31}$$

Tín hiệu FSK với giả thiết như trên, được gọi là tín hiệu khóa dịch chuyển tần số với pha liên tục (*Continuous Phase*- CP-FSK). Ngoài tín hiệu CP- FSK, hay được đề cập, cũng tồn tại tín hiệu FSK với pha không liên tục NCP - FSK.

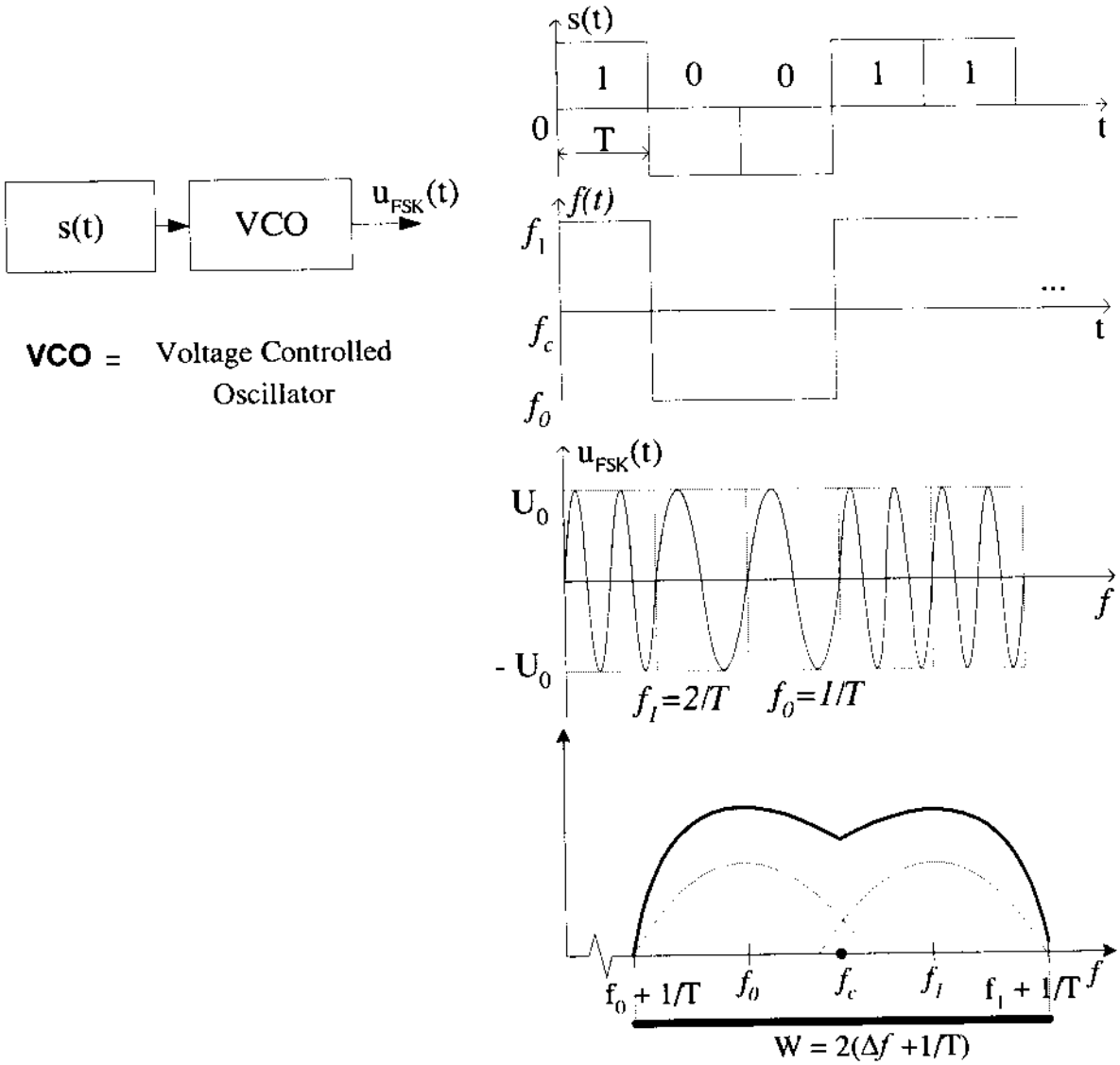
Khi nghiên cứu tín hiệu FSK, người ta cũng định nghĩa *chỉ số điều tần*, như sau:

$$m = \frac{2\Delta f}{1/T} = \frac{2\Delta f}{R} = \frac{2\Delta f}{B} \tag{6.32}$$

trong đó R = 1/T- vận tốc bit, còn B - bề rộng phổ tần số của s(t).

Trên hình 6.14 biểu diễn sơ đồ khối mạch điện tạo tín hiệu FSK, hình sóng của các tín hiệu s(t), f(t) và $u_{FSK}(t)$. Hơn nữa, hình sóng của tín hiệu FSK được vẽ với giả thiết $\theta_1 = \theta_2 = 0$, giá trị

các tần số được xác định như sau: $f_1 = \frac{2}{T}$, $f_0 = \frac{1}{T}$ với T- là thời gian bit của tín hiệu đem điều chế; giả thiết cuối cùng là các sóng có dạng $U_0 \sin(\omega_i t)$.



Hình 6.14. Sơ đồ khối mạch tạo tín hiệu FSK, hình sóng và PSD của nó.

Trên hình 6.14 cũng biểu diễn hàm PSD của tín hiệu FSK. Khi xây dựng hàm PSD chúng ta giả thiết rằng chuỗi bít s(t) là một thể hiện của QTNN dừng, có các đặc tính đã xét ở mục 6 chương 5. Hàm PSD của tín hiệu FSK coi là xếp chồng của hai hàm PSD; một của tín hiệu ASK có tần số mang f_1, còn hàm kia của tín hiệu ASK khác với tần số sóng mang f_0.

Bề rộng phổ tần số FSK được xác định:

$$W_{FSK} = 2(\Delta f + \frac{1}{T}) \tag{6.33}$$

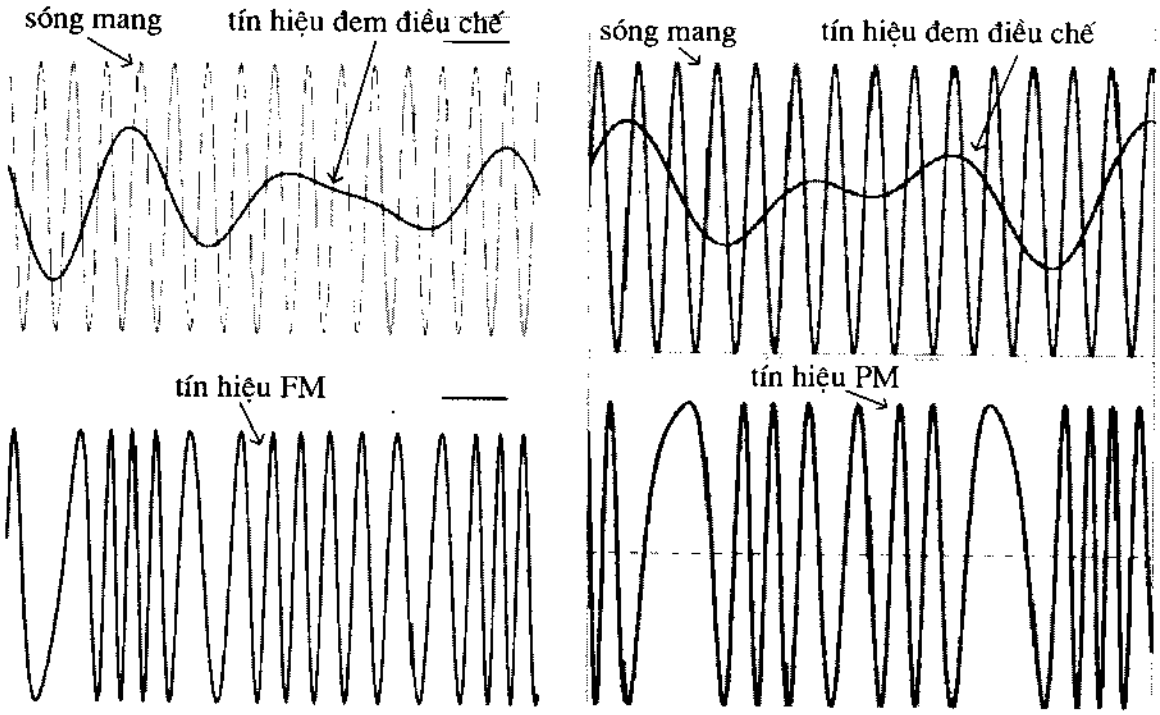
trong đó $\Delta f = f_1 - f_c = f_c - f_0 = \frac{f_1 - f_0}{2}$.

Tùy thuộc vào giá trị của chỉ số điều tần m, mà khoảng chiến dụng của phổ tần số tín hiệu FSK sẽ khác nhau.

4. CÁC TÍN HIỆU ĐIỀU CHẾ PHA

Tín hiệu điều chế pha, cũng như tín hiệu điều tần gọi chung là *tín hiệu điều chế góc*. Chúng cũng có thể phân chia thành hai dạng: điều chế bằng tín hiệu tương tự và điều chế pha bằng tín hiệu số.

4.1. Điều chế pha bằng tín hiệu tương tự (PM)



Hình 6.15. Tín hiệu điều chế tần số (FM) và điều chế pha (PM).

Thuật ngữ tiếng Anh gọi dạng điều chế bằng tín hiệu tương tự là *Phase Modulation-PM*.

Để thấy rõ mối quan hệ giữa điều chế tần số và điều chế pha bằng tín hiệu tương tự, chúng ta so sánh bằng bảng dưới.

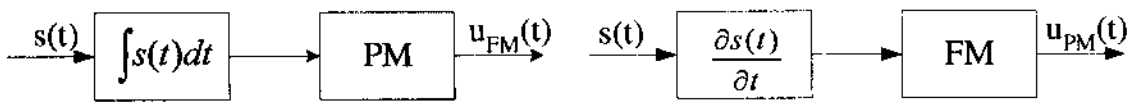
Như vậy, khi điều chế tần số, thì tần số tức thời tỉ lệ trực tiếp với s(t). Còn khi điều chế pha, thì tần số tức thời tỉ lệ với đạo hàm của s(t) theo thời gian.

Từ các mối quan hệ này, chúng ta thấy rằng có thể thực hiện điều chế tần số-FM, gián tiếp thông qua điều chế pha-PM; nếu trước mạch điều chế pha, tín hiệu s(t) được cho qua mạch tích

phân với hệ số truyền thích hợp, như biểu diễn trên hình 6.16a). Ngược lại, có thể thực hiện điều pha- PM, gián tiếp thông qua điều chế tần số- FM; nếu trước mạch điều chế tần số, tín hiệu s(t) được cho qua mạch vi phân với hệ số truyền thích hợp, như biểu diễn trên hình 6.16b).

Điều chế tần số - FM	Điều chế pha - PM
Tần số tức thời $f(t) = f_c + as(t)$ Pha tức thời $\Phi(t) = 2\pi \int f(t)dt$ $\Phi(t) = 2\pi f_c t + 2\pi \int s(t)dt + \theta$	Pha tức thời $\Phi(t) = 2\pi f_c t + as(t) + \theta$ Tần số tức thời $f(t) = \frac{1}{2\pi}\frac{\partial\Phi(t)}{\partial t}$ $f(t) = f_c + \frac{a}{2\pi}\frac{\partial s(t)}{\partial t}$

Để thấy rõ mối quan hệ này, trên hình 6.15 biểu diễn dạng sóng của tín hiệu FM (bên trái), và PM (bên phải) thay đổi phụ thuộc vào tín hiệu đem điều chế s(t) cụ thể.



a). FM thông qua PM b). PM thông qua FM

Hình 6.16. Thực hiện FM thông qua PM a) và ngược lại b).

4.2. Điều chế pha bằng tín hiệu số (PSK)

Phương án đơn giản nhất khi thực hiện điều chế pha bằng tín hiệu số là thực hiện nhân tín hiệu đem điều chế s(t) với dao động mang $u_c(t)$, nhờ một mạch nhân như hình 6.17. Kết quả của phép nhân nhận được tín hiệu với hai mức dịch pha, gọi là *khóa dịch pha hai mức-BPSK (Binary Phase Shift Keying)*.

Biểu thức của tín hiệu BPSK được xác định như sau:

$$u_{BPSK}(t) = U_0 s(t)\cos(2\pi f_c t + \theta) \tag{6.34}$$

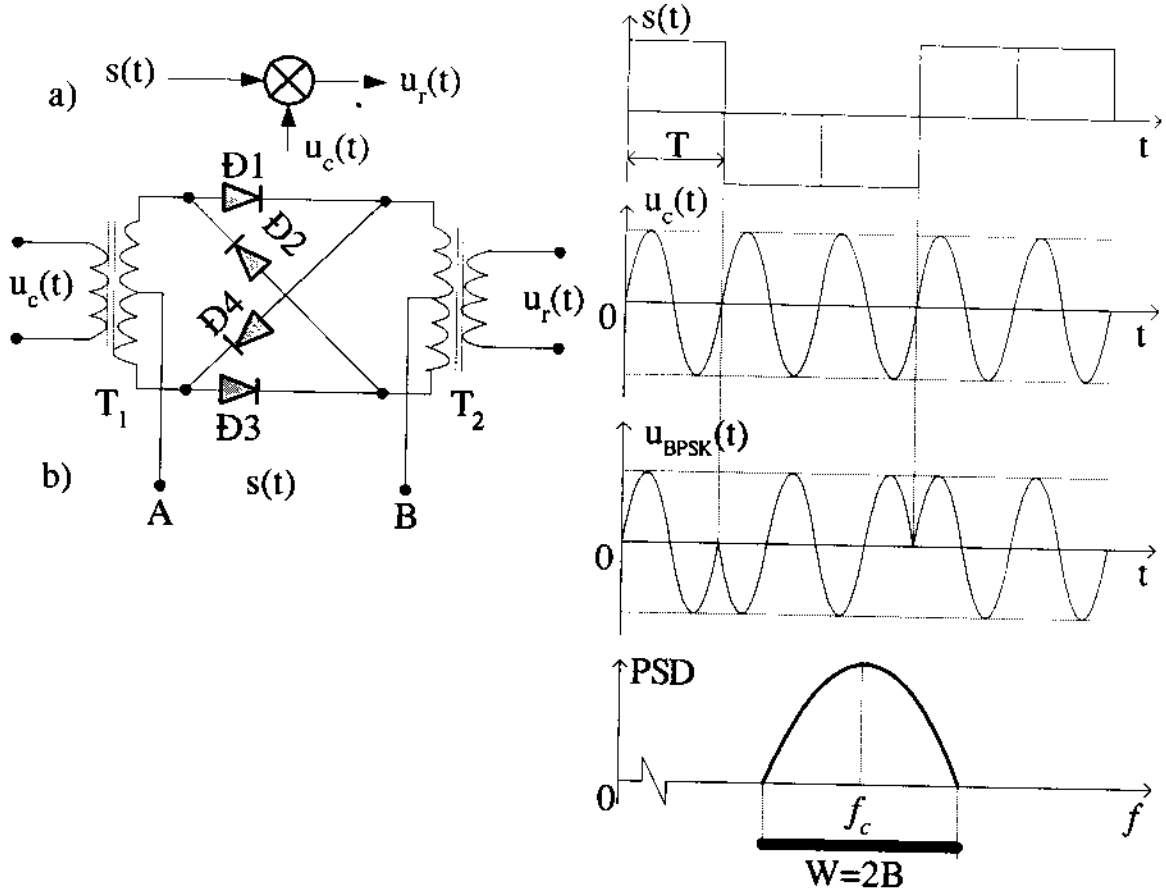
Một mạch nhân thường được sử dụng là *mạch cầu vòng* bao gồm hai biến áp T1, T2, bốn điôt Đ1, Đ2, Đ3, Đ4. Tín hiệu s(t) được nối: một đầu vào điểm giữa của cuộn thứ cấp biến áp T1, đầu kia nối vào điểm giữa của cuộn sơ cấp biến áp T2. Tín hiệu s(t) đóng vai trò tín hiệu *điều khiển đóng, ngắt* các cặp điôt để thay đổi pha dao động mang khi truyền qua mạch.

Khi s(t) = 1, giả thiết rằng điện thế ở điểm A cao hơn điện thế ở điểm B, tức là: $\varphi(A) > \varphi(B)$. Lúc đó, điôt Đ1, Đ3 phân cực thuận, và cho rằng nội trở của chúng bằng không; còn điôt Đ2, Đ4 phân cực ngược, và cho rằng nội trở của chúng bằng vô cùng. Vì thế, dao động mang từ đầu vào (bên trái) truyền thẳng qua mạch, với pha không bị thay đổi.

Khi s(t) = 0, chúng ta cần giả thiết ngược lại, điện thế ở điểm B cao hơn điện thế ở điểm A, tức là: $\varphi(B) > \varphi(A)$. Lúc này, điôt Đ2, Đ4 phân cực thuận, có nội trở bằng không, còn điôt Đ3, Đ4 phân cực ngược, có nội trở bằng vô cùng. Mạch thực hiện truyền theo dạng *bắt chéo*, vì vậy dao động mang từ đầu vào (bên trái), truyền qua mạch với pha bị thay đổi 180^0.

