HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN THANH BÌNH

XÂY DỰNG THUẬT TOÁN TRUYỀN DỮ LIỆU QUA KÊNH THOẠI CỦA MẠNG GSM VÀ ỨNG DỤNG THUẬT TOÁN SINH SỐ GIẢ NGẪU NHIÊN DỰA TRÊN CÁC DÃY PHI TUYẾN LỒNG GHÉP ĐỂ BẢO MẬT DỮ LIỆU

LUẬN ÁN TIẾN SỸ KỸ THUẬT

HÀ NỘI – 2022

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

NGUYỄN THANH BÌNH

XÂY DỰNG THUẬT TOÁN TRUYỀN DỮ LIỆU QUA KÊNH THOẠI CỦA MẠNG GSM VÀ ỨNG DỤNG THUẬT TOÁN SINH SỐ GIẢ NGẪU NHIÊN DỰA TRÊN CÁC DÃY PHI TUYẾN LỒNG GHÉP ĐỂ BẢO MẬT DỮ LIỆU

Chuyên ngành: Kỹ thuật điện tử

Mã số: 9.52.02.03

LUẬN ÁN TIẾN SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC:

GS. TSKH. NGUYỄN XUÂN QUỲNH

HÀ NỘI – 2022

LỜI CAM ĐOAN

Nghiên cứu sinh xin cam đoan đây là công trình nghiên cứu của chính mình. Các số liệu, kết quả trong luận án là trung thực và chưa từng được công bố trong bất cứ công trình của bất kỳ tác giả nào khác.

Người cam đoan

Nguyễn Thanh Bình

LỜI CẢM ƠN

Luận án tiến sỹ này được nghiên cứu sinh thực hiện tại Học viện Công nghệ Bưu chính Viễn thông dưới sự hướng dẫn khoa học của GS.TSKH Nguyễn Xuân Quỳnh. Nghiên cứu sinh xin được bày tỏ lòng biết ơn sâu sắc đối với GS.TSKH Nguyễn Xuân Quỳnh, TS. Lê Chí Quỳnh, GS.TS Nguyễn Bình, các thầy đã định hướng khoa học, chỉ dẫn thực hiện những nhiệm vụ cần thiết cũng như tạo mọi điều kiện thuận lợi để công trình nghiên cứu này được hoàn thành.

Nghiên cứu sinh xin được trân trọng cám ơn Ban Cơ yếu Chính phủ đã tạo điều kiện để nghiên cứu sinh hoàn thành nhiệm vụ nghiên cứu.

Nghiên cứu sinh cũng xin chân thành cảm ơn Lãnh đạo Học viện Công nghệ Bưu chính Viễn thông, Khoa Đào tạo sau đại học và các đồng nghiệp đã luôn hỗ trợ, tạo điều kiện để hoàn thành công trình nghiên cứu này.

Cuối cùng là sự biết ơn tới gia đình, bạn bè đã thông cảm, động viên giúp đỡ nghiên cứu sinh có thêm nghị lực để hoàn thành luận án này.

Hà Nội – 2022.

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

2G	Second Generation	Thế hệ hai
3G	Third Generation	Thế hệ ba
3GPP	Third Generation Partnership Project	Dự án đối tác thế hệ thứ 3
A		
AQAM	Adaptive Quadrature Amplitude Modulation	Điều chế biên độ thích nghi vuông góc
AuC	Authentication Center	Trung tâm xác thực
ACELP	Algebraic code-excited linear prediction	Dự đoán tuyến tính mã kích thích đại số
ADPCM	Adaptive Differential Pulse Code Modulation	Điều chế sung mã vi sai thích nghi
B		
BSC	Base Station Control	Điều khiển trạm gốc
BSS	Base Station System	Hệ thống trạm gốc
BTS	Base transceiver station	Trạm phát sóng cơ sở
C		
CCIP	Conditional cochannel interference probability	Nhiễu đồng kênh có điều kiện
CDMA	Code Division Multiple Access	Đa truy nhập phân chia theo mã
CN	Core Network	Mạng lõi
CS	Chanel Switching	Chuyển mạch kênh
CSD	Circuit Switched Data	Dữ liệu chuyển mạch
CELP	Code-excited linear prediction	Dự đoán tuyến tính mã kích thích
E		

EIR	Equitment Identity Register	Đăng ký nhận dạng thiết bị
F		
FDD	Frequency Division Duplex	Ghép song công phân chia theo tần số
FDMA	Frequence Division Mutiplex Access	Đa truy nhập phân chia theo tần số
FIR	Finite Impulse Response	Bộ lọc Đáp ứng xung hữu hạn
G		
GGSN	Gateway GPRS Support Node	Nút hỗ trợ GPRS cổng
GMSC	Gateway Mobile Service Center	Trung tâm chuyển mạch các dịch vụ
GPRS	General Packet Radio Service	Dịch vụ gói vô tuyến
GSM	Global System for Mobile Communication	Hệ thống toàn cầu cho truyền thông di động
H		
HLR	Home Location Register	Đăng ký Thuê bao – HLR
HF	High Frequency	3-30Mhz
I		
IMSI	International Mobile Subscriber Identity	Mã nhận dạng thuê bao di động quốc tế
IMT	International Mobile Telecommunications	Thông tin di động toàn cầu
IIR	Infinite Impluse Response	Bộ lọc đáp ứng xung vô hạn
IP	Internet Protocol	Giao thức chuyển mạch gói
ISI	Inter-Symbol Interference	Nhiễu giữa các ký hiệu
ITU	International Telecommunications Union	Hiệp hội Viễn thông Quốc tế

L		
LMS	Least Mean Square	Bình phương trung bình bé nhất
LPC	LinearPredictiveCoding	Mã hóa dự đoán tuyến tính
LSF	Line Spectral Frequencies	Tần số phổ vạch
LSD	Log Spectral Distortion	Méo dạng phổ loga
LSP	Line Spectrum Pairs	Cặp phổ vạch
LTE	Long-term evolution	Phát triển dài lâu
LTP	Long Term Predictor	bộ lọc dự đoán thời gian dài
LFSR	Linear Feedback Shift Register	Thanh ghi dịch phản hồi tuyến tính
LAI	Location Area Identity	Nhận diện vùng
M		
MIMO	Multiple-input and multiple-output	Đa đầu vào và đa đầu ra
MISO	Multiple Input single Output	Đa đầu vào đơn đầu ra
MS	Mobile Station	Trạm di động
ME	Mobile Equitment	Thiết bị di động
MSC	Mobile Switching Center	Trung tâm chuyển mạch
MELP	Mixed Excitation Linear Prediction	Thuật toán nén dự đoán tuyến tính kích thích hốn hợp
MPE	Multi Pulse Excited	Đa xung kích thích
N		
NGN	Next Genneration Netword	Mạng viễn thông thế hệ mới
NSS	Network Subsystem	Hệ thống mạng lõi
NMS	Network Management Subsystem	Hệ thống quản lý mạng
O		

OFDM	Orthogonal Frequency Division Multiplex	Ghép kênh phân chia theo tần số trực giao
P		
PLMN	Public Land Mobile Network	Mạng thông tin di động mặt đất
PCM	Pulse Code Modulation	Điều chế sung mã
PS	Packet switching	Chuyển mạch gói
PSTN	Public Switched Telephone Network	Mạng điện thoại chuyển mạch công cộng
PESQ	Perceptual Evaluation Speech Quality	Đánh giá chất lượng tín hiệu thoại theo tri giác
PN	Pseudo Noise	Giả ngẫu nhiên
Q		
QAM	Quadrature Amplitude Modulation	Điều chế biên độ vuông góc
QoS	Quality of Service	Chất lượng dịch vụ
QPSK	Quadrature Phase Shift Keying	Khoá dịch pha vuông góc
R		
RAN	Radio Access Network	Mạng truy nhập vô tuyến
RF	Radio Frequency	Tần số vô tuyến
RNC	Radio Network Controller	Bộ điều khiển mạng vô tuyến
S		
SCF	Service Control Function	Chức năg điều khiển dịch vụ
SGSN	Serving GPRS Support Node	Nút hỗ trợ GPRS dịch vụ
SISO	Single Input Single Output	Đơn đầu vào đơn đầu ra
SMS	Short Message Service	Dịch vụ tin nhắn di động
SDR	Software Define Radio	Vô tuyến điều khiển bằng phần mềm