Kết quả, hình thành nên tín hiệu với hai mức dịch pha (0 và π). Trên hình 6.17 biểu diễn dạng sóng của tín hiệu BPSK, với giả thiết dạng sóng hình sine, có góc pha đầu θ = 0, còn tần số mang $f_c = 1/T$. Trên hình 6.17 cũng biểu diễn hàm PSD của tín hiệu BPSK. Khoảng chiến dụng phổ chính (cũng gọi là *khoảng phổ Nyquist*) của tín hiệu BPSK là:

$$W_{BPSK} = 2B \tag{6.35}$$



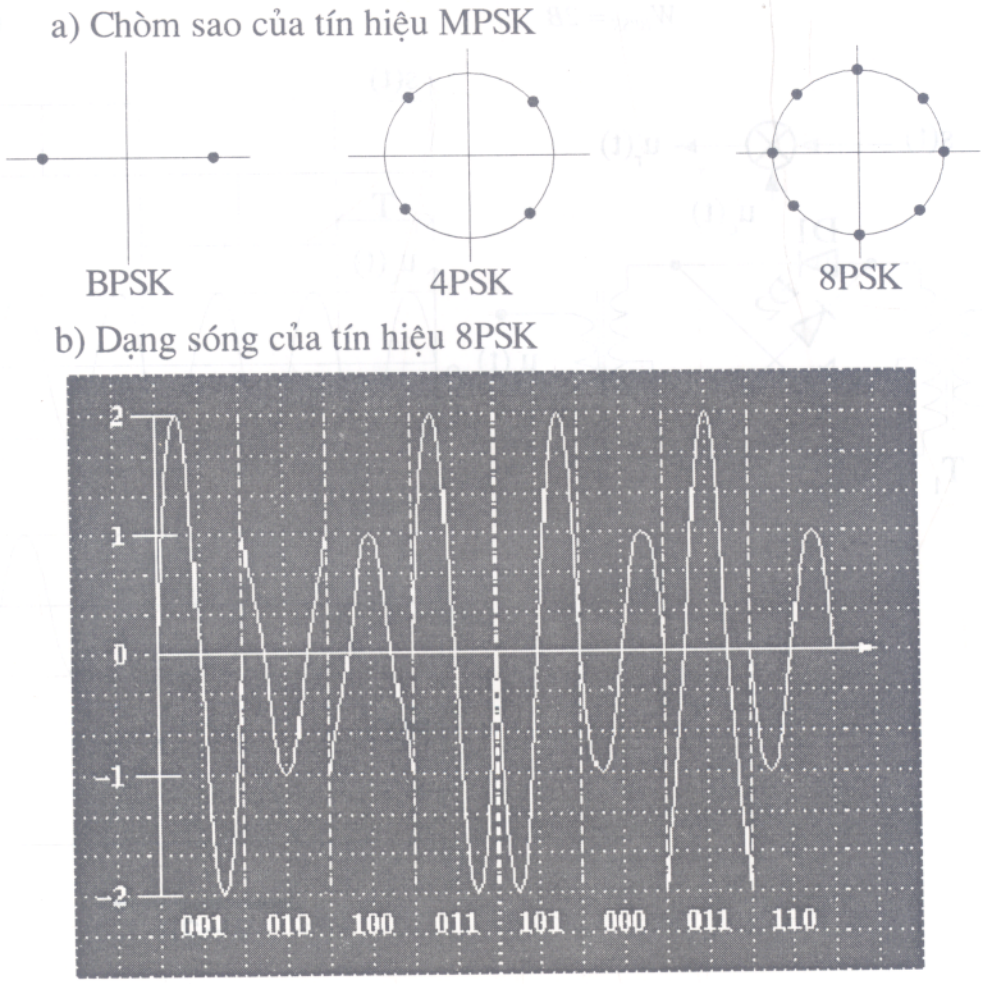
Hình 6.17. Sơ đồ khối mạch tạo tín hiệu BPSK, hình sóng và PSD của nó.

Người ta cũng thực hiện tín hiệu điều chế pha với nhiều mức thay đổi pha, ký hiệu là MPSK, với M = 4, 8, 16,...Khi đó mức dịch pha của hai trạng thái liên tiếp là: 90°, 45°, 22,5°. Muốn vậy, trước khi điều chế cần thực hiện các xử lý tín hiệu băng cơ sở thích hợp và chuyển đổi tín hiệu nhị phân thành tín hiệu L mức biên độ-DAC (*Digital-Analog Converter*) (xem hình 6.20).

Các mạch thực hiện tín hiệu MPSK khá phức tạp, độc giả có thể tham khảo ở các tài liệu khác, chẳng hạn, [4].

Trên hình 6.18 trình bày chòm sao của một vài tín hiệu khóa dịch pha. Cũng trên hình 6.18 (hình dưới cùng) trình bày hình sóng của tín hiệu 8PSK. Tín hiệu này có hai mức biên độ và 8 mức dịch pha. Khoảng chiếm dụng phổ tần số của tín hiệu 8PSK chỉ bằng hai phần ba độ rộng phổ tần số của tín hiệu đem điều chế, tức là:

$$W_{8PSK} = \frac{2}{3}B \tag{6.36}$$



Hình 6.18. Chòm sao của các tín hiệu MPSK và dạng sóng của tín hiệu 8PSK.

5. TÍN HIỆU ĐIỀU CHẾ BIÊN ĐỘ CẦU PHƯƠNG (MQAM)

Các tín hiệu điều chế biên độ cầu phương (*Quadrature Amplitude Modulation - QAM)* là một trong các dạng điều chế tín hiệu tiên tiến nhất hiện nay, vì hiệu suất sử dụng phổ tần số cao của chúng.

Hiệu suất sử dụng phổ tần số của một tín hiệu điều chế được định nghĩa bằng công thức:

$$\eta = \frac{R}{W}\left[\frac{bit/\sec}{Hz}\right] \tag{6.37}$$

Cùng vận tốc bit R của tín hiệu đem điều chế, dạng điều chế nào tạo ra tín hiệu với khoảng chiếm dụng phổ tần số nhỏ nhất, sẽ có hiệu suất sử dụng phổ tần số cao nhất.

Trên hình 6.19a) trình bày sơ đồ khối mạch tạo tín hiệu 4QAM. Mạch bao gồm:

1). Chuyển đổi chuỗi bit s(t) từ dạng nối tiếp thành hai chuỗi *symbol* song song, ký hiệu $s_1(t)$ và $s_2(t)$. Hơn nữa, chuỗi bit s(t) có thời gian bit là T, vận tốc bit là R=1/T. Các chuỗi *symbol* song song, có thời khoảng *symbol* là 2T, vì vậy vận tốc *symbol* là $R_{kh} = \frac{1}{2T} = \frac{R}{2}$. Chuỗi ký hiệu $s_2(t)$ chậm pha so với chuỗi ký hiệu $s_1(t)$ một khoảng thời gian là $T_{kh}/2$. Nhánh trên của sơ đồ (truyền chuỗi ký hiệu $s_1(t)$), được gọi là nhánh I, còn nhánh dưới (truyền chuỗi ký hiệu $s_2(t)$), được gọi là nhánh Q.

2). Mạch DAC: mạch chuyển đổi tín hiệu nhị phân thành tín hiệu L mức biên độ (xem hình 6.20a)). Khi điều chế 4QAM không sử dụng.

Công đoạn 1 và 2 được gọi là công đoạn *xử lý tín hiệu băng cơ sở*. Sơ đồ 4QAM, cũng còn gọi là sơ đồ QPSK, còn nếu với nhánh Q của sơ đồ có mắc thêm bộ trễ $T_{kh}/2$, chúng ta có sơ đồ OQPSK.

3). Mạch nhân chuỗi ký hiệu $s_i(t)$ với dao động mang. Mạch nhân này cũng thường gọi là *mạch điều chế cân bằng*. Dao động mang gồm hai thành phần có pha trực giao nhau, vì thế mạch điều chế có tên *cầu phương (Quadrature)*.

4). Mạch cộng thực hiện phép cộng hai tín hiệu.

Trên hình 6.19b) trình bày hoạt động của mạch chuyển đổi nối tiếp thành song song. Các bit của s(t), chẳng hạn, được đánh số thứ tự 1, 2, 3, 4,...Khi đó các bit có số thứ tự lẻ sẽ chuyển thành các *symbol* của $s_1(t)$ và mở rộng thời khoảng *symbol* thành 2T. Còn các bit có số thứ tự chẵn sẽ chuyển thành các *symbol* của $s_2(t)$ và mở rộng thời khoảng *symbol* thành 2T.

Các mạch *điều chế cân bằng* tạo ra các tín hiệu BPSK tương ứng như sau:

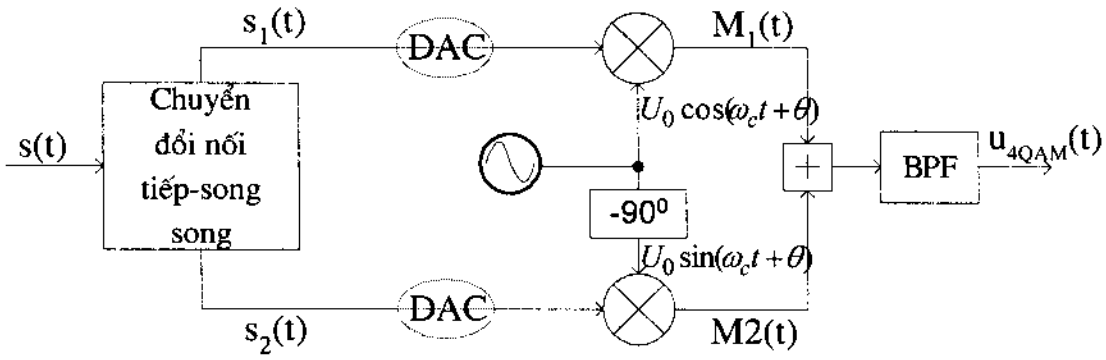
$$M_1(t) = U_0 s_1(t)\cos(2\pi f_c t + \theta)$$

$$M_2(t) = U_0 s_2(t)\sin(2\pi f_c t + \theta)$$

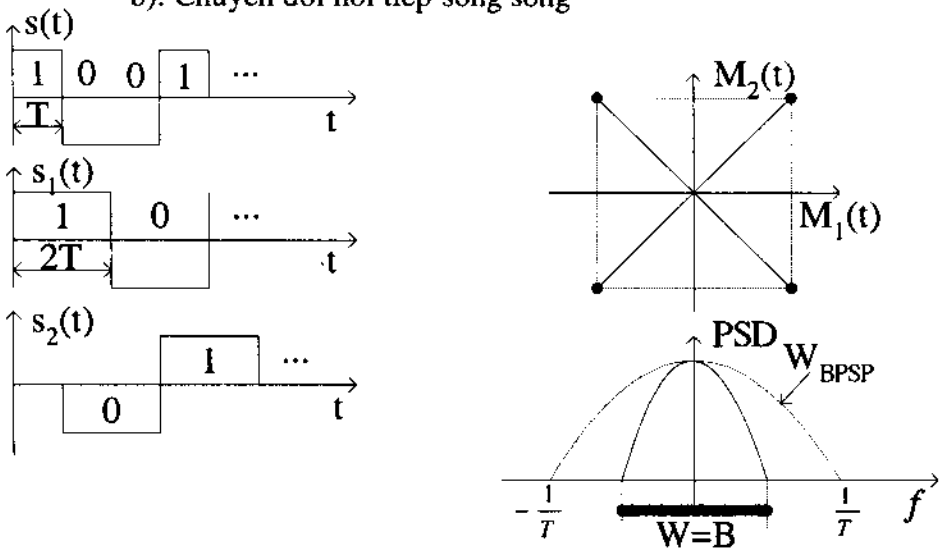
Phép cộng hai véctơ, để tạo thành tín hiệu 4QAM, như sau:

$$\bar{u}_{4QAM}(t) = \bar{x}M_1(t) + \bar{y}M_2(t) = \bar{x}U_0 s_1(t)\cos(2\pi f_c t + \theta) + \bar{y}U_0 s_2(t)\sin(2\pi f_c t + \theta)$$

a). Bộ điều chế 4QAM



b). Chuyển đổi nối tiếp-song song



Hình 6.19. Sơ đồ mạch điều chế 4QAM và PSD của nó.

Kết quả của phép cộng hai véctơ, sẽ tạo ra tín hiệu có 4 mức dịch pha, như trình bày trong bảng dưới:

$s_1(t)$	+	-	-	+
$s_2(t)$	+	+	-	-
$\Delta\varphi$	π/4	3π/4	5π/4	7π/4

Trên hình 6.19, dưới cùng, trình bày hàm PSD của tín hiệu 4QAM. Khoảng chiếm dụng phổ tần số được xác định:

$$W_{4QAM} = \frac{W_0}{2} = B \tag{6.38}$$

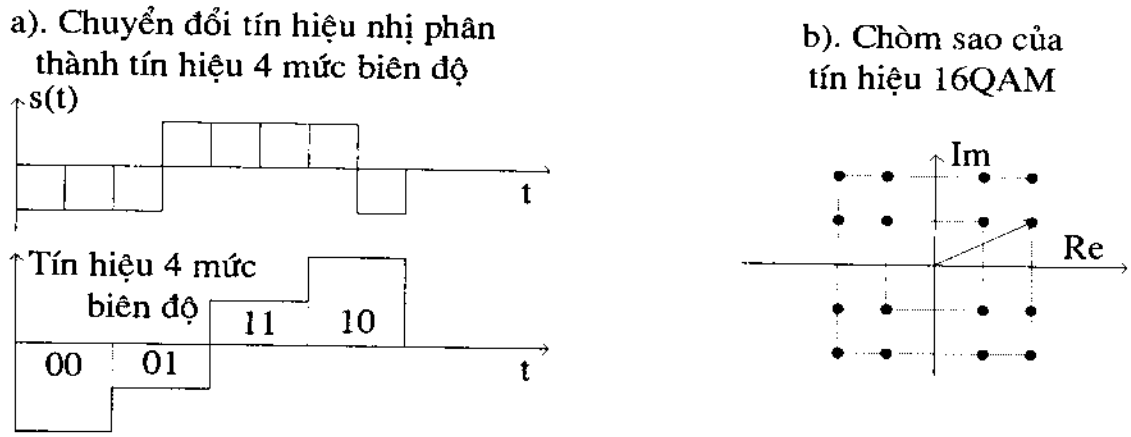
trong đó W_0 được hiểu là *khoảng chiến dụng phổ tần số* của tín hiệu BPSK khi điều chế chuỗi bit s(t) có cùng tốc độ R (hoặc cùng B).

Khi thực hiện điều chế biên độ cầu phương M mức, cần tiến hành trên mỗi nhánh song song các mạch chuyển đổi tín hiệu nhị phân thành tín hiệu L mức biên độ (mạch DAC). Trên hình 6.20a) biểu diễn cách thức chuyển đổi tín hiệu nhị phân thành tín hiệu 4 mức biên độ. Còn hình 6.20b) biểu diễn chòm sao của tín hiệu 16QAM.

Nếu chúng ta ký hiệu mỗi véctơ tín hiệu của tín hiệu MQAM là $\bar{u}(t)$ thì chúng gồm hai thành phần: phần thực- Re và phần ảo- Im như sau:

$$u(t) = x(t) + iy(t) = |u(t)|e^{i\varphi} = \mathrm{Re}[u(t)] + i\,\mathrm{Im}[u(t)] \tag{6.39}$$

trong đó $|u(t)| = \sqrt{[x(t)]^2 + [y(t)]^2}$ - chỉ độ lớn biên độ tín hiệu, còn $\varphi = acrtg\dfrac{[x(t)]}{[y(t)]}$ - chỉ góc lệch pha của tín hiệu.



Hình 6.20. a) Chuyển đổi tín hiệu nhị phân thành tín hiệu 4 mức biên độ
b) Chòm sao của tín hiệu 16QAM.

Các tín hiệu MQAM có *khoảng chiếm dụng phổ tần số* được xác định theo công thức sau:

$$W_{MQAM} = \frac{W_0}{\log_2 \sqrt{M}} \tag{6.40}$$

trong đó $W_0 = 2B$.

Rõ ràng, số mức điều chế M càng lớn, thì *khoảng chiếm dụng phổ tần số* càng nhỏ và hiệu suất sử dụng phổ tần số càng lớn. Tuy nhiên, kèm theo là mức độ phức tạp của giải pháp kỹ thuật và đòi hỏi công nghệ cao phù hợp, cũng như nhạy cảm với pha đỉnh biên độ.

6. KHÓA DỊCH CỰC TIỂU CÓ LỌC GAUSS TRƯỚC ĐIỀU CHẾ (GMSK)

Tín hiệu khóa dịch cực tiểu (Minimum Shift Keying- MSK) thực chất là tín hiệu FSK với pha liên tục (CP- FSK), có chỉ số điều tần cực tiểu m = 0,5 để tạo thành tín hiệu có 2 thành phần trực giao nhau.