SWR	Standing Wave Ratio	Tỉ lệ sóng đứng
SIM	Subcriber Identity Module	Module nhận diện thuê bao
STP	Short Term Predictor	Bộ lọc dự đoán thời gian ngắn
T		
TDD	Time Division Duplex	Ghép song công phân chia theo thời gian
TDMA	Time Division Multiplex Access	Đa truy nhập phân chia theo thời gian
TMSI	Temporary Mobile Subscriber Identity	Mã nhận dạng thuê bao tạm thời
U		
UE	User Equipment	Thiết bị người dùng
UMTS	Universal Mobile Telephone System	Hệ thống viễn thông di động toàn cầu
UTRAN	Universal Terrestrial Radio Access Network	Mạng truy nhập vô tuyến
V		
VLR	Visitor Location Register	Đăng ký đăng nhập vùng
Vbp	Band pass voicing	Giải thông thoại
VHF	Very High Frequency	Dải tần 30-300Mhz
VAD	Voice Activity Detectors	Nhận diện tín hiệu thoại
VSELP	Vector sum excited linear prediction	Dự đoán tuyến tính kích thích tổng vect
W		
WCDMA	Wideband Code Division Multiple Access	Đa truy nhập phân chia theo mã băng rộng
Z		
ZF	Zero Forcing	Cưỡng bức về không

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG BIỂU

PHẦN MỞ ĐẦU

I. TÍNH CẤP THIẾT CỦA LUẬN ÁN

Với sự bùng nổ của các thiết bị điện thoại thông minh, ngoài thông tin thoại, mạng di động đang được khai thác triệt để cho các ứng dụng giá trị gia tăng dựa trên các dịch vụ như SMS, thông tin trực tuyến, thậm chí cả dịch vụ nhạy cảm là thanh toán trực tuyến bằng điện thoại di động. Đi kèm với tiện ích là các lỗ hổng bảo mật, bảo mật cho thông tin thoại và dữ liệu người dùng từ thiết di động đến di động hay đến các thiết bị đầu cuối mạng cố định. Để tránh được các nguy cơ như bị nghe lén, lộ lọt thông tin, bị cài đặt các thành phần gián điệp để phục vụ cho nhiều mục đích bất hợp pháp là nhu cầu rất cần thiết trong các giao dịch thương mại và đặc biệt cấp thiết trong quốc phòng, an ninh. Bảo mật thông tin trong mạng di động đang trở thành một chủ đề nóng.

Mặc dù trong mạng viễn thông di động GSM vấn đề bảo mật và an toàn thông tin đã có, đã được xử lý qua quá trình sinh khóa, xác thực, mã hóa bằng các thuật toán chuẩn (A8, A3, A5) từ các thiết bị đầu cuối (MS) đến các trạm gốc (BSS). Tuy nhiên, đến phần mạng lõi thì các thông tin thoại vẫn là rõ, hơn nữa với các thủ đoạn đánh cắp tinh vi và tấn công nghệ cao của các đối tượng thù địch thì giải pháp và thuật toán xác thực, bảo mật trên là không bảo đảm và không triệt để cho mục đích bảo mật thông tin thoại, dữ liệu từ đầu cuối đến đầu cuối di động (MS to MS) hay đầu cuối di động đến các đầu cuối cố định (mạng PSTN) trong các giao dịch quan trọng, đặc biệt là trong quốc phòng, an ninh. Đây chính là mục tiêu và là tính cấp thiết của Luận án đặt ra.

Nếu như tất cả các hạ tầng mạng viễn thông di động được phủ kín sóng 3G/LTE và mạng truyền dẫn trên nền tảng IP, các giao dịch thông tin thoại, dữ liệu truyền dẫn trên cùng nền tảng này thì vấn đề bảo mật thông tin thoại và dữ liệu từ đầu cuối đến đầu cuối sẽ cơ bản thuận lợi, thực hiện dễ dàng. Tuy nhiên, thực tế một số mạng viễn thông di động ở các Vùng không phải chỗ nào cũng đã được phủ kín thế hệ 3G/LTE, đặc biệt ở Việt Nam tỷ lệ này còn nhiều. Ngoài ra, trong một số giao dịch thương mại

và đặc biệt là trong quốc phòng, an ninh thực tế vẫn đang triển khai cả mạng truyền dẫn PSTN, các mạng Satellite, sóng ngắn, sóng cực ngắn HF/VHF/UHF, vì những mạng truyền dẫn này mặc dù băng thông không lơn, nhưng có tính cơ động cao dễ triển khai, lắp đặt; độ bảo mật cao. Từ thực tiễn này, đòi hỏi cần phải có giải pháp và kỹ thuật để dễ dàng kết nối liên thông – bảo mật thông tin thoại và dữ liệu cho đa môi trường truyền dẫn trên để phục vụ cho mục đích quốc phòng an ninh và một số giao dịch thương mại đặc biệt. Đây chính là mục tiêu, giải pháp Luận án cần nghiên cứu giải quyết.

Ngoài ra, thuật toán sinh khóa, xác thực và mã hóa là các thuật toán chuẩn, không đủ mạnh để tin tưởng dùng cho mục đích bảo mật thông tin giao dịch thương mại chứ chưa nói đến thông tin quốc phòng an ninh của quốc gia. Từ đó đặt ra là cần nghiên cứu, xây dựng thuật toán đủ mạnh để bảo mật dữ liệu và tín hiệu thoại ở mức cao nhất, nhưng thuật toán đó độ phức tạp thực thi tương đối để phù hợp với ứng dụng cài đặt, chạy trên thiết kế có tài nguyên hạn chế. Đây cũng là bài toán khó, vì vừa phải bảo đảm độ mật ở mức cao nhất, độ phức tạp tính toán cao nhất, vừa phải bảo đảm tài nguyên hạn chế khi thực thi thuật toán.

Mặc dù đã có nhiều nghiên cứu, nhiều sản phẩm bảo mật thông tin thoại của các hãng trên thế giới (như Crypto AG, Motorola, Rohde & Schwarz, Secfone, Go-Trust, GSMK CryptoPhone,..) về chủ đề này, nhưng mới chỉ dừng lại ở phạm vi trên thiết bị đầu cuối cùng công nghệ truyền dữ liệu qua các kênh truyền dữ liệu (như 3G/LTE, CSD,..) và trên một mạng truyền dẫn (hoặc liên mạng thì lại cần một hệ thống gateway chuyển đổi chuyên dụng cho các mạng công nghệ khác nhau). Do vậy, cần phải giải quyết bảo mật thông tin thoại và dữ liệu được xử lý từ các thiết bị đầu cuối công nghệ khác nhau, truyền dẫn liên thông qua các mạng công nghệ khác nhau, *Và như vậy chủ đề nghiên cứu về bảo mật thông tin thoại (mã hóa dữ liệu số tín hiệu thoại) trên hệ thống các thiết bị đầu cuối bất kỳ và truyền liên mạng truyền dẫn vẫn là lĩnh vực mở và sẽ có nhiều cách tiếp cận, giải quyết khác nhau*. Xuất phát từ những lý do phân tích ở trên, Nghiên cứu sinh đã quyết định chọn đề tài **"Xây dựng thuật toán truyền dữ liệu qua kênh thoại của mạng GSM và ứng dụng thuật toán sinh số giả ngẫu**

nhiên dựa trên các dãy phi tuyến lồng ghép để bảo mật dữ liệu" cho luận án của mình.

II. MỤC TIÊU, ĐỐI TƯỢNG, PHẠM VI VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1.Mục tiêu nghiên cứu

Có 03 mục tiêu chính của luận án, đó là:

- Nghiên cứu, đề xuất giải pháp truyền dữ liệu thoại mã hóa hiệu quả trên các thiết bị đầu cuối đi qua các kênh thoại analog trên các liên mạng truyền dẫn viễn thông khác nhau; thực hiện mã hóa bảo mật thông tin thoại **Số** thông suốt từ thiết bị thoại đầu cuối đến đầu cuối trong các dịch vụ thoại và dữ liệu mạng di động các thế hệ 2G/3G/LTE và từ đầu cuối trên mạng di động đến máy điện thoại đầu cuối mạng PSTN đảm bảo chất lượng tiếng nói ở mức chấp nhận được sau giải mã, và phổ tần tín hiệu tiếng nói sau mã hóa tựa nhiễu trắng.
- Lựa chọn và xây dựng thuật toán đảm bảo độ tin cậy, tính khả thi về khả năng thực hiện thời gian thực thuật toán trên các thiết bị có tài nguyên tính toán hạn chế, nhưng phải bảo đảm độ phức tạp tính toán để đạt được Độ mật ở mức cao nhất.
- Sử dụng Kit thực thi mô tả thuật toán để chứng minh độ an toàn, bảo mật của thuật toán dựa trên các đặc tính tương quan, đồng sắc xuất, phân bố nhọn của dãy giả ngẫu nhiên tạo ra. Thực nghiệm thuật toán xử lý nén tính hiệu tiếng nói, mã hóa và điều chế để truyền dữ liệu đã được mã hóa bảo mật truyền qua kênh tiếng nói mạng GSM, không yêu cầu thay đổi cấu hình thiết bị đầu cuối đang dùng, không yêu cầu thay đổi dịch vụ mạng viễn thông đang dùng, đảm bảo tính dịch vụ liên mạng. Ghép nối các kết quả nghiên cứu đóng gói thành sản phẩm hoàn chỉnh.