Trước hết chúng ta cần chứng minh rằng chỉ số điều tần cực tiểu của tín hiệu CP-FSK là 0,5. Thực vậy, từ biểu thức (6.31) xác định các thành phần của tín hiệu CP-FSK, và từ điều kiện trực giao xét ở mục 2 chương 5, chúng ta có:

$$\int_0^T U_0^2 \cos(2\pi f_1 t + \theta) \times \cos(2\pi f_0 t + \theta) dt = 0$$

triển khai tích phân trên chúng ta có:

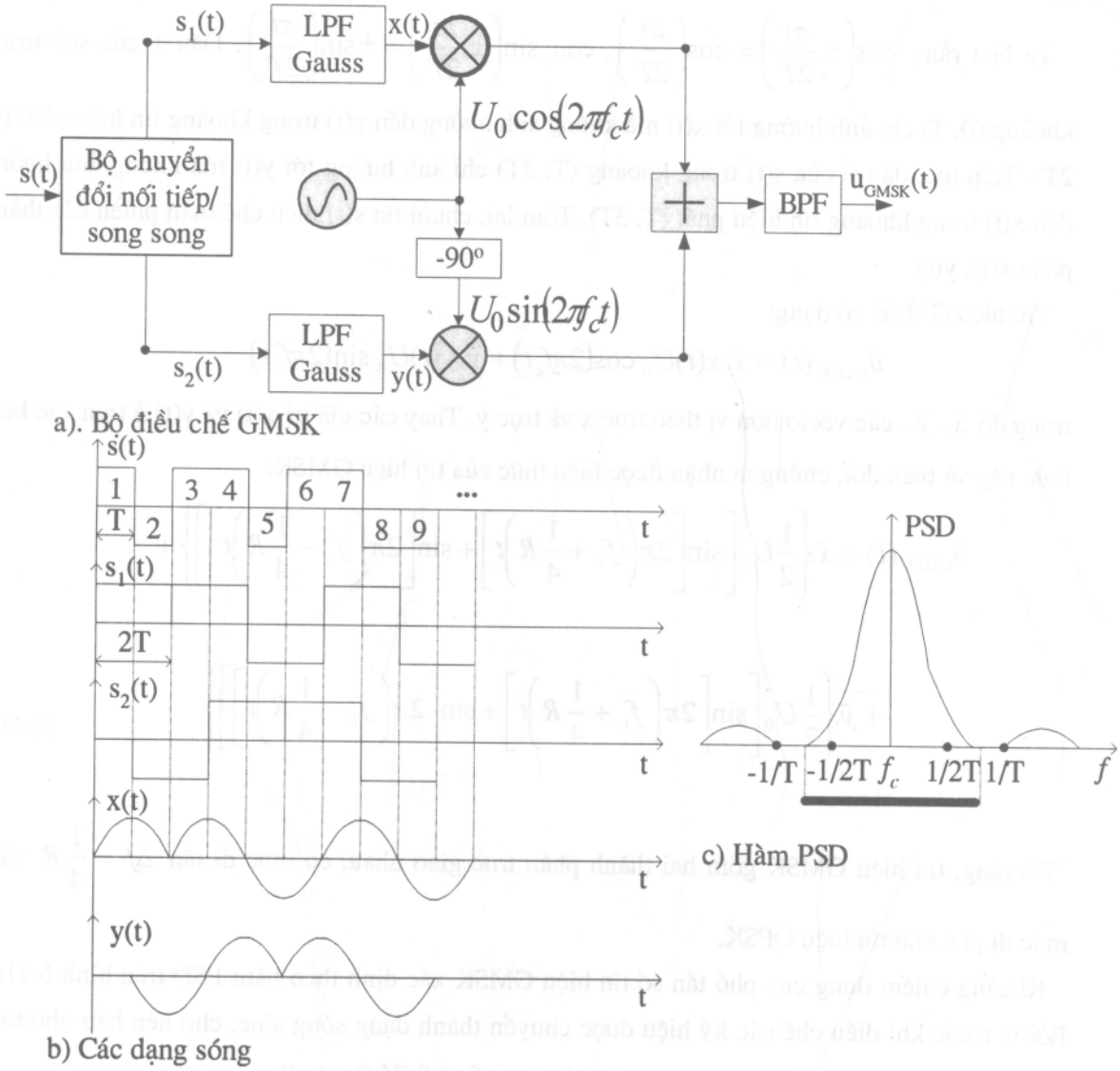
$$\frac{U_0^2}{2}\left[\frac{\sin(2\pi(f_1 - f_0)T}{2\pi(f_1 - f_0)}\right] + \frac{U_0^2}{2}\left[\frac{\sin(2\pi(f_1 + f_0) + 2\theta)T}{2\pi(f_1 + f_0)}\right]$$

Chúng ta có $f_1 - f_0 = 2\Delta f$, còn $f_1 + f_0 = 2f_c$. Vì f_c rất lớn, cho nên chúng ta có thể cho rằng thành phần thứ hai xấp xỉ bằng 0. Như vậy, điều kiện trực giao dẫn đến: $\sin(2\pi.2\Delta fT) = 0$, từ đây xác định được $\Delta f = \frac{1}{4}R$, hay chỉ số điều tần $m = 0,5$. Cuối cùng, chúng ta đã chứng minh được tín hiệu CP- FSK với chỉ số điều tần cực tiểu bằng 0,5, chuyển thành tín hiệu MSK. Tín hiệu khóa dịch tần cực tiểu-MSK có thể được thực hiện nhờ mạch lọc thông thấp- LPF có hàm truyền đạt hình Gauss, với dải thông ở mức 3dB, đặt trước bộ điều chế, được gọi là GMSK (Gaussian premodulation filtered Minimum Shift Keying), và giải điều chế liên kết.

Trên hình 6.21, biểu diễn sơ đồ khối mạch điều chế GMSK a), diễn giải quá trình hình thành dạng sóng b) và hàm PSD.

Mạch chuyển đổi nối tiếp thành song song có chức năng chuyển đổi chuỗi bít s(t) nối tiếp đầu vào thành hai chuỗi symbol song song $s_1(t)$ và $s_2(t)$ có thời khoảng các symbol mở rộng ra gấp đôi, tức 2T, như hình 6.21b).

Mạch lọc thông thấp Gauss với mức tần số cắt 3dB, ứng với dải thông B, có tác dụng như mạch *tạo dạng xung hình sine* (Chúng ta có thể giải thích như sau: hàm truyền của LPF có dạng $H(f) = \exp(-a.f^2)$, với a > 0, đáp ứng xung h(t) được xác định: h(t) $= FT^{-1}(H(f) =$ $\int_{-\infty}^{+\infty} \exp(-a.f^2).\exp(i2\pi f.t)df = 2\int_0^{+\infty} \exp(-af^2) \times \cos(2\pi f.t)df = \frac{\sqrt{\pi}}{\sqrt{a}}.\exp\left(-\frac{\pi^2}{a}.t^2\right)$. Tín hiệu đầu ra LPF $u_{ra}(t) = u_v(h) * h(t)$, ở đây $u_v(t)$ là các symbol có giá +1, hoặc - 1, còn $u_{ra}(t)$, chúng ta ký hiệu là x(t), y(t) sẽ có dạng sóng (gần đúng) là nửa hình sine, hoặc nửa hàm cosine).



Hình 6.21. Bộ điều chế GMSK a), các dạng sóng b), và hàm PSD c).

Đầu ra của chúng tạo ra dạng sóng x(t) có dạng nửa hàm sine, với chu kỳ 4T, tức là:

$$x(t) = \sin\left(\pm\frac{\pi t}{2T}\right) \quad \text{với } 0 < t < T$$

và dạng sóng y(t) có dạng nửa hàm sine cũng có chu kỳ 4T. Nhưng vì y(t) chậm thời gian 1/4 chu kỳ, hay chậm pha 90° so với x(t), nên y(t) có dạng hàm cosine, tức là:

$$y(t) = \cos\left(\pm\frac{\pi t}{2T}\right) \quad \text{với } 0 < t < T$$

Ta biết rằng $\cos\left(\pm\frac{\pi t}{2T}\right)=\cos\left(\frac{\pi t}{2T}\right)$, còn $\sin\left(\pm\frac{\pi t}{2T}\right)=\pm\sin\left(\frac{\pi t}{2T}\right)$. Dấu ± của s(t) trong khoảng (0, T) chỉ ảnh hưởng tới x(t) mà không ảnh hưởng đến y(t) trong khoảng tín hiệu phát (0, 2T). Tiếp tục, dấu ± của s(t) trong khoảng (T, 2T) chỉ ảnh hưởng tới y(t) mà không ảnh hưởng đến x(t) trong khoảng tín hiệu phát (T, 3T). Tóm lại, chuỗi bit s(t) điều chế luân phiên các thành phần x(t), y(t).

Tín hiệu GMSK có dạng:

$$\vec{u}_{GMSK}(t)=\vec{x}.x(t)U_0\cos(2\pi f_c t)+\vec{y}.y(t)U_0\sin(2\pi f_c t)$$

trong đó $\vec{x}$, $\vec{y}$ - các véctơ đơn vị theo trục x và trục y. Thay các giá trị x(t) và y(t) ở trên vào biểu thức này và biến đổi, chúng ta nhận được biểu thức của tín hiệu GMSK:

$$\vec{u}_{GMSK}(t)=\vec{x}.\left\{\frac{1}{2}U_0\left[\sin\left[2\pi\left(f_c+\frac{1}{4}R\right)t\right]+\sin\left[2\pi\left(f_c-\frac{1}{4}R\right)t\right]\right]\right\}+$$

$$+\vec{y}.\left\{\frac{1}{2}U_0\left[\sin\left[2\pi\left(f_c+\frac{1}{4}R\right)t\right]+\sin\left[2\pi\left(f_c-\frac{1}{4}R\right)t\right]\right]\right\} \qquad (6.41)$$

Rõ ràng, tín hiệu GMSK gồm hai thành phần trực giao nhau, có mức di tần $\Delta f=\frac{1}{4}R$ và 4 mức di pha như tín hiệu QPSK.

Khoảng chiếm dụng của phổ tần số tín hiệu GMSK xác định theo hàm PSD trên hình 6.21c). Bởi vì trước khi điều chế các ký hiệu được chuyển thành dạng sóng sine, cho nên hầu như toàn bộ năng lượng của tín hiệu tập trung trong khoảng $f_c\pm 0{,}75B$, tức là:

$$W_{GMSK}=2\times 0{,}75R=1{,}5R=1{,}5B \qquad (6.42)$$

Tín hiệu GMSK được sử dụng rộng rãi trong thực tiễn kỹ thuật: thông tin vệ tinh, thông tin vi ba, thông tin di động GSM.

Sau đây chúng ta đưa ra so sánh các sơ đồ điều chế: BPSK, QPSK (4QAM), OQPSK và GMSK về bốn tham số:

- Hiệu suất sử dụng phổ tần η [bit/sec/Hz], được tính ở khoảng phổ Nyquist;
- Tỉ số E_b/N_0, trong đó E_b năng lượng bit, $N_0/2$ mật độ công suất nhiễu (PSD), khi xác suất lỗi bit $P_b=10^{-5}$. Xác suất lỗi bit được xác định bằng công thức:

$$P_b=Q\left[\sqrt{\frac{2E_b}{N_0}}\right],\text{ trong đó } Q(y)=\frac{1}{\sqrt{2\pi}}\int_y^{\infty}e^{-\frac{x^2}{2}}dx$$

- Ảnh hưởng của méo phi tuyến (chẳng hạn, do bộ khuếch đại công suất);

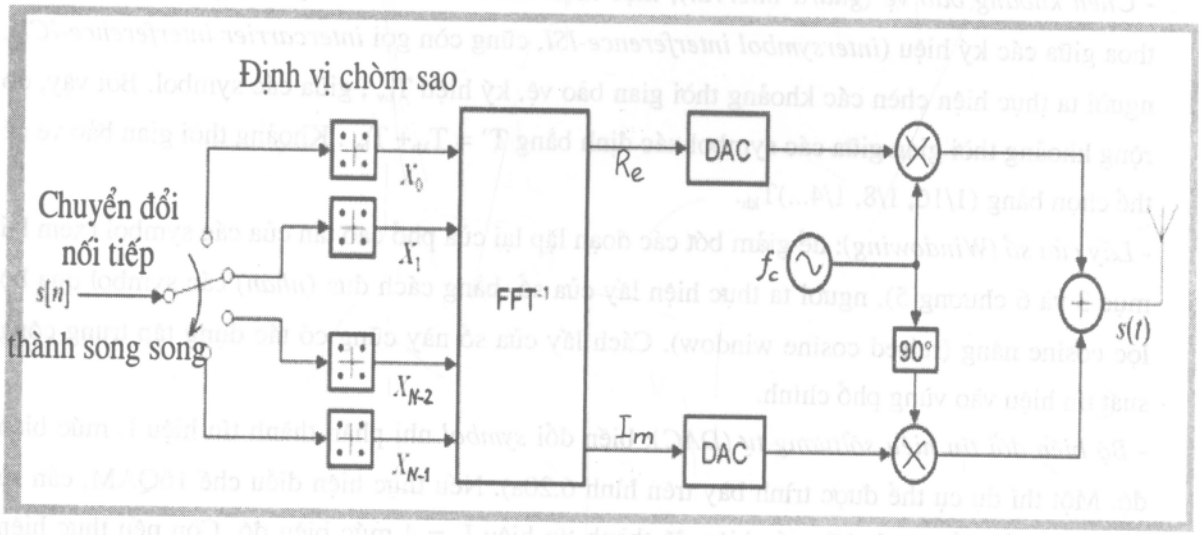
- Độ phức tạp trong chế tạo (kéo theo giá thành), được xếp hạng theo A, B, C, D, trong đó mức D là cao nhất.

Sơ đồ điều chế	Hiệu suất η (bit/sec/Hz)-Nyquist	E_b/N_0 (dB) khi $P_b = 10^{-5}$	A'nh hưởng méo phi tuyến	Độ phức tạp (giá thành)
BPSK	0,5	9,6	D	A
QPSK	1,0	9,6	C	B
OQPSK	1,0	9,6	B	C
GMSK	0,75	9,6	A	D

Với các tín hiệu điều chế, mà kết quả tạo ra tín hiệu nhiều mức (biên độ và pha), sẽ có hiệu suất sử dụng phổ tần số η > 1 (để truyền 1bit/sec không yêu cầu dải thông của kênh đến 1Hz). Tuy nhiên, trong một vài dạng hệ thống truyền tin không thể sử dụng chúng, chẳng hạn, hệ thống thông tin vệ tinh; bởi vì trong các hệ thống truyền tin như thế thường xuyên xảy ra pha đinh sâu về biên độ.

7. ĐIỀU CHẾ OFDM

Điều chế OFDM (*Orthogonal Frequency Division Multipplexing*) là phương thức điều chế số đa sóng mang, trong đó khoảng cách tần số giữa các sóng mang được xác lập sao cho các sóng mang trực giao lẫn nhau.



Hình 6.22. Mô hình bộ điều chế OFDM sử dụng FFT^{-1}.

Trên hình 6.22 trình bày mô hình bộ điều chế OFDM. Mô hình bộ điều chế OFDM bao gồm:

- *Bộ dồn kênh*: có chức năng chuyển đổi chuỗi bit s(n) nối tiếp, có vận tốc bit R, thành N chuỗi *symbol* song song, khi đó thời khoảng *symbol* mở rộng ra N lần, chúng ta ký hiệu là T_{kh}, còn vận tốc *symbol* được xác định $R_{kh} = \frac{1}{T_{kh}} = \frac{R}{N}$.

- *Bộ định vị chòm sao (constellation mapping)*: biến đổi các *symbol* ở dạng nhị phân thành các mẫu tín hiệu phức trong miền tần số, sau điều chế chúng trở thành các sóng mang con. Bộ định vị chòm sao sẽ *định vị* để các sóng mang con có đặc tính *trực giao* nhau, tức là chọn khoảng cách giữa chúng bằng $1/T_{kh}$. Trên hình vẽ ký hiệu các mẫu tín hiệu phức trong miền tần số là $X_0, X_1, ..., X_{N-2}, X_{N-1}$.

- *Gài tín hiệu lái (hoa tiêu) (Pilot Insertion)*: cùng với N các symbol OFDM, trong kỹ thuật này, người ta còn thực hiện gài một số *các symbol lái*, xen kẽ trong các symbol OFDM. Các symbol lái sau điều chế trở thành các sóng mang lái, đóng vai trò làm các điểm đánh dấu đồng bộ về tần số và thời gian.