2.2.Đối tượng nghiên cứu

Đối tượngnghiên cứu của Luận án này giới hạn ở các giải pháp truyền dữ liệu số tín hiệu thoại bảo mật qua kênh truyền analog bao gồm:

(i) Nghiên cứu Tổng quan về các mạng viễn thông di động: cơ chế đăng nhập, xác thực, bảo mật, các thuật toán mã hóa tiếng nói (Vocoder), đề xuất lựa chọn một thuật

toán nén thoại để áp dụng trong các kênh truyền bang hẹp, yêu cầu độ trễ thấp, tính toán thời gian thực.

(ii) Kỹ thuật xử lý tín hiệu thoại và mô hình mạng, các thông số kỹ thuật, đặc trưng cơ bản của các thành phần mạng dành cho xử lý và truyền dẫn tín hiệu thoại qua mạng và liên mạng: Nghiên cứu một số phương pháp điều chế và điều chế để tạo tín hiệu với phổ tần và đặc tính gần tín hiệu tiếng nói của con người để truyền qua kênh thoại mạng GSM và liên mạng GSM/PSTN/HF/VHF...

(iii) Phân tích, xây dựng và sử dụng dãy tạo số tựa ngẫu nhiên phi tuyến 2 chiều theo kiểu lồng ghép để mã hóa dữ liệu:

Nghiên cứu các phương pháp sinh dãy lồng ghép và lồng ghép phi tuyến, lựa chọn một phương pháp thực hiện lồng ghép phi tuyến đa chiều, để tạo ra dãy giả ngẫu nhiên với các thuộc tính có độ dài đủ lớn, độ phức tạp cao, hàm tương quan tốt và thực thi nhanh trên các vi xử lý có tài nguyên hạn chế.

(iiii) Tổng hợp các kết quả nghiên cứu, mô phỏng và đóng gói thành sản phẩm bảo mật thoại hoàn chỉnh có thể chứng minh ở mức Demo sản phẩm trên các kênh Voice của các thiết bị điện thoại thông thường để kiểm chứng chất lượng tiếng nói sau giải mã và chất lượng mã xem trên máy phân tích phổ sau mã hóa.

Phạm vi nghiên cứu

(i) Nghiên cứu về các phương pháp nén và các bộ mã tín hiệu tiếng nói; nghiên cứu về đặc điểm cơ bản mạng truyền dẫn thoại (tập trung vào mạng PSTN và GSM);

(ii) Nghiên cứu về phương pháp điều chế/giải điều chế dữ liệu;

(iii) Nghiên cứu mô hình toán học, xây dựng dãy PN phi tuyến có cấu trúc lồng ghép hai chiều. Đánh giá đặc tính của mã phi tuyến lồng ghép theo các tiêu chí hàm tương quan, kích thước tập hợp, khả năng ngẫu nghiên hóa, tốc độ sinh và mã hóa dữ liệu trên vi xử lý có tài nguyên hạn chế.

Phương pháp nghiên cứu

Phương pháp nghiên cứu là dựa trên các tài liệu, công trình nghiên cứu đã công bố; Dựa trên các nghiên cứu tổng hợp và phân tích các kết quả của nhóm nghiên cứu về hiện trạng mạng viễn thông di động tại Việt Nam, đặc tuyến tiếng nói của con người, về một số giải pháp bảo mật thông tin thoại hiện nay, so sánh với thực tế hạ tầng viễn thông, thực tế về yêu cầu bảo mật ở Việt Nam và trên thế giới. Bước đầu tiên, dựa vào các công cụ toán học, công cụ lập trình mô phỏng lý thuyết MATLAB trên PC để tạo lập bộ nén, điều chế biến đổi dữ liệu tựa ngẫu nhiên thành tín hiệu có đặc trưng và phổ tần tựa tiếng nói theo các phương pháp mã mã hóa; nghiên cứu Hàm vết và biến đổi *d* để xây dựng cấu trúc tổng quát của mã phi tuyến đa cấp theo kiểu lồng ghép và đánh giá các đặc tính cơ bản theo các tiêu chí trải phổ. Bước 2, dựa vào các kết quả mô phỏng trên máy tính PC đã đạt được chuyển hóa sang thực thi trên chip vi xử lý DSP hoặc ARM để nén tín hiệu tiếng nói, mã hóa và điều chế để biến đổi lại thành tín hiệu có đặc trưng tựa tiếng nói truyền trên các mạng truyền dẫn. Các kết quả nghiên cứu được kiểm chứng bằng mô phỏng và kết quả bằng sản phẩm thử nghiệm được trong thực tế.

Ý nghĩa khoa học và thực tiễn

Về mặt lý thuyết, luận án đã đề xuất phương pháp và xây dựng một kỹ thuật về điều chế dữ liệu tựa ngẫu nhiên (dữ liệu thoại sau nén đã được sử dụng dãy phi tuyến lồng ghép 2 chiều mã hóa) thành dạng tín hiệu tương tự có cấu trúc phổ tần gần giống với phổ tần của tiếng nói để tránh được các bộ phân tích và nhận dạng tiếng nói trên các thiết bị đầu cuối và trên các thiết bị trong hệ thống mạng viễn thông.

Về ý nghĩa thực tiễn, kết quả nghiên cứu đã đưa ra một phương pháp, một sản phẩm hoàn chỉnh để bảo mật thông tin thoại bằng kỹ thuật số đã được đưa vào ứng dụng trong ngành Cơ yếu của Việt Nam. Hướng phát triển tiếp có thể xây dựng giải pháp truyền dữ liệu mật được dấu dưới dạng tín hiệu giả thoại truyền trên các môi trường khác kênh GSM như PSTN, HF, Satellite, các mạng IP,..

III. CÁC KẾT QUẢ NGHIÊN CỨU ĐÃ ĐẠT ĐƯỢC

Các đóng góp khoa học của luận án bao gồm:

(i) Đề xuất một kiến trúc lồng ghép mới cho m-dãy lồng ghép (một phương pháp mới sinh dãy lồng ghép và lồng ghép phi tuyến, được công bố chi tiết trong bài báo [1b]) và xây dựng giải pháp bảo mật dữ liệu thoại sử dụng thuật toán sinh số giả ngẫu nhiên dựa trên dãy phi tuyến lồng ghép kiểu mới;

(ii) Đề xuất thuật toán cải tiến tốc độ nén, nâng cao chất lượng mã thoại MELPe (có công bố các nội dung liên quan trong bài báo [2b]);

(iii) Đề xuất thực hiện kỹ thuật điều chế và giải điều chế để truyền dữ liệu thoại đã được mã hóa bảo mật qua các thiết bị đầu cuối và mạng (liên mạng) truyền dẫn và đề xuất giải pháp truyền dữ liệu thoại bảo mật qua kênh thoại GSM, các kênh hữu tuyến và vô tuyến bang hẹp khác (được trình bày cụ thể trong bài báo [3b]).

(iiii) Tùy biến rút gòn để đưa được các chương trình thực thi nén, điều chế biến đổi tín hiệu số viết mô phỏng trên máy tính vào Vi xử lý STM32 chạy đầy đủ các tính năng như trên máy tính mà vẫn đáp ứng xử lý thời gian thực (đã đóng gói được thành sản phẩm).

IV. BỐ CỤC CỦA LUẬN ÁN

Cấu trúc của luận án gồm có ba chương với các nội dung được tóm tắt như sau:

Chương 1:Tổng quan về vấn đề nghiên cứu, Chương này trình bày tổng quan: về mạng viễn thông di động GSM; về an toàn, bảo mật và một số điểm yếu dễ bị tấn công trong mạng GSM, phân tích kỹ về vấn đề xác thực và bảo mật trong mạng; trình bày về đặc điểm tín hiệu tiếng nói cơ bản trong mạng GSM và mô hình tạo tiếng nói; về mã hóa, nén, truyền tín hiệu thoại qua các mạng khác nhau; đưa ra các định hướng giải quyết cho giải pháp truyền dữ liệu thoại mã hóa hiệu quả trên các thiết bị đầu cuối đi qua các kênh thoại analog trên các liên mạng truyền dẫn viễn thông khác nhau và giải pháp đảm bảo tính bảo mật cao thông tin thoại thời gian thực, không yêu cầu thay đổi cấu hình thiết bị đầu cuối đang dùng, không yêu cầu thay đổi dịch vụ mạng viễn thông đang dùng, đảm bảo tính dịch vụ liên mạng