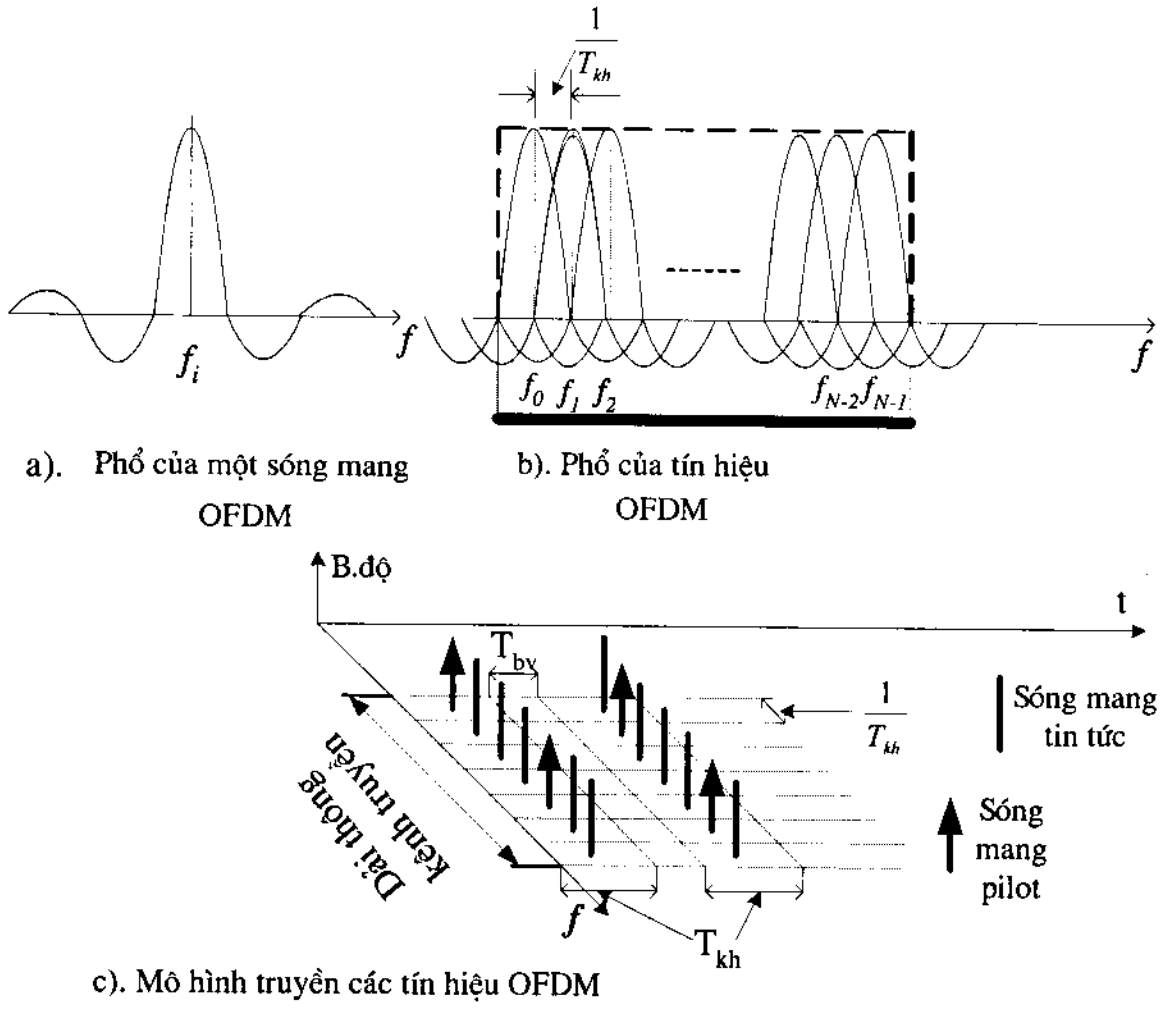
- *Mạch biến đổi Fourier nhanh ngược (FFT^{-1})*: biến đổi các mẫu phức của tín hiệu thuộc từng băng con trong miền tần số thành các mẫu phức của tín hiệu cũng thuộc mỗi băng con đó trong miền thời gian. Các mẫu phức của tín hiệu trong miền thời gian được tách thành phần thực-*Re* và phần ảo-*Im*. Các phần thực hình thành nên các *symbol* nhị phân, được DAC và được điều chế ở nhánh trên- nhánh I (*in-phase*). Các phần ảo cũng hình thành các *symbol* nhị phân, được DAC và được điều chế ở nhánh dưới-nhánh Q (*quadrature-phase*).

- *Chèn khoảng bảo vệ* (*guard interval*), thực hiện sau IFFT: để chống lại hiện tượng nội giao thoa giữa các ký hiệu (*intersymbol interference-ISI*, cũng còn gọi *intercarrier interference-ICI*), người ta thực hiện chèn các khoảng thời gian bảo vệ, ký hiệu T_{bv}, giữa các symbol. Bởi vậy, độ rộng khoảng thời gian giữa các symbol xác định bằng $T' = T_{kh} + T_{bv}$. Khoảng thời gian bảo vệ có thể chọn bằng (1/16, 1/8, 1/4...)T_{kh}.

- *Lấy cửa sổ (Windowing)*: để giảm bớt các đoạn lặp lại của phổ cao tần của các symbol (xem lại mục 3 và 6 chương 5), người ta thực hiện lấy cửa sổ, bằng cách đưa (*nhân*) các symbol qua bộ lọc cosine nâng (raised cosine window). Cách lấy cửa sổ này cũng có tác dụng tập trung công suất tín hiệu vào vùng phổ chính.

- *Bộ biến đổi tín hiệu số/tương tự (DAC)*: biến đổi *symbol* nhị phân thành tín hiệu L mức biên độ. Một thí dụ cụ thể được trình bày trên hình 6.20a). Nếu thực hiện điều chế 16QAM, cần sử dụng chuyển đổi *symbol* 2 mức biên độ thành tín hiệu L = 4 mức biên độ. Còn nếu thực hiện điều chế 64QAM, cần sử dụng chuyển đổi *symbol* 2 mức biên độ thành tín hiệu L = 8 mức biên độ, v.v...

- *Mạch điều chế MQAM*: gồm mạch tạo dao động mang, để tạo ra hai thành phần dao động mang có pha trực giao nhau. Mạch điều chế cân bằng (mạch nhân). Cuối cùng là mạch cộng và mạch lọc dải thông.



Hình 6.23. a) Phổ một sóng mang OFDM, b) Phổ của tín hiệu OFDM, c) Mô hình truyền các tín hiệu OFDM.

Điều chế OFDM là một dạng đặc biệt của hệ thống điều chế đa sóng mang, dựa trên nguyên tắc phân chia luồng dữ liệu tốc độ bit cao thành các luồng dữ liệu con có tốc độ ký hiệu thấp. Bộ định vị chòm sao sẽ xác định các sóng mang con có đặc tính trực giao nhau. Các sóng mang con với tốc độ bit thấp và với số lượng lớn sẽ mang được luồng dữ liệu có tốc độ bit cao. Nói một cách hình tượng, phương pháp điều chế đa sóng mang OFDM cũng giống như đấu nối dữ liệu giữa đầu phát và đầu thu bằng *cổng song song* thay vì *cổng nối tiếp*. Hình tượng này được thực hiện bởi *các bộ dồn kênh* phía phát và phía thu (xem hình 6.22 và 6.24).

Trên hình 6.23a) biểu diễn phổ tần số của một sóng mang con. Khoảng chiếm dụng phổ tần số của nó là: $W_i = \frac{2}{T_{kh}} = 2B$.

Còn trên hình 6.23b) biểu diễn phổ tần số của tín hiệu OFDM. Các tần số f_i được xác định: $f_i = f_c + F_i$.

Từ hình vẽ này, chúng ta thấy rõ các sóng mang con trực giao nhau, điều này được thể hiện ở khoảng cách tần số giữa các sóng mang con: $f_{i+1} - f_i = \frac{1}{T_{kh}}$. Khi đó, điểm phổ có năng lượng cao nhất của một sóng mang con này, rơi vào điểm phổ có năng lượng bằng không của một sóng mang con khác. Khi giải điều chế tín hiệu cao tần của băng con này, bộ điều chế sẽ không nhìn thấy các tín hiệu cao tần của các băng con khác, tức không bị gây nhiễu bởi các tín hiệu các sóng con khác. Vì các sóng mang con được đặt rất gần nhau, nên khoảng chiến dụng phổ tổng cộng cũng như khi điều chế trực tiếp dòng dữ liệu tốc độ cao.

Còn trên hình 6.23b) biểu diễn mô hình truyền các tín hiệu OFDM theo kênh. Từ mô hình này, chúng ta thấy rõ các khái niệm:

- Dải thông kênh truyền;
- Các sóng mang con, sóng mang "pilot" trong một vòng (*cyclic*). Khoảng cách tần số của các sóng mang con bằng $1/T_{kh}$, hơn nữa các sóng mang con trong một cyclic có chậm thời gian so với sóng mang con đầu cyclic;
- Khoảng thời gian symbol - T_{kh}, khoảng thời gian bảo vệ - T_{bv}.

Khoảng chiếm dụng phổ tần số của tín hiệu OFDM với N' (N'= N+N_p) băng con được xác định bằng công thức:

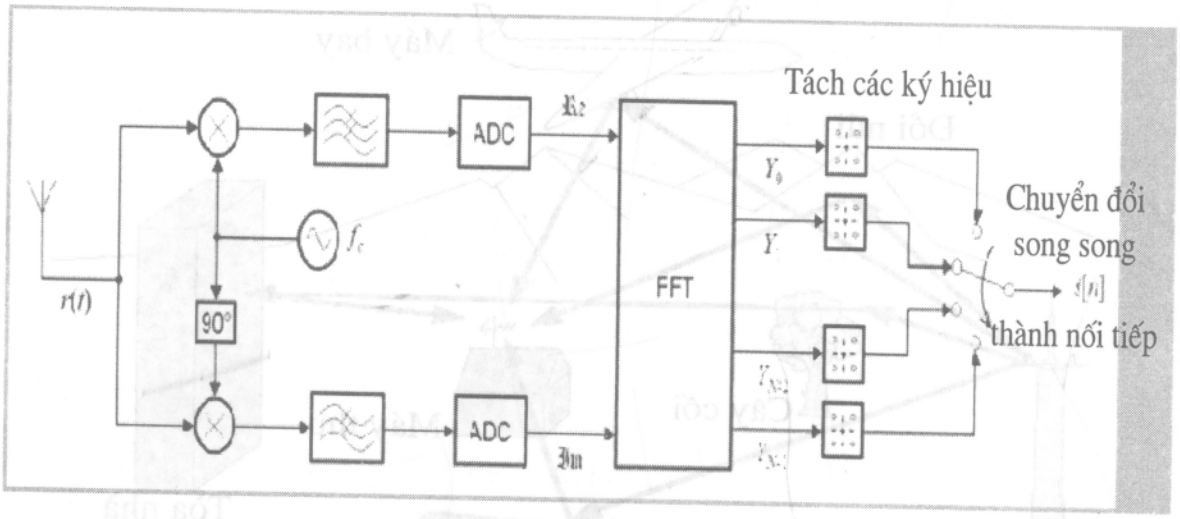
$$W_{OFDM} = f_{N-1} - f_0 + \frac{2}{T_{kh}} = (N' - 1) \times \frac{1}{T_{kh}} + \frac{2}{T_{kh}} = \frac{N' + 1}{T_{kh}} \qquad (6.43)$$

Trên hình 6.24. trình bày mô hình bộ giải điều chế tín hiệu OFDM.

Ngày nay, nguyên lý OFDM được sử dụng rộng rãi trong thực tiễn kỹ thuật, chẳng hạn:

-Trong truyền hình kỹ thuật số, chẳng hạn, DVB-T *(Digital Video Broadcasting- Terrestrial), DVB-S (Digital Video Broadcasting- Satellite), DTH (Direct To Home);*

- Modem ADSL *(Asymetric Digital Subscriber Line)*, VDSL *(Very high speed Digital Subscriber Line)*, cho phép truy cập Internet tốc độ cao bằng đôi dây đồng;
- *Công nghệ Wimax (Wireless MAN)*;
- *Công nghệ Wi-Fi (Wireless LAN)*;
- Ứng cử viên sáng giá cho *thông tin di động thế hệ G4.*

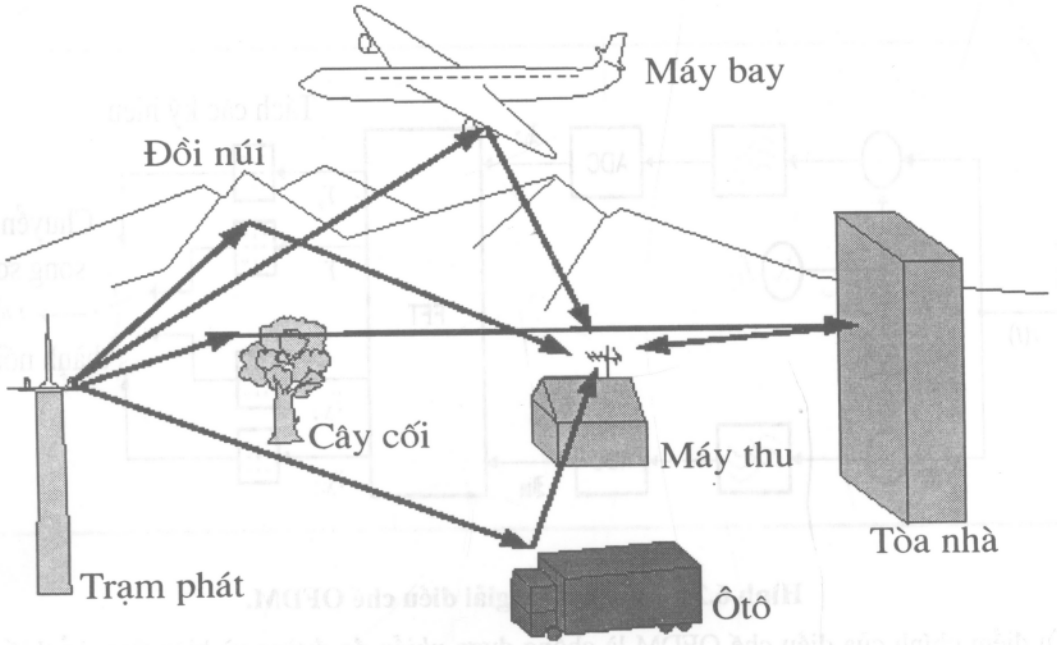


Hình 6.24. Mô hình bộ giải điều chế OFDM.

Ưu điểm chính của điều chế OFDM là chống được *nhiễu đa đường* và hiệu ứng do *thiếu dải thông* của kênh truyền. Nguyên nhân của hai hiệu ứng này có bản chất hoàn toàn khác nhau, nhưng hậu quả thì giống nhau đều dẫn đến giao thoa giữa các ký hiệu (ISI) và phát sinh lỗi tại phía thu.

Trên hình 6.25 mô tả mô hình kênh đa đường. Nhiễu *đa đường* sẽ có hậu quả khi thời gian trễ của *các tia đến sau* so sánh được với thời khoảng ký hiệu, tức là $\tau_{di} \geq T_{kh}$, trong đó τ_{di} - thời gian trễ của *các tia đến sau*, còn T_{kh} - thời khoảng *các symbol*. Có thể loại bỏ loại nhiễu loại này bằng giải pháp *mở rộng thời khoảng ký hiệu*, như giải pháp điều chế OFDM. Lúc đó $\tau_{di} << T_{kh}$, cho nên, phép cộng của nhiều tia sóng có kết quả gần giống như chỉ của một tia sóng (sự lệch pha của các tia sóng gần như nhau).

Hiệu ứng do *thiếu dải thông* của kênh truyền, có thể loại bỏ bằng cách *giảm nhỏ tốc độ ký hiệu* (cũng tức là mở rộng *thời khoảng ký hiệu*) của từng băng con, như giải pháp điều chế OFDM. Khi đó dải thông yêu cầu của từng băng con sẽ nhỏ hơn rất nhiều so với dải thông có sẵn của kênh truyền.



Hình 6.25. Mô hình kênh đa đường.

8. CÁC TÍN HIỆU TRẢI PHỔ

8.1. Khái niệm chung về điều chế trải phổ

Như chúng ta đã nghiên cứu trong các mục trước, ở các hệ thống truyền tin sử dụng kỹ thuật *điều chế truyền thống*, vấn đề độ rộng phổ của tín hiệu là vấn đề được quan tâm chính. Ở các hệ thống này được thiết kế sao cho độ rộng phổ của tín hiệu càng hẹp càng tốt. Hiệu suất sử dụng phổ tần số được xác định bằng công thức (6.37):

$$\eta = \frac{R}{W}\left[\frac{bit/\sec}{Hz}\right]$$

Cùng một vận tốc bit R [bit/sec] của tín hiệu đem điều chế, tín hiệu điều chế nào cho độ rộng phổ W [Hz] càng nhỏ, thì hiệu suất sử dụng phổ tần số càng cao.

Các hệ thống truyền tin sử dụng kỹ thuật *điều chế trải phổ*, bề rộng phổ của tín hiệu rộng gấp hàng trăm lần bề rộng phổ của tín hiệu băng cơ sở. Để đánh giá mức độ trải phổ, người ta đưa vào tham số *hệ số khuếch đại xử lý* (*Processing Gain - PG*), được định nghĩa:

$$PG = \frac{\text{Độ rộng băng tần của tín hiệu trải phổ}}{2(\text{Độ rộng băng tần của tín hiệu băng cơ sở})} = \frac{W_{SS}}{2B} \qquad (6.44)$$

Các hệ thống điều chế trải phổ hiện nay, bao gồm:

- Hệ thống trải phổ dãy trực tiếp (*Direct Sequence Spread Spectrum*-DS/SS)
- Hệ thống trải phổ nhảy tần (*Frequency Hopping Spread Spectrum*-FH/SS)
- Hệ thống trải phổ lai ghép của hai hệ thống trên (DS/FH/SS)

Các hệ thống điều chế trải phổ được sử dụng rộng rãi trong các hệ thống truyền tin:

- Thông tin di động theo các tiêu chuẩn IS-95, IMT2000, sử dụng công nghệ đa truy nhập theo mã (*Code Division Multiple Access*- CDMA);
- Hệ thống định vị toàn cầu của Mỹ (*Global Positioning System*-GPS), sử dụng bộ tạo chuỗi giả ngẫu nhiên bằng chuỗi Gold;
- Thông tin vệ tinh sử dụng công nghệ CDMA;
- Hệ thống đa truy nhập bằng *sóng hồng ngoại Bluetooth* (công nghệ FH/SS);
- Các hệ thống thông tin liên lạc quân sự sử dụng FH/SS;
- v.v...