Chương 2: Đề xuất thuật toán nén, đề xuất giải pháp bảo mật và truyền dữ liệu qua kênh thoại GSM. Chương này trình bày một số giải pháp bảo mật tín hiệu thoại di dộng phổ biến, đề xuất giải pháp bảo mật tối ưu nhất; lựa chọn bộ mã hoá dự đoán tuyến tính kích thích hỗn hợp MELP, mô tả, phân tích cải tiến thuật toán và đề xuất bộ mã hoá MELP cải tiến tốc độ thấp vào ứng dụng thực tế để nén thoại. Trình bày các giải pháp truyền dữ liệu qua kênh thoại GSM bao gồm các vấn đề về hạn chế kênh thoại GSM và các kênh truyền băng hẹp; Đề xuất một giải pháp điều chế và giải điều chế để truyền dữ liệu qua kênh thoại GSM và các kênh truyền thoại băng hẹp nói chung; thực nghiệm mô phỏng chứng minh chất lượng tiếng nói tái tạo sau truyền trên kênh và giải mật.

Chương 3: Bảo mật dữ liệu sử dụng thuật toán sinh số giả ngẫu nhiên dựa trên dãy phi tuyến hai chiều lồng ghép. Chương này tổng quan về m-dãy; cấu trúc và các tính chất dãy lồng ghép (tuyến tính và phi tuyến); thực hiện đánh giá mã phi tuyến lồng ghép cụ thể theo khả năng "ngẫu nhiên hóa" tín hiệu của mã. Đánh giá các phương pháp sinh dãy lồng ghép và lồng ghép phi tuyến; Đề xuất một phương pháp tính toán tối ưu để sinh dãy lồng ghép nói chung cho các thiết bị có tài nguyên hạn chế; đánh giá chứng minh sự tối ưu của phương pháp mới đề xuất. Đề xuất ứng dụng dãy lồng ghép phi tuyến trong kỹ thuật mật mã và thực thi thuật toán mã theo cấu trúc hai cấp. Thực thi thuật toán nén/giải nén Melpe và các thủ tục mã mật/giải mã bằng Vi xử lý ARM;

Kết luận: Phần này tổng kết các kết quả chính đã đạt được và hướng phát triển tiếp theo từ luận án này.

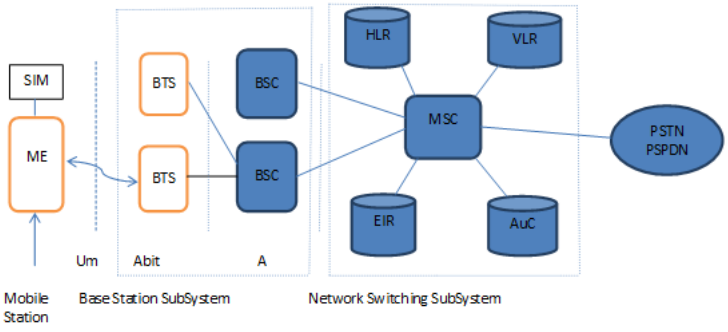
CHƯƠNG 1: TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU

Tóm tắt: Trong chương này, trình bày những thông tin cơ bản liên quan đến hệ thống thông tin di động; về vấn đề an toàn và bảo mật trong mạng GSM; về các phương pháp mã thoại nói chung và mã (nén) trong hệ thống GSM; cả nhiều phương pháp luận được tóm lược trình bày như động lực của vấn đề nghiên cứu.

1.1. Tổng quan về mạng viễn thông di động GSM [1][2][29][34]

Cấu trúc mạng di động GSM được chia thành 3 khối chính gồm: Hệ thống con MS, Hệ thống con trạm gốc BSS (bao gồm BSC và các BTS), Hệ thống con mạng lõi NSS; ngoài ra còn có 3 lớp giao diện: Um (giao diện vô tuyến giữa MS và BTS), Abit (là giao tiếp giữa BTS và BSC) và A giao tiếp giữa BSC và MSC (như ***Hình 1.1*** dưới). Trong đó, khối MS (bao gồm thiết bị máy điện thoại ME và Module SIM); khối BSS (gồm các BTS và BSC) là phần kết nối các mobile user với mạng lõi GSM; khối mạng lõi (gồm NSS và NMS), NSS là một tổng đài đầy đủ các khả năng định tuyến các cuộc gọi giữa các mobile user qua BSS và mobile user – thuê bao PSTN qua MSC và GMSC; NMS là hệ thống con thực hiện quản lý, giám sát các chức năng, các thành phần của toàn mạng. Các khối con cơ bản của hệthống được mô tả dưới đây:

Cấu trúc cơ bản mạng GSM:



Hình 1.1. Cấu trúc cơ bản mạng GSM

- *Module nhận thực thuê bao – SIM: Subscriber Identity Module*

Module này thực chất là một Smart Card chưa các thuật toán A3/8, số IMSI, khóa bí mật Ki và số điện thoại.