Trong mục này chúng ta sẽ nghiên cứu sơ bộ các tín hiệu trải phổ, chi tiết hơn độc giả có thể tham khảo thêm các tài liệu khác, chẳng hạn, [16].

8.2. Bộ tạo chuỗi tín hiệu giả ngẫu nhiên (PN)

Thành phần quan trọng nhất trong mọi hệ thống điều chế trải phổ là *bộ tạo chuỗi tín hiệu giả ngẫu nhiên (Pseudo-Noise)*. Chúng ta cần nghiên cứu chúng trước tiên.

Hiện thời tồn tại một vài phương pháp tạo chuỗi tín hiệu giả ngẫu nhiên sau:

- Chuỗi giả ngẫu nhiên loại m, hay còn gọi là chuỗi m;
- Chuỗi giả ngẫu nhiên loại Gold;
- Chuỗi giả ngẫu nhiên loại Kasami;
- v.v...

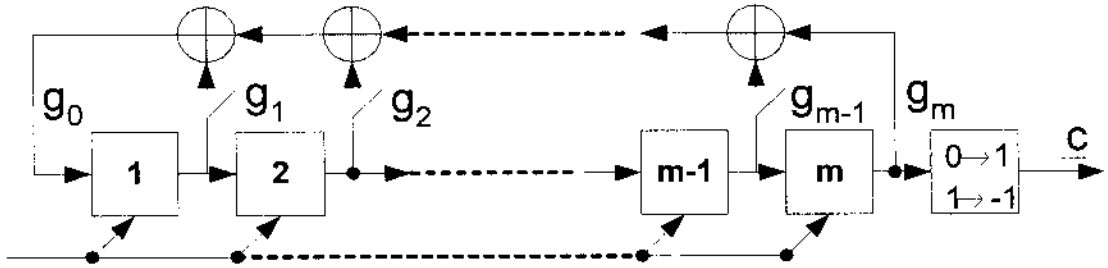
a). Các chuỗi PN m

Trong các chuỗi PN, quan trọng nhất là chuỗi m. Các chuỗi m được phát sinh từ đa thức nguyên thủy g(x) bậc m. *Đa thức nguyên thủy g(x) bậc m là đa thức, mà chỉ có đa thức* $\left(x^n+1\right)$ *với* $n = 2^m - 1$, *chia hết cho nó. Ngoài ra không có đa thức nào có dạng* $\left(x^p+1\right)$ *với* $1 \le p < m$, *chia hết cho nó.* Đa thức nguyên thủy g(x) bậc m có dạng:

$$g(x) = g_0 + g_1(x) + g_2x^2 + ... + g_{m-1}x^{m-1} + g_mx^m \qquad (6.45)$$

trong đó các hệ số của đa thức có đặc tính: $g_0, g_m \equiv 1$, còn các hệ số g_i khác hoặc bằng 1, hoặc bằng 0.

Mạch tạo chuỗi m bao gồm m ô ghi dịch, và một số bộ cộng mod2 thích hợp ở nhánh hồi tiếp, như hình 6.26.



Hình 6.26. Mạch tạo chuỗi PN m trong trường hợp chung.

Trong mạch tạo chuỗi PN m trên hình 6.26, quy ước hệ số của đa thức sinh g_i nào khác không, công tắc tương ứng với chúng ở vị trí đó nối mạch, đồng thời tồn tại phần tử cộng mod2; ngược lại hệ số g_i nào bằng không, công tắc tương ứng với chúng ở vị trí đó ngắt mạch, đồng thời không tồn tại phần tử cộng mod2. Như vậy, có thể xác định được số bộ cộng mod2 của mạch bằng W[g(x)] - 2. Ở đầu ra của mạch có phần tử chuyển đổi mức logic của các chip, phù hợp theo quy tắc $0 \to 1$ và $1 \to -1$.

Chúng ta hãy bắt đầu nghiên cứu chuỗi PN m bằng các thí dụ đơn giản nhất.

Thí dụ 1: Xét trường hợp m = 3, n = $2^3 - 1 = 7$. Như đã biết, trong trường hợp này có hai đa thức nguyên thủy. Đa thức sinh được chọn trong thí dụ này là:

$$g(x) = x^3 + x^2 + 1 \tag{6.46}$$

Mạch điện tạo chuỗi m theo đa thức sinh ở trên có dạng được biểu diễn trên hình 6.27. Mạch bao gồm 3 ô ghi dịch, 1 bộ cộng mod2. Mạch nằm trong khung viền nét đứt được gọi là "mặt chắn"- Mask ; mạch này tạo ra chuỗi $\underline{c}$'- chuỗi ra qua mặt chắn.

Chúng ta giả thiết trạng thái ban đầu của các ô ghi dịch là: 111- gọi là nạp ban đầu.

Chuỗi ra $\underline{c}$, với giá trị nạp ban đầu 111, được xác định theo bảng bên phải của hình 6.27, trong khuôn khổ giữa hai mũi tên, dùng để định vị chu kỳ:

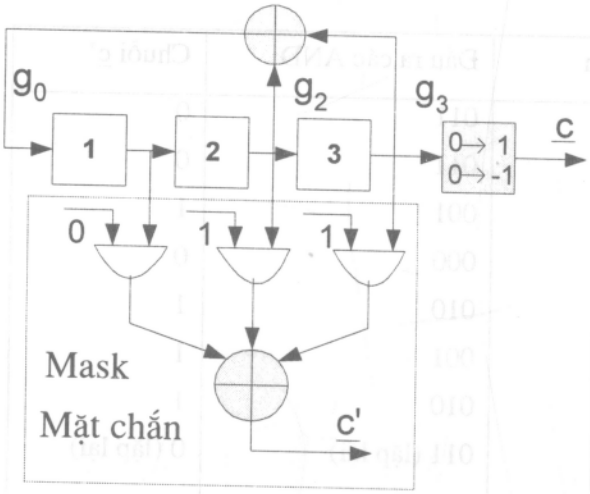
$$\underline{c} = 1110010\ldots \tag{6.47}$$

Công thức (6.47) xác định một chu kỳ của chuỗi $\underline{c}$. Nếu chúng ta ký hiệu chu kỳ của chuỗi $\underline{c}$ là N, thì N được xác định: $N = 2^m - 1 = 2^3 - 1 = 7$. Hơn nữa, với đa thức sinh g(x) là *đa thức nguyên thủy*, thì giá trị của chu kỳ N đạt giá trị *cực đại*. Một thí dụ xét sau này, sẽ chứng tỏ với đa thức sinh không phải là đa thức nguyên thủy, giá trị của chu kỳ N sẽ *không* đạt giá trị *cực đại*.

Mỗi phần tử trong chuỗi $\underline{c}$ được gọi là một *chip* và ký hiệu:

$$\underline{c} = c_0 c_1 c_2 \ldots c_{N-2} c_{N-1}$$

Với mỗi chip tồn tại các định nghĩa: *thời gian chip*, T_c [sec], *tốc độ chip*, R = $\frac{1}{T_c}$ [chip/sec].



Xung nhịp	Trạng thái các ô ghi dịch
0	111
1	011
2	001
3	100
4	010
5	101
6	110
7	111 (lặp lại)
...	...

Hình 6.27. Mạch tạo chuỗi PN với m = 3.

Giá trị ban đầu của các ô ghi dịch, gọi là *nạp ban đầu*. Có tất cả $2^m - 2$ giá trị nạp ban đầu (trừ giá trị toàn 0). Với các giá trị nạp ban đầu khác nhau, chu kỳ của chuỗi c sẽ được bắt đầu khác nhau. Chẳng hạn, với thí dụ trên, khi giá trị nạp ban đầu là 100, thì chuỗi c được xác định:

$$\underline{c}' = 0010111... = T^{-3}\underline{c}$$

trong đó $T^{-3}\underline{c}$ - ký hiệu dịch vòng trái của chuỗi c đi ba vị trí. Như vậy, có thể kết luận: chuỗi c phát sinh từ các giá trị nạp ban đầu khác nhau sẽ là dịch vòng trái (dịch vòng phải) của nhau.

Với mỗi mạch tạo chuỗi m, có hai cách lấy ra chuỗi c. Cách thứ nhất lấy chuỗi c ra như ở trên, gọi là chuỗi c ra *trực tiếp*. Cách thứ hai lấy chuỗi c ra qua "mặt chắn"-Mask. Mặt chắn được hình thành từ các phần tử AND và phần tử cộng mod2. Với mỗi phần tử AND, có một đầu vào được thiết lập mức logic hoặc 1, hoặc 0, được gọi *giá trị của mặt chắn*. Có $2^m - 2$ giá trị của mặt chắn (trừ giá trị toàn 0). Trong thông tin di động sử dụng công nghệ CDMA, giá trị mặt chắn của các trạm công cộng được gọi là *số seri điện tử* (Electronic Serial Number-ERN), giá trị mặt chắn của các máy cá nhân được gọi là *số định danh trạm di động* (Mobile station Identify Number MIN). Ứng với mỗi giá trị mặt chắn, xác định được một chuỗi ra c'. Chẳng hạn, với giá trị 011, chuỗi ra c'được xác định như bảng dưới đây.

Các giá trị ở cột 3 của bảng được xác định bằng *nhân logic* tương ứng các giá trị của cột 2 với giá trị của mặt chắn - 011.

Kết quả cho chuỗi c' được xác định ở cột cuối cùng của bảng, các phần tử của cột 4 là kết quả cộng mod2 của các phần tử đầu ra các AND tương ứng ở cột 3:

$$\underline{c}' = 0010111... = T^{-3}\underline{c}$$

Như vậy, chuỗi tín hiệu PN của các cá nhân khác nhau, trong thông tin di động, là dịch vòng của nhau.

Xung nhịp	Đầu ra các ô ghi dịch	Đầu ra các AND	Chuỗi c'
1	111	011	0
2	011	011	0
3	001	001	1
4	100	000	0
5	010	010	1
6	101	001	1
7	110	010	1
8	111 (lặp lại)	011 (lặp lại)	0 (lặp lại)
...	...	...	...

Tập các chuỗi $\underline{c}$, ký hiệu S_m, được xác định:

$$S_m = \begin{cases} \underline{c} & = 1110010 \\ T^{-1}\underline{c} & = 1100101 \\ T^{-2}\underline{c} & = 1001011 \\ T^{-3}\underline{c} & = 0010111 \\ T^{-4}\underline{c} & = 0101110 \\ T^{-5}\underline{c} & = 1011100 \\ T^{-6}\underline{c} & = 0111001 \\ \underline{0} & = 0000000 \end{cases} \tag{6.48}$$

Thí dụ 2: Xét trường hợp m = 4, n = $2^4 - 1 = 15$.

$$\left(x^{15}+1\right) = (x+1)\left(x^2+x+1\right)\left(x^4+x+1\right)\left(x^4+x^3+1\right)\left(x^4+x^3+x^2+x+1\right)$$

trong ba đa thức bậc 4, chỉ có 2 đa thức đầu là đa thức nguyên thủy. Đa thức bậc 4 cuối cùng không phải là đa thức nguyên thủy, bởi vì:

$$\left(x^5+1\right) = (x+1)\left(x^4+x^3+x^2+x+1\right)$$

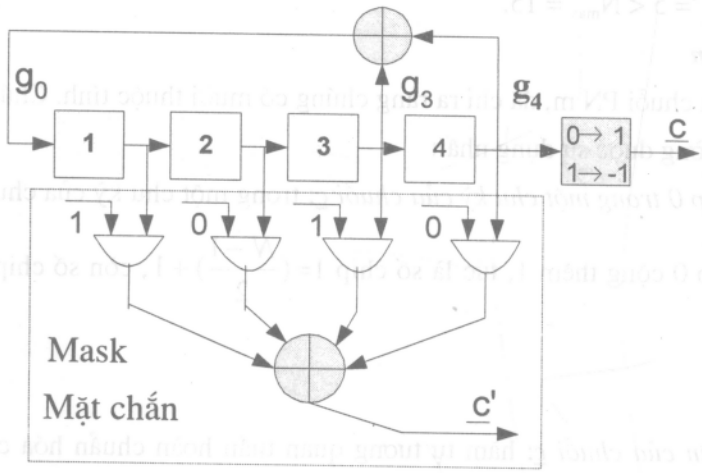
tức là đa thức $\left(x^5+1\right) \vdots \left(x^4+x^3+x^2+x+1\right)$.

Chọn đa thức sinh $g(x) = \left(x^4+x^3+1\right)$. Mạch tạo chuỗi $\underline{c}$ từ đa thức sinh này, như biểu diễn trên hình 6.28.

Với giá trị nạp ban đầu: 111; chuỗi $\underline{c}$ ra trực tiếp có dạng:

$$\underline{c} = 111100010011010.... \tag{6.49}$$

Chu kỳ của chuỗi $\underline{c}$ được xác định: $N = 2^4 - 1 = 15$



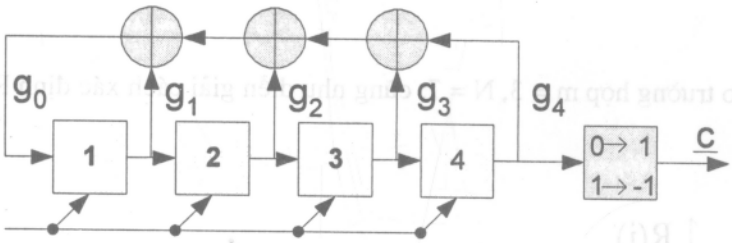
Xung nhịp	Trạng thái các ô ghi dịch	Đầu ra các AND	Chuỗi c'
0	1111	1010	0
1	0111	0010	1
2	0011	0010	1
3	0001	0000	0
4	1000	1000	1
5	0100	0000	0
...	....	....	...

Hình 6.28. Mạch tạo chuỗi c từ đa thức sinh $g(x) = \left(x^4 + x^3 + 1\right)$.

Với giá trị mặt chắn 1010, chuỗi c' ra qua mặt chắn có dạng:

$$c' = 011010111100010... = T^{-9}c$$

Quá trình hình thành chuỗi c, cũng như chuỗi c' được giải thích (trích đoạn đầu) ở bảng bên phải của hình 6.28.



Xung nhịp	Trạng thái các ô ghi dịch
0	1111
1	0111
2	1011
3	1101
4	1110
5	1111
...	(lặp lại)
	....

Hình 6.29. Mạch tạo chuỗi PN từ đa thức không nguyên thủy.

Thí dụ 3: Xét quá trình tạo chuỗi PN, với đa thức sinh *không phải đa thức nguyên thủy*. Chẳng hạn, chọn đa thức sinh $g(x) = \left(x^4 + x^3 + x^2 + x + 1\right)$. Như trên đã chứng tỏ, đây không phải là đa thức nguyên thủy. Mạch tạo chuỗi PN như hình 6.29.

Chu kỳ của chuỗi PN có giá trị N'= 5 < N_{max} = 15.

b). Các thuộc tính của chuỗi PN m

Các nghiên cứu về thuộc tính của chuỗi PN m, đã chỉ ra rằng chúng có mười thuộc tính. Chúng ta chỉ dẫn ra đây hai thuộc tính thường được sử dụng nhất:

- *Thuộc tính về số chip 1 và số chip 0 trong một chu kỳ của chuỗi c*: trong một chu kỳ của chuỗi c số chip 1 luôn luôn bằng số chip 0 cộng thêm 1, tức là số chip $1=(\frac{N-1}{2})+1$, còn số chip 0 $=\left(\frac{N-1}{2}\right)$.

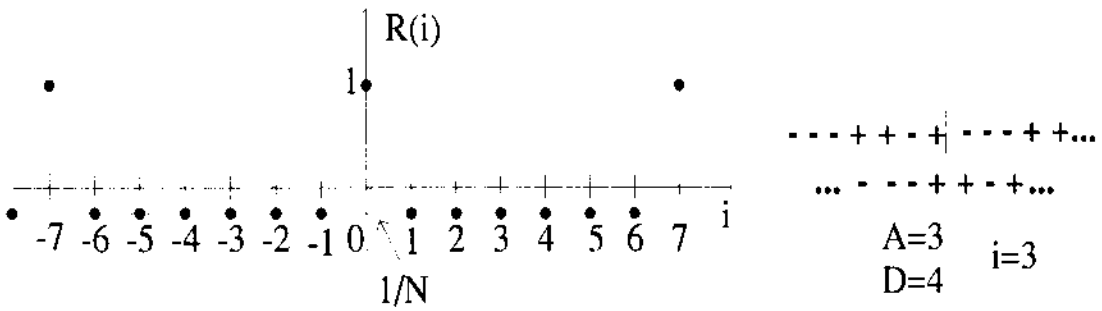
- *Thuộc tính về hàm tự tương quan của chuỗi c*: hàm tự tương quan tuần hoàn chuẩn hóa của chuỗi m có dạng đầu đỉnh.

Hàm tự tương quan tuần hoàn chuẩn hóa của chuỗi m được xác định như sau:

$$R_c(i)=\frac{1}{N}\sum_{j=0}^{N-1}c_j c_{j+i}=\frac{1}{N}(A-D)=\begin{cases}1 & i=\pm nN\\ -1/N & i\neq nN\end{cases} \qquad (6.50)$$

trong đó: i = số lượng vị trí dịch của hai chuỗi; n = 0, 1, 2, 3,...; A = số lượng của tích hai chip cùng dấu, còn D = số lượng của tích hai chip khác dấu. Rõ ràng, khi số lượng vị trí dịch i = 0, hoặc nguyên lần chu kỳ N, thì A = N, còn D = 0; còn khi i ≠ nN, thì A = $\frac{N-1}{2}$, còn $D=\left(\frac{N-1}{2}\right)+1$. Kết quả tính toán này được giải thích nhờ thuộc tính số chip 1 và 0 trong một chu kỳ khác nhau một.

Đồ thị hàm tự tương quan cho trường hợp m = 3, N = 7; cũng như diễn giải cách xác định R(i), được biểu diễn trên hình 6.30.



Hình 6.30. Hàm tự tương quan R(i) của chuỗi c với m = 3.

Khi xác định A và D, chúng ta đã thực hiện chuyển đổi: $0\rightarrow 1$, còn $1\rightarrow -1$

c). Các chuỗi Gold

Tập các chuỗi Gold được thiết lập trên cơ sở từ *cặp chuỗi m được ưa chuộng* (Preferred Pair) có *cùng chu kỳ N*, để bảo đảm hàm tương quan chéo tuần hoàn giữa hai chuỗi là nhỏ nhất.

Chúng ta gọi hai chuỗi m là $\underline{u}$ và $\underline{v}$, thì hàm tương quan chéo tuần hoàn giữa hai chuỗi được định nghĩa:

$$R_{u,v}(n) = \sum_{i=0}^{N-1} u_i v_{i+n} \tag{6.51}$$

với mọi giá trị của n nguyên, còn chỉ số (i + n) được lấy theo modN.

Cặp chuỗi m được gọi là *ưa chuộng*, ký hiệu $\underline{x}$ và $\underline{y}$, nếu hàm tương quan chéo tuần hoàn giữa hai chuỗi chỉ nhận ba giá trị sau:

$$R_{x,y}(n) = -1,\ -t(m),\ t(m) - 2$$

với mọi giá trị của n nguyên, còn $t(m) = 1 + 2^{\left[\frac{m+2}{2}\right]}$, trong đó ký hiệu [.] chỉ phép tính lấy phần nguyên.

Tập các chuỗi Gold bao gồm cặp chuỗi m được ưa chuộng $\underline{x}$, $\underline{y}$ và các tổng mod2 của $\underline{x}$ với các dịch vòng trái của $\underline{y}$:

$$S_{Gold} = \left\{\underline{x},\ \underline{y},\ \underline{x} \oplus \underline{y},\ \underline{x} \oplus T^{-1}\underline{y},\ \underline{x} \oplus T^{-2}\underline{y}, \ldots,\ \underline{x} \oplus T^{(N-1)}\underline{y}\right\} \tag{6.52}$$

Thí dụ : Xét trường hợp m = 3, cặp chuỗi m ưa chuộng có đa thức sinh:

$g_1(x) = x^3 + x^2 + 1$ và $g_2(x) = x^3 + x + 1$. Sơ đồ mạch điện tạo tập chuỗi Gold được biểu diễn trên hình 6.31a). Còn hàm tương quan chéo tuần hoàn của hai chuỗi $\underline{x}$ và $\underline{y}$, được biểu diễn trên hình 6.31b).

Với giá trị nạp ban đầu các ô ghi dịch cho cả hai mạch là 001. Tập các chuỗi Gold được xác định:

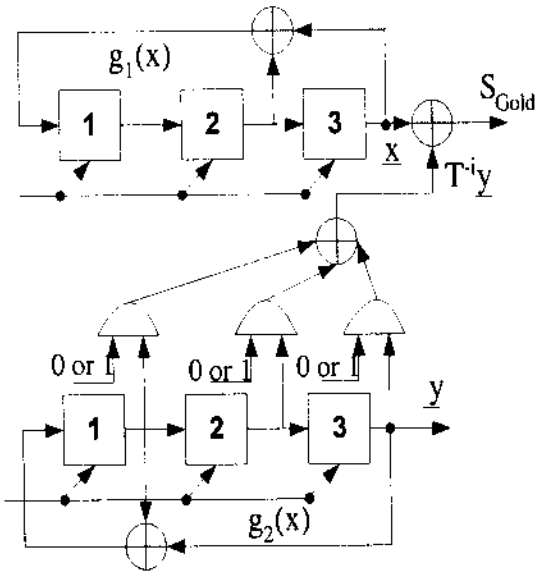
$$\underline{x} = 1001011;\ \underline{x} \oplus \underline{y} = 0000101;\ \underline{x} \oplus T^{-1}\underline{y} = 1010110;\ \underline{x} \oplus T^{-2}\underline{y} = 1110001$$

$$\underline{y} = 1001110;\ \underline{x} \oplus T^{-3}\underline{y} = 0111111;\ \underline{x} \oplus T^{-4}\underline{y} = 0100010;$$

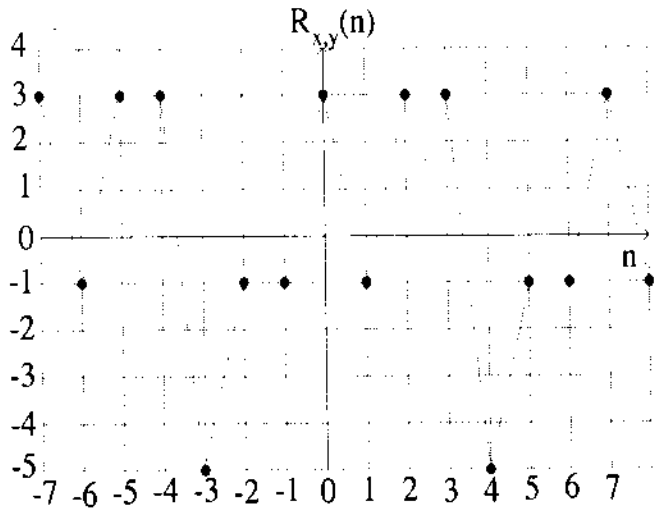
$$\underline{x} \oplus T^{-5}\underline{y} = 0011000;\ \underline{x} \oplus T^{-6}\underline{y} = 1101100$$

Với cặp chuỗi m là $\underline{x}$ và $\underline{y}$ như trên, hàm tương quan chéo giữa chúng chỉ nhận ba giá trị: −1; 3 và −5. Đồ thị hàm $R_{x,y}(n)$ trong hai chu kỳ được biểu diễn trên hình 6.31b).

Các cặp chuỗi m ưa chuộng không tồn tại với m = 4, 8, 12, 16 và phỏng đoán rằng với mọi m = 0mod4.



a). Mạch tạo chuỗi Gold



b). Hàm tương quan chéo của 2 chuỗi:

- + + - + - - và - + + - - - +

Hình 6.31. Mạch tạo tập chuỗi Gold với m = 3 a), và hàm tương quan chéo của hai chuỗi - + + - + - - và - + + - - - + b).

Với m = 5, cặp đa thức m ưa chuộng được xác định là:

$$g_1(x) = x^5 + x^2 + 1 \text{ và } g_2(x) = x^5 + x^4 + x^2 + x + 1$$

trong trường hợp này $t(m) = 1 + 2^{\left\lfloor \frac{m+2}{2} \right\rfloor} = 9$. $R_{x,y}(n) = -1,\ -9,\ 7.$

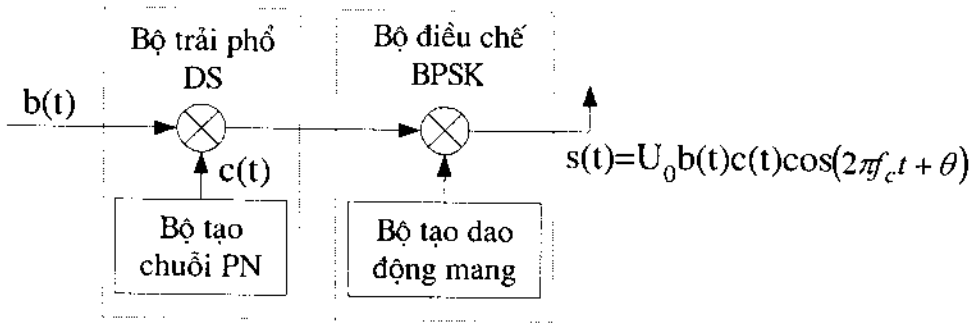
Chúng ta nhận thấy rằng tỉ số: $t(m)/N \cong 2^{-m/2} \to 0$ khi $m \to \infty$; tức là các chuỗi Gold với m lớn sẽ có các giá trị của hàm tương quan chéo rất nhỏ.

8.3. Hệ thống trải phổ dãy trực tiếp DS/SS

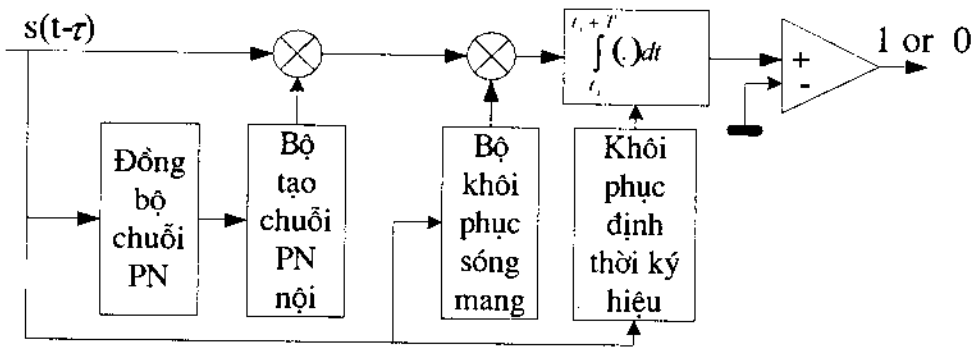
Hệ thống trải phổ chuỗi trực tiếp có thể thực hiện điều chế BPSK, hoặc QPSK. Chúng ta hạn chế chỉ xét hệ thống DS/SS - BPSK.

Sơ đồ khối hệ thống trải phổ chuỗi trực tiếp, điều chế khóa dịch pha hai mức DS/SS -BPSK được biểu diễn trên hình 6.32. Trong đó, sơ đồ khối hệ thống phát được biểu diễn trên hình 6.32a). Hệ thống phát bao gồm: bộ trải phổ DS. Ở đây, tín hiệu nhị phân băng cơ sở b(t) được nhân (cũng có thể cộng mod2) với tín hiệu PN c(t). Kết quả của phép nhân [b(t)×c(t)], tạo ra một tín hiệu băng cơ sở đã được trải phổ. Tín hiệu này, sau đó được điều chế khóa dịch tần hai mức, cũng nhờ một phép nhân. Kết quả của phép nhân [b(t)×c(t)] với dao động mang $U_0 \cos(2\pi f_c t + \theta)$ tạo ra tín hiệu DS/SS - BPSK có dạng:

$$s(t) = U_0 b(t) c(t) \cos(2\pi f_c t + \theta) \tag{6.53}$$



a). Sơ đồ khối máy phát DS/SS-BPSK



b). Sơ đồ khối máy thu DS/SS-BPSK

Hình 6.32. Hệ thống DS/SS – BPSK.

Đồ thị hàm PSD của các tín hiệu: b(t), c(t), [b(t)×c(t)], s(t) được biểu diễn trên hình 6.33. Cũng trên hình vẽ này, chúng ta cũng biểu diễn hàm tự tương quan chuẩn hóa của tín hiệu c(t), ký hiệu $R_c(\tau)$. Trong đó, hàm tự tương quan chuẩn hóa $R_c(\tau)$ trong chu kỳ đầu được xác định:

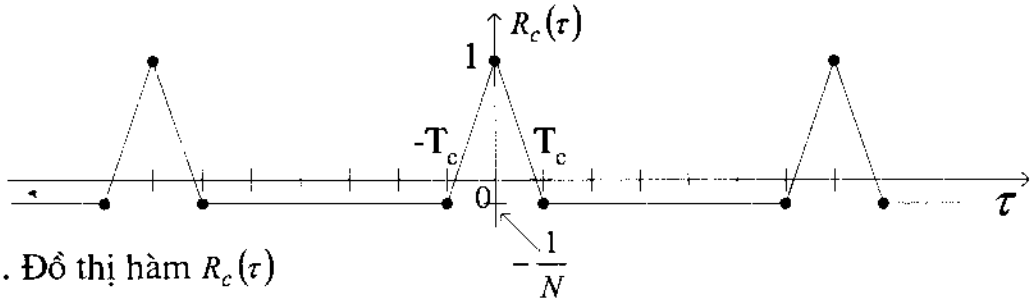
$$R_c(\tau) = \frac{1}{N}[c(t) \times c(t \pm \tau)] = \begin{cases} 1 - \dfrac{|\tau|}{T_c}\left(1 + \dfrac{1}{N}\right) & 0 \le |\tau| \le T_c \\ -\dfrac{1}{N} & T_c < |\tau| < (N-1)T_c \end{cases} \tag{6.54}$$

Khi biểu diễn đồ thị của các tín hiệu b(t), c(t), [b(t) × c(t)], s(t), chúng ta giả thiết $T = NT_c$, với N = 7, trong đó T- thời gian bit của tín hiệu b(t), còn T_c - thời gian chip của tín hiệu c(t).

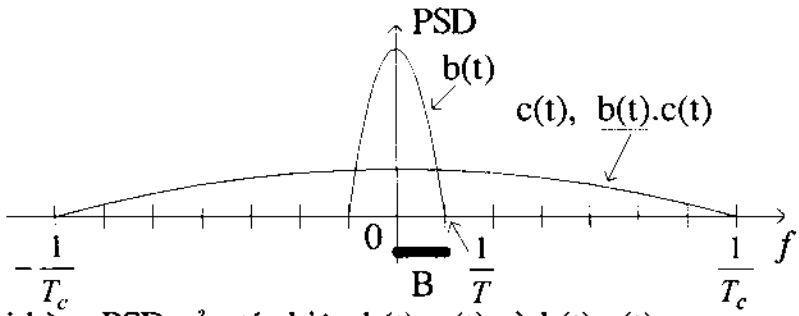
Hệ số khuếch đại xử lý PG, trong trường hợp này, được xác định:

$$PG = \frac{W_{SS}}{2B} = \frac{2/T_c}{2/T} = N \tag{6.55}$$

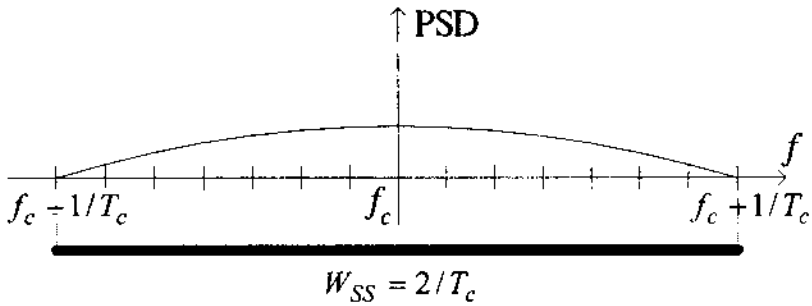
Với m = 3, chúng ta có PG = 7; còn với m = 10, chúng ta có PG = 1023.



a). Đồ thị hàm $R_c(\tau)$



b). Đồ thị hàm PSD của tín hiệu b(t), c(t) và b(t).c(t)



c). Đồ thị hàm PSD của tín hiệu DS/SS-BPSK

Hình 6.33. Đồ thị của hàm $R_c(\tau)$ a), của các tín hiệu b(t), c(t), b(t).c(t) b), và của tín hiệu DS/SS-BPSK c).

Trên hình 6.32b), biểu diễn sơ đồ khối máy thu DS/SS - BPSK. Ở đây cần giải quyết các vấn đề cơ bản sau:

- Đồng bộ chuỗi PN đến với chuỗi PN nội, gồm hai pha chính:
  + Bắt đồng bộ;
  + Bám đồng bộ
- Bộ khôi phục sóng mang, để thực hiện giải điều chế liên kết.
- Bộ khôi phục định thời ký hiệu (xác định thời điểm bắt đầu của ký hiệu-t.), cùng bộ ngưỡng, để *quyết định* bit 1 hoặc 0.

**** Hiệu năng của các hệ thống DS/SS***

- Các hệ thống DS/SS chịu tác động của tạp âm trắng, nhưng có thể loại bỏ các nhiễu dải hẹp.

- Các hệ thống DS/SS có thể chống được nhiễu đa đường, nhưng chịu ảnh hưởng của nhiễu giao thoa. Để giảm nhỏ nhiễu giao thoa cần thiết kế các tín hiệu PN có tương quan chéo nhỏ.
- Các hệ thống DS/SS chịu ảnh hưởng mạnh của hiệu ứng gần xa, để giảm nhỏ hiệu ứng này cần thiết kế *bộ tự điều khiển công suất.*
- Tính chất khó thu trộm, vì tín hiệu do trải phổ thường chìm trong nền nhiễu.

8.4. Hệ thống trải phổ nhảy tần FH/SS

Ý tưởng của các hệ thống FH/SS là *nhảy* hay *chuyển đổi tần số sóng mang theo tập hợp các mẫu tần số được xác định bởi chuỗi PN*. Tốc độ nhảy tần số có thể nhanh hay chậm, tùy theo quan hệ $T/T_h > 1$ hay $T/T_h < 1$.

Trên hình 6.34 biểu diễn sơ đồ khối máy phát FH/SS. Tín hiệu nhị phân b(t) trước hết được điều chế FSK (hoặc BPSK). Ta chỉ xem xét điều chế FSK. Trong khoảng thời gian bit, tín hiệu x(t) nhận một trong hai tần số f' hoặc f'+Δf, tùy theo bit 0 hoặc bit 1 của b(t) đầu vào, tức là:

$$x(t) = \begin{cases} \cos(2\pi f' t) & b(t) = 0 \\ \cos[2\pi(f' + \Delta f)t] & b(t) = 1 \end{cases} \tag{6.56}$$

Tín hiệu này được trộn tần (thông qua phép nhân) với tín hiệu y(t) từ bộ tổng hợp tần số. Cứ trong mỗi khoảng thời gian của bước nhảy T_h, tần số của tín hiệu y(t) lại nhận một trong các giá trị từ tập 2^k tần số do bộ tổng hợp tần số tạo ra, tức là:

$$y(t) = U_0 \cos[2\pi(f_0 + i_m \Delta f)t + \theta_m]$$

trong đó: $i_m \in \{0, 2, ..., 2^{(k-1)}\}$ là số nguyên chẵn, m- chỉ bước nhảy thứ m.

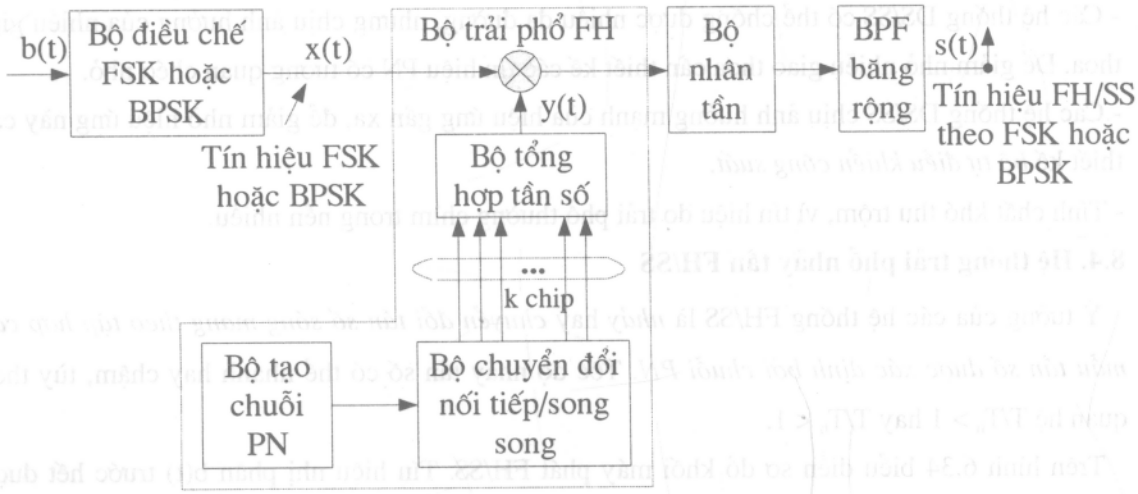
Bộ trộn tạo ra tín hiệu có tần số là tổng và hiệu của hai tần số của tín hiệu x(t) với tín hiệu y(t). Chẳng hạn, bộ lọc dải thông băng rộng được thiết kế để chọn ra tín hiệu có tần số là tổng hai tần số (lúc này tạm bỏ qua bộ nhân tần), tức là:

$$s(t) = 2U_0 \cos[2\pi(f_0 + i_m \Delta f + b_m \Delta f)t + \theta_m] \tag{6.57}$$

trong đó: $b_m \in \{0, 1\}$. Như vậy, các tần số phát nằm trong tập:

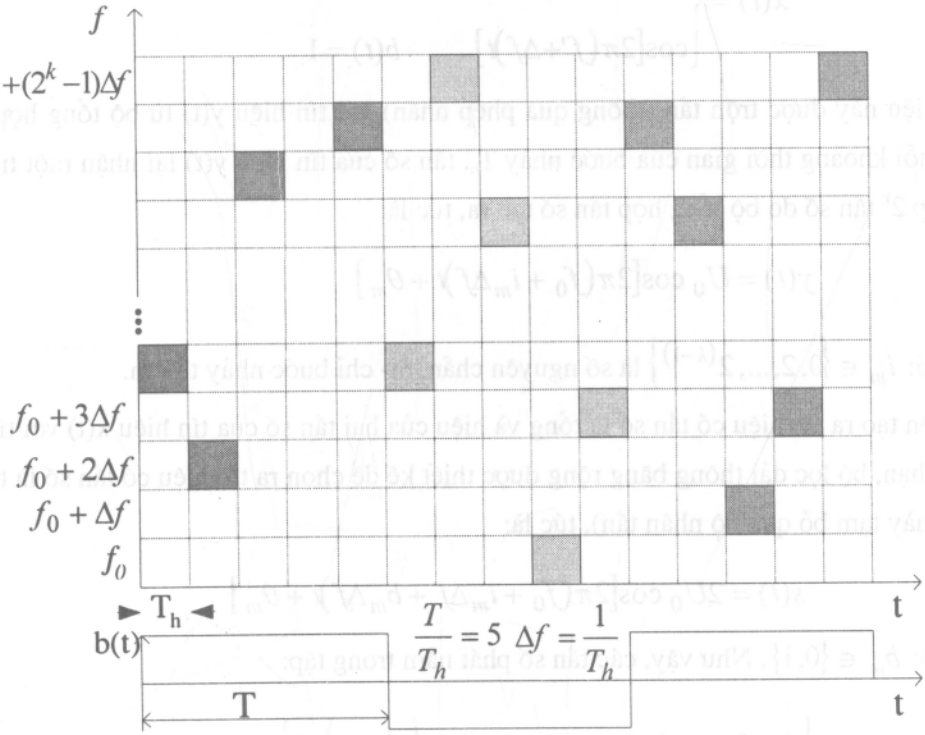
$$\{f_0, f_0 + \Delta f, f_0 + 2\Delta f, ..., f_0 + (2^k - 1)\Delta f\} \tag{6.58}$$

trong đó: Δf được chọn bằng $1/T_h$, để các tín hiệu s(t) trong các bước nhảy lân cận nhau là trực giao với nhau.



Hình 6.34. Sơ đồ khối máy phát FH/SS.

Bộ nhân tần (*tùy chọn*) sẽ nhân tần số, cũng như pha đầu của tín hiệu s(t) lên β lần.



Hình 6.35. Biểu đồ tần số cho một hệ thống FH/SS- FSK.

Trên hình 6.35. biểu diễn biểu đồ tần số cho một hệ thống FH/SS nhảy tần nhanh với $T/T_h = 5$.

Bởi vì, bộ tổng hợp tần số không có khả năng duy trì liên kết pha của dao động khi nó nhảy từ tần số này sang tần số kia, cho nên hệ thống FH/SS thường sử dụng kỹ thuật *giải điều chế không liên kết*.

Độ rộng băng tần của tín hiệu FH/SS là:

$$W_{SS} = 2^k \Delta f$$

Hệ số khuếch đại xử lý của hệ thống FH/SS là:

$$PG = \frac{2^k \Delta f}{2/T} = \frac{2^k T}{2T_h} \tag{6.59}$$

Nếu tồn tại bộ nhân tần β lần, thì hệ số khuếch đại xử lý của hệ thống FH/SS là:

$$PG = \frac{\beta 2^k T}{2T_h} \tag{6.59a}$$

CÂU HỎI VÀ BÀI TẬP

Câu1: Vẽ sơ đồ khối tổng quát của hệ thống truyền tin. Nêu chức năng chính từng khối?

Câu 2: Nêu và phân tích các yêu cầu chính của hệ thống truyền tin?

a). Phân loại các hệ thống nén số liệu?

- Thuật toán nén ảnh đen trắng JPEG có mấy bước?

- Thuật toán nén tín hiệu tiếng nói có thể phân chia thành mấy loại, kết quả dải tốc độ bit R của từng loại ?

b). Nguyên lý chung xây dựng mã chống nhiễu? Hãy lấy thí dụ đơn-giản nhất để giải thích các nguyên lý chung này?

- Phân loại mã chống nhiễu?

c). Tốc độ truyền thông tin R trên kênh phụ thuộc vào các yếu tố nào?

Câu 3: Hãy nêu và giải thích các đại lượng:

- Lượng thông tin riêng $I(x_i)$, lượng thông tin trung bình H(X), lượng thông tin tương hỗ I(X,Y), Entropy có điều kiện H(Y/X), H(X/Y)?

- Độ dư thừa của nguồn tin g_M, độ dư thừa của tập mã g_C ?

- Khả năng thông qua C, dải thông B của kênh truyền phụ thuộc vào những yếu tố nào?

Câu 4: Cho tập tin có 1000 tin, bao gồm 4 loại tin a_1 , a_2 , a_3 , a_4 , với xác suất tương ứng 1/2, 1/4 , 1/8 , 1/8.

- Tính số lượng mỗi loại tin?

- Nếu đem mã hoá bằng mã có độ dài đều. Hãy xác định chiều dài từ mã ?. Viết ra các từ mã ? Tính độ dư thừa của tập mã đều g_{CD}?

- Lập các từ mã theo Shannon-Fano ? Xác định $\overline{n}$, g_{CSF} của bộ mã?

- Xác định số ký hiệu mã thiết kiệm được khi mã hoá đều và mã Sh-F?

Câu 5: Kể tên một vài phép mã hoá nguồn mà em biết ? Cho tập tin X có 8 tin, với xác suất xuất hiện các tin $p(x_i)$ = 1/8; 1/8; 1/4; 1/16; 1/16; 1/16; 1/16; 1/4.

- Hãy xác định các lượng thông tin riêng, lượng thông tin trung bình, độ dư thừa của tập tin?

- Hãy lập các từ mã theo Shannon-Fano; theo Huffman?

- Xác định $\overline{n}$, $\overline{n}_{min}$, g_C?

- Biểu diễn cây cho tập mã

Câu 6: Mã khối tuyến tính là gì? Thuật toán mã hoá ? Thuật toán giải mã sửa sai? Cho mã khối tuyến tính M(7, 4), có ma trận sinh:

$$G_{(7,4)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Biết các bit thông tin I=1010, tìm từ mã V tương ứng?
- Vẽ mạch điện mã hoá?
- Thu được từ mã P=1111010. Giải mã từ mã này?
- Vẽ mạch điện giải mã?

Câu 7: Mã Cyclic là gì ? Thuật toán mã hoá ? Thuật toán giải mã phát hiện sai ? Các thuật toán giải mã sửa sai ? Mã Cyclic được ứng dụng ở đâu ?

Câu 8: Mã Cyclic C(7, 4), biết đa thức sinh $g(x)= x^3+ x^2+ 1$:

- Vẽ mạch điện mã hoá?
- Biết các bit thông tin I= 1010, tìm từ mã V tương ứng?
- Thu được từ mã P= 1001101, giải mã từ mã này bằng thuật toán chia dịch vòng, hoặc phương pháp phân tích phần dư (Meggit)?

Câu 9: Mã Cyclic C(7, 3), có đa thức sinh $g(x) = (x+1)(x^3+x^2+1)$

- Xác định đa thức kiểm tra h(x)?
- Biết các bit thông tin đầu vào I = 110, tìm từ mã V tương ứng?
- Vẽ mạch điện mã hoá?
- Thu được từ mã P = 1011011, giải mã từ mã này?
- Vẽ mạch điện giải hoá theo phương pháp chia, dịch vòng?

Câu 10: Mã Cyclic C(7, 3), có đa thức sinh $g(x) = (x+1)(x^3+x+1)$

- Xác định đa thức kiểm tra h(x)?
- Biết các bit thông tin đầu vào I = 110, tìm từ mã V tương ứng?
- Vẽ mạch điện mã hoá?
- Thu được từ mã P = 1011110, giải mã từ mã này?
- Vẽ mạch điện giải hoá theo phương pháp Meggit?

Câu 11: Phân tích đa thức $(x^{15}+1)$ thành các đa thức nguyên tố, được:

$$(x^{15}+1) = (x+1)(x^2+x+1)(x^4+x+1)(x^4+x^3+1)(x^4+x^3+x^2+x+1)$$

- Các phương án chọn đa thức sinh g(x) và xác định mã C(15,k) tương ứng, cũng các tham số của bộ mã: khoảng cách Hamming cực tiểu d_0, số lỗi phát hiện được e, số lỗi sửa được t ? Trình bày kết quả theo bảng sau:

Số tt	Chọn đa thức sinh $g_t(x)$	Mã C(15,k) tương ứng	Khoảng cách cực tiểu- d_0	Số lỗi phát hiện- e	Số lỗi sửa- t

Câu 12: Mã RS (7, 5, 2), biết các chuỗi bit thông tin có dạng:

I = 010_011_101_011_111

- Xác định từ mã V tương ứng?
- Thu được chuỗi mã: P = 111_011_101_100_110_101_111, hãy giải mã P để tìm chuỗi mã, chuỗi bit thông tin đã phát hay $\overline{P}$, $\overline{I}$?

Câu 13: Mã RS (7, 3, 4), biết các chuỗi bit thông tin có dạng:

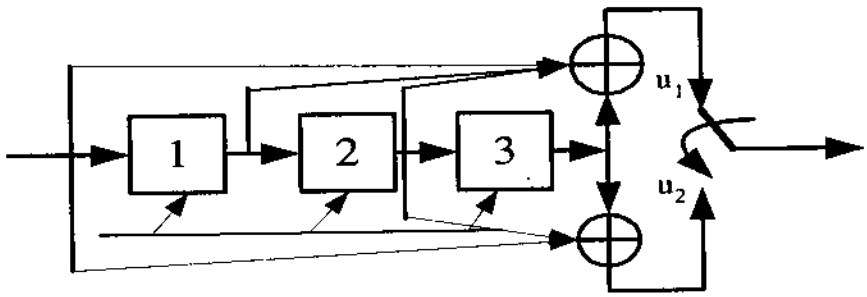
I = 010_011_101

- Xác định từ mã V tương ứng?
- Thu được chuỗi mã: P = 010_110_111_101_000_010_101, hãy giải mã P để tìm chuỗi mã, chuỗi bit thông tin đã phát hay $\overline{P}$, $\overline{I}$?

Câu 14: Mã xoắn X(2, 1, 2), mạch điện mã hoá như bài học:

- Biết các bit thông tin I = 110101..., tìm từ mã V tương ứng bằng phương pháp đa thức kết nối, hoặc phương pháp sơ đồ trạng thái ?
- Thu được tập ký hiệu mã P =11-01-11-00-01-..., hãy giải mã bằng thuật toán Viterbi tìm tập ký hiệu mã đúng $\overline{P}$ và tập bit thông tin đúng $\overline{I}$?
- Mã xoắn được ứng dụng ở đâu ?

Câu 15: Mã xoắn X(2, 1, 3), có sơ đồ mã hoá như hình dưới:



Biết các bit thông tin đầu vào I = 11010...

- Xác định tập dấu mã đầu ra theo phương pháp đa thức kết nối?
- Xác định trạng thái của các ô ghi dịch tương ứng, nếu trạng thái ban đầu A=110 ?

Câu 16: Thuật toán RSA, biết khoá công khai KU = {5, 161}, văn bản mã C = 66. Hãy:

- Xác định khoá riêng KR?
- Tìm văn bản rõ M?

Câu 17: Xét QTNN dừng X(t), có HMĐSX 1 chiều dạng:

$$f(x) = ae^{-|x|} \quad \text{với } a > 0$$

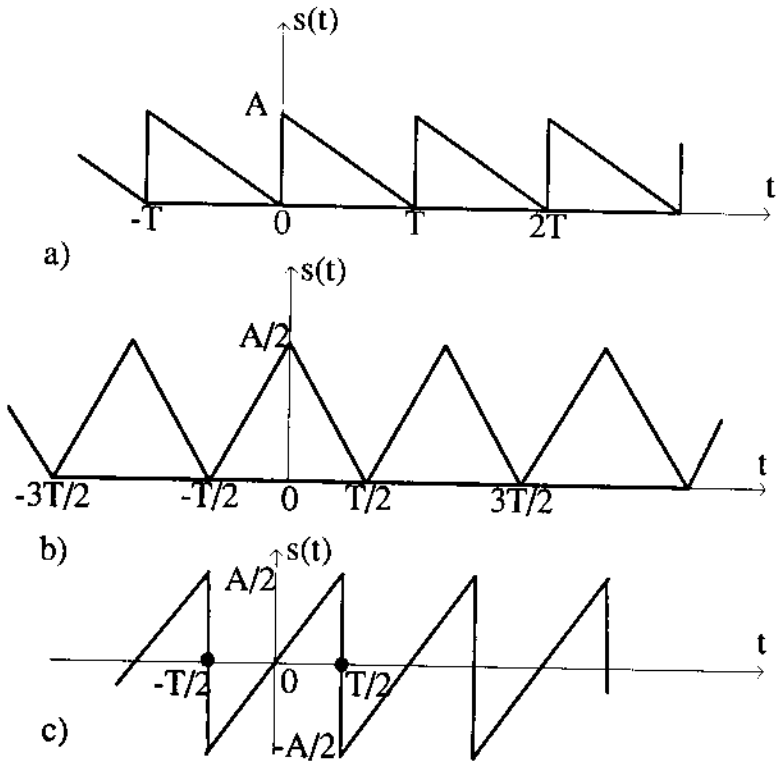
- Xác định hệ số a, vẽ đồ thị hàm f(x)?
- Xác định hàm F(x) và vẽ đồ thị của nó?
- Xác định kỳ vọng E[X(t)], phương sai D[X(t)]?
- Xác định xác suất để X(t) lấy giá trị trong khoảng (0, 1)?

Câu 18: Phân tích các chuỗi xung s(t) có đồ thị như hình vẽ, thành chuỗi Fourier. *Chú ý*: chuỗi xung a) có biên độ là A vôn, chu kỳ T msec;

chuỗi xung b) có biên độ là A/2 vôn, chu kỳ T msec;

chuỗi xung c) có biên độ là ±A/2 vôn, chu kỳ T msec.

Với các giá trị cụ thể: A = 1,5 vôn; T = 0,5 msec



Câu 19: Cho tín hiệu s(t) là chuỗi xung vuông tuần hoàn (như thí dụ trong chương 5). Cho biên độ xung A=1vôn, độ rộng xung τ = 1μsec, chu kỳ chuỗi xung T = 6μsec. Hãy:

- Viết chuỗi Fourier của tín hiệu s(t)?
- Xác định ít nhất 7 giá trị của biên độ (C_k), tần số (f_k) tương ứng của các thành phần tần phổ của chuỗi (bắt đầu từ C_0)?
- Vẽ đồ thị phổ biên độ - tần số?

Câu 20: Tín hiệu audio có phổ tần số F = 16 ÷ 20.000 Hz, được số hoá bằng thuật toán PCM. Hãy:

a) Xác định tần số, chu kỳ chuỗi xung mẫu thoả mãn định lý lấy mẫu. Lấy 3 thí dụ về tần số chuỗi xung mẫu thoả mãn điều kiện này và có dự trữ chống trùm phổ?

b) Xác định tần số Nyquist

c) Cho rằng LPF của mạch mã hoá/giải mã là lý tưởng, hãy xác định tần số cắt F_c của chúng?

d) Cho rằng quá trình lượng tử hoá là đồng đều, quá trình mã hoá mỗi xung mẫu được từ mã dài 16 bit. Hãy xác định số mức lượng tử hoá N, tốc độ bit đầu ra R phù hợp với thí dụ 1 ở mục a)?

Câu 21: Đem điều chế số dãy số liệu I = 11001..., có vận tốc bít 10,5 Kb/sec theo sơ đồ:

- ASK khoá ON- OFF;
- FSK với 2 mức tần số;
- và BPSK.
- Vẽ đồ thị thời gian của các tín hiệu điều chế này (tuỳ chọn giả thiết về các tần số sóng mang f_c)?
- Vẽ đồ thị phổ của các tín hiệu điều chế trên?
- Xác định các bề rộng phổ tín hiệu điều chế (W)?
- Nếu đem điều chế dãy số liệu trên theo sơ đồ 4QAM, xác định bề rộng phổ của tín hiệu?

Câu 22 : Tín hiệu điều chế biên độ, biết:

- Dao động mang $u_c(t) = 10\cos(2\pi f_c t + \theta)$;
- Tín hiệu điều chế $s(t) = 3\cos(2\pi F_1 t) + 2\cos(2\pi F_2 t) + 5\cos(2\pi F_3 t)$;

. Lập biểu thức tín hiệu điều chế biên độ?

. Tính các chỉ số điều chế ?

. Vẽ đồ thị phổ biên độ - tần số (Cho rằng: $f_c >> (F_3 > F_2 > F_1)$)?

. Xác định bề rộng phổ tín hiệu (Nếu $F_3 = 2F_2 = 3F_1 = 13{,}4$ kHz)?

Câu 23: Tín hiệu điều tần FM, biết chỉ số điều chế m = 3,5; tần số của tín hiệu đem điều chế F = 3,1kHz. Hãy:

- Xác định độ di tần cực đại Δf_{max} ?
- Vẽ đồ thị phổ của tín hiệu FM (sử dụng đồ thị của các hàm Bessel loại I) và xác định bề rộng phổ của nó?

Câu 24: Các ưu điểm chính của điều chế OFDM?

- Nêu một vài ứng dụng của điều chế OFDM và thử giải thích vì sao điều chế OFDM được sử dụng ở đó?

Câu 25: Cho đa thức nguyên thủy $g(x) = x^3 + x + 1$

- Vẽ mạch điện tạo chuỗi m, cùng mặt chắn của nó?
- Xác định chuỗi c với nạp ban đầu 101?

- Xác định các giá trị mặt chẵn, nếu chuỗi $\underline{c}' = T^{-4}\underline{c}$?

Câu 26: Cho đa thức nguyên thủy $g(x) = x^4 + x + 1$

- Vẽ mạch điện tạo chuỗi m, cùng mặt chẵn của nó?
- Xác định chuỗi $\underline{c}$ với nạp ban đầu 1001?
- Xác định các giá trị mặt chẵn, nếu chuỗi $\underline{c}' = T^{-4}\underline{c}$?

Câu 27: Vẽ mạch điện tạo tập chuỗi Gold với cặp đa thức bậc 5?

- Với nạp ban đầu của cả hai mạch là 11100, hãy xác định tập chuỗi Gold?
- Vẽ đồ thị hàm $R_{x,y}(n)$ của cặp chuỗi $\underline{x}$, $\underline{y}$?

Câu 28: Xác định tập chuỗi Gold với m = 3 khi giá trị nạp ban đầu 101?

-Vẽ đồ thị hàm $R_{x,y}(n)$ của cặp chuỗi $\underline{x}$, $\underline{y}$?

PHỤ LỤC 1: CHƯƠNG TRÌNH VÀ KẾT QUẢ TÍNH DFT [cos(αn)]

```
Program DFT [cos(αn)];
    Uses Crt;
    const
    pi = 3.141592653589793238;
    K = 32;
    var
    n : interger;
    a, b :real;
    T1, T2, T3, T4, M1, M2, Re, Im : real;
    Begin
    Clrscr;
    a : = (3 *pi)/(K-1);
    b : = (2 * pi)/K;
    For n : = 0 to (K-1) do
Begin
T1: = ((0.5 + 0.5*cos(b*n))*(1- cos (a-(b*n))))-((0.5*sin(b*n))*(sin(a-(b*n))));
T2: = ((0.5 + 0.5*sin(b*n))*(1- cos (a-(b*n))))+((0.5+0.5*cos(b*n))*(sin(a-(b*n))));
T3: = ((0.5 + 0.5*cos(b*n))*(1- cos (a-(b*n))))-((0.5*sin(b*n))*(sin(a+(b*n))));
T2: = ((0.5 + 0.5*sin(b*n))*(1- cos (a+(b*n))))+((0.5+0.5*cos(b*n))*(sin(a+(b*n))));
M1: =((1-cos(a-(b*n)))*(1-cos(a-(b*n))))+((sin(a-(b*n)))*sin(sin(a-(b*n))));
M2: =((1-cos(a+(b*n)))*(1-cos(a+(b*n))))+((sin(a+(b*n)))*sin(sin(a+(b*n))));
    Re:=(T1/M1)+(T3/M2);
    Im:=(T2/M1)-(T4/M2);
    Writeln(n,':','Re[X(',n,')]=',Re:6:3,'','Im[X'.n,')]=',Im:6:3);
    end;
    readln;
    End.
```

KẾT QUẢ DFT [cos(αn)]

0: Re[X(0)=1.000	Im[X(0)]=0.000
1: Re[X(1)=-0.106	Im[X(1)]=7.251
2: Re[X(2)=2.855	Im[X(2)]=-12165
3: Re[X(3)=1.594	Im[X(3)]=-4.147
4: Re[X(4)=1.282	Im[X(4)]=-2.443
5: Re[X(5)=1.090	Im[X(5)]=-1.623
6: Re[X(6)=0.933	Im[X(6)]=-1.118
7: Re[X(7)=0.791	Im[X(7)]=-0.772
8: Re[X(8)=0.657	Im[X(8)]=-0.524
9: Re[X(9)=0.530	Im[X(9)]=-0.343
10: Re[X(10)=0.412	Im[X(10)]=-0.213
11: Re[X(11)=0.305	Im[X(11)]=-0.122
12: Re[X12)=0..210	Im[X(12)]=-0.062
13: Re[X(13)=0.131	Im[X(13)]=-0.026
14: Re[X(14)=0.069	Im[X(14)]=-0.008
15: Re[X(15)=0.025	Im[X(15)]=-0.001
16: Re[X(16)=0.000	Im[X(16)]=-0.000
17: Re[X(17)=-0.005	Im[X(17)]=0.001
18: Re[X(18)=0.008	Im[X(18)]=0.008
19: Re[X(19)=0.038	Im[X(19)]=0.026
20: Re[X(20)=0.083	Im[X(20)]=0.062
21: Re[X(21)=0.140	Im[X(21)]=0.122
22: Re[X(22)=0. 205	Im[X(22)]=0.213
23: Re[X(23)=0. 275	Im[X(23)]=0.343
24: Re[X(24)=0.343	Im[X(24)]=0.524
25: Re[X(25)=0.404	Im[X(25)]=0.772
26: Re[X(26)=0.449	Im[X(26)]=1.118
27: Re[X(27)=0.466	Im[X(27)]=1.623
28: Re[X(28)=0.425	Im[X(28)]=2.443
29: Re[X(29)=0.238	Im[X(29)]=4.147
30: Re[X(30)=-0.931	Im[X(30)]=12.165
31: Re[X(31)=2.086	Im[X(31)]=-7.251

PHỤ LỤC 2: CHƯƠNG TRÌNH TÍNH DCT [x(i)]

```
Program DCT [x(i)];
    Uses Crt;
    const
    pi = 3.141592653589793238;
    n = 8;
    var
    i, u : interger;
    x, Y, TG, B : array[0..n-1] of real;
    Cu : real;
      Begin
      Writeln ('Nhập x(i) :');
      for i : = 0 to (n-1) do
          begin
             write('x[',i,'] =');            {Nhập các giá trị của x(i)}
                   readln (x[i]);
         end;
      for u : = 0 to (n-1) do
           begin
          Clrscr;
       if (u< > 0) then Cu: =1else Cu: =1/sqrt(2) ;
         TG[u]: = 0;
      for i : = 0 to (n-1) do
          begin
          B[i]: = cos(((2*i +1)*u*pi)/(2*n));
      TG[u]: = TG[u] + (x[i] * B[i, j]);
          end;
      Y[u]: = (TG[u] *sqrt(2/n)*Cu;
          end;
    writeln ('Ket qua:');
        for u : = 0 to (n-1) do
              writeln('y[',u,'] =',y[u]:6:3,';');
      readln;
      End.
```

PHỤ LỤC 3: CHƯƠNG TRÌNH VÀ KẾT QUẢ TÍNH DCT [x(i, j)]

```
Program DCT [x(i, j)];
    Uses Crt;
    const
    pi = 3.141592653589793238;
    n = 8;
    var
    i, j, u, v : interger;
    x, TG, B : array[0..n-1,0..n-1] of real;
    Y : array[0..n-1,0..n-1] of interger;
    Cu, Cv : real;
      Begin
      Writeln ('Nhập x(i, j) :');
      for i : = 0 to (n-1) do
       for j : = 0 to (n-1) do
         begin
           write('x[',i,',',j,'] ='); {Nhập cường độ sáng các pixels}
                readln (x[i, j]);
        end;
      for u : = 0 to (n-1) do
       for v : = 0 to (n-1) do
         begin
         Clrscr;
       if (u< > 0) then Cu: = 1 else Cu: = 1/sqrt(2) ;
       if (v< > 0) then Cv: = 1 else Cv: = 1/sqrt(2) ;
       y[u, v]: = 0;
      for i : = 0 to (n-1) do
       for j : = 0 to (n-1) do
         begin
         B[i, j]: = cos(((2*i +1)*u*pi)/(2*n))* cos(((2*j +1)*v*pi)/(2*n));
      y[u, v]: = y[u, v] + (x[i, j] – 128)* B[i, j];
         end;
      y[u, v]: = y[u, v] *(2/n)*Cu*Cv;
         end;
    writeln ('Ket qua:');
       for u : = 0 to (n-1) do
         for v : = 0 to (n-1) do
         writeln('y[',u,',',v,'] =',y[u, v]:6:3,';');
      readln;
      End.
```

KẾT QUẢ DCT [x(i, j)]

y[0,0] =-415	y[2,0] =-47	y[4,0] =12	y[6,0] =-1
y[0,1] =-30	y[2,1] =7	y[4,1] =-7	y[6,1] =0
y[0,2] =-61	y[2,2] =77	y[4,2] =-13	y[6,2] =0
y[0,3] =27	y[2,3] =-25	y[4,3] =-4	y[6,3] =-2
y[0,4] =56	y[2,4] =-29	y[4,4] =-2	y[6,4] =-1
y[0,5] =-20	y[2,5] =10	y[4,5] =2	y[6,5] =-3
y[0,6] = -2	y[2,6] =5	y[4,6] =-3	y[6,6] =4
y[0,7] = 0	y[2,7] =-6	y[4,7] =3	y[6,7] =-1
y[1,0] = 4	y[3,0] =-49	y[5,0] =-8	y[7,0] =0
y[1,1] =-22	y[3,1] =12	y[5,1] =3	y[7,1] =0
y[1,2] =-61	y[3,2] =34	y[5,2] =2	y[7,2] =-1
y[1,3] =10	y[3,3] =-15	y[5,3] =-6	y[7,3] =-4
y[1,4] =13	y[3,4] =-10	y[5,4] =-2	y[7,4] =-1
y[1,5] =-7	y[3,5] =6	y[5,5] =1	y[7,5] =0
y[1,6] =-9	y[3,6] =2	y[5,6] =4	y[7,6] =1
y[1,7] =5	y[3,7] =2	y[5,7] =2	y[7,7] =2

TÀI LIỆU THAM KHẢO

[1]. **Ричард Рид,** "Основы теории передачи информации", Вильямс, 2005 г., 320 стр.

[2]. **Darrel Hankerson, Greg A. Harris, and Peter D. Johnson Jr**, "Information Theory and Data Compression", CRC Press, 1997.

[3]. **А.А.Макаров, Л.А.Чиненков**, "Основы теории передачи информации", Новосибирск – 1998.

[4]. **В.И.Шульгин**, "Основы теории передачи информации", Харьков "хаи" – 2003.

[5]. **ITU** – Handbook spectrum monitoring Radiocommunication Bureau – edition 2002.

[6]. **William Stallings**, "Cryptography and Network Security – Principles and Practice", Second Edition, Prentice Hall, New Jersey 07458.

[7]. **Joan Daemen and Vincent Rijmen**, "The Design of Rijndael: AES - The Advanced Encryption Standard" Springer-Verlag, 2002.

[8]. **Đặng Văn Chuyết, Nguyễn Tuấn Anh** – Cơ sở lý thuyết truyền tin T1. NXB Giáo dục - 2001.

[9]. **Tổng cục bưu điện** – Vi ba số T1, NXB Bưu điện - 2002.

[10]. **LG information & communication, Ltd** – General introduction of telecommunication theory (Bản dịch tiếng Việt của Tổng cục bưu điện-NXB Thanh niên - 1995).

[11]. **Thái Hồng Nhị, Phạm Minh Việt** – Hệ thống viễn thông T1 & T2. NXB Giáo dục - 2001.

[12]. **Hồ Anh Túy** – Xử lý tín hiệu số – NXB Khoa học và Kỹ thuật - 1996.

[13]. **Nguyễn Thúy Vân** – Lý thuyết mã - NXB Khoa học và Kỹ thuật - 2000.

[14] **Trần Hồng Quân, Nguyễn Hữu Hậu** – Nguyên lý thông tin di động. NXB Bưu Điện - 2003.

[15]. **Đỗ Hoàng Tiến, Dương Thanh Phương** – Truyền hình kỹ thuật số. NXB Khoa học và Kỹ thuật - 2003.

[16]. **Dr. Alex, W. Lam and Dr. Saward Tantaratana** – Theory and Applications of Spread Spectrum Systems, IEEE, 1994

[17]. **Nguyễn Phạm Anh Dũng** – Giáo trình thông tin di động thế hệ ba NXB Bưu Điện - 2004.

NGUYỄN VĂN HẬU – NGUYỄN HIẾU MINH

Cơ sở lý thuyết truyền tin

Chịu trách nhiệm xuất bản : PGS.TS.TÔ ĐĂNG HẢI

Biên tập : ThS. NGUYỄN HUY TIẾN

QUANG NGỌC

Trình bày bìa : XUÂN DŨNG

NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

70 TRẦN HƯNG ĐẠO HÀ NỘI

In 200 cuốn, khổ 19 x 27 cm, tại Xưởng in NXB Văn hoá Dân tộc

Quyết định xuất bản số: 82 – 2008/CXB/277 – 02/KHKT – 14 – 1 - 2008

In xong và nộp lưu chiểu tháng 2 năm 2008

2 0 8 0 5 7
cơ sở lý thuyết truyền tin
020503000030
108,000

Giá: 108 000đ