- *Thiết bị đầu cuối – ME: Mobile Equipment*

Thiết bị này hoạt động độc lập (tương đối) với các thiết bị mạng truyền dẫn, nó chưa thuật toán A5 và nó chỉ có thể kết nối với mạng GSM khi nó phải có SIM và nó không bao giờ biết được thuật toán A3/8 và Ki trên SIM.

- *Trạm thu phát gốc – BTS: Base Transceiver Station*

BTS là thành phần mạng thông tin di động mặt đất phục vụ cho các MS (Thiết bị ME khi được gắn Module SIM). Các trạm gốc sẽ kết nối liên thông các Cells sóng vô tuyến trong khu vực và các trạm gốc được kết nối đến trạm điều khiển MSC của vùng đó.

- *Trạm điều khiển gốc – BSC: Base Staition Controller*

BSC thực hiện chức năng điều khiển điều khiển các BTS quanh nó, như: điều khiển công suất phát cho các MS; cấp phát tần số kênh cho các BTS và MS; điều khiển nhảy tần số của các MS khi bị nhiễu...

- *Trung tâm chuyển mạch – MSC: Mobile Switching Center*

MSC là một nốt (node) điều khiển một số BSC. Nó là thiết bị trung tâm và có nhiều chức năng trong hệ thống GSM.Nó thực hiện chuyển mạch, xác thực, đăng ký và liên kết giữa các nốt (node). Nó còn được kết nối đến mạng PSTN.

- *Đăng ký Thuê bao – HLR: Home Location Register*

HLR là cơ sở dữ liệu của các thuê bao di động được tạo ra, bị chặn hay bị xóa bởi nhà cung cấp dịch vụ mạng. HLR chứa tất cả các thông tin lâu dài của các thuê bao cũng như các dịch vụ, giới hạn dịch vụ thuê bao được phép sử dụng, bao gồm: Số IMEI, bản copy Khóa bí mật Ki, VLR hiện tại của thuê bao và các dịch vụ hiện tại của thuê bao di động.

- *Đăng ký đăng nhập vùng - VLR: Visitor Location Register*

Cũng là một CSDL để chứa thông tin thuê bao của di động. Thông tin trong VLR cho biết vị trí hiện tại, trạng thái của MS... và VLR cung cấp dữ liệu của thuê bao để xử lý cuộc gọi bất cứ khi nào được yêu cầu. Thông tin trong VLR được cập nhật thường xuyên theo Cell MS đang tham gia, việc cập nhật thông tin trong VLR xảy ra khi: MS bật máy, MS di chuyển sang BTS hay BSC khác và theo định kỳ;

- *Trung tâm xác thực thuê bao – AuC: Authentication Center*

AuC có chức năng kết hợp với HLR để cung cấp thông tin cho VLR các thông số xác thực một thuê bao MS có quyền truy nhập vào mạng hay không và các quyền dịch vụ của nó, sinh số ngẫu nhiên (RAND). Cơ sở dữ liệu của AuC lưu các thông tin: danh sách các thiết bị ME chuẩn quốc tế (IMSI), các thông tin xác thực như khóa bí mật Ki, thông tin định danh Vùng LAI (*Location Area Identity)*, định danh thuê bao tạm TMSI - *Temporary Mobile Subscriber Identity,*

- *Đăng ký nhận dạng thiết bị - EIR: Equitment Identity Register*

EIR lưu giữ một CSDL để giám sát toàn bộ các ME có IMSI sử dụng trên mạng mạng đó (mỗi mạng chỉ có một EIR).

1.2. An toàn, bảo mật và một số điểm yếu về vấn đề này trong hệ thống mạng GSM *[2,3,6,7,16,29,31,34]*

Ban đầu GSM được dự định là một hệ thống không dây an toàn. Để đạt được mục đích này, hệ thống đã được xây dựng bằng các cơ chế:

• Xác thực người dùng, xác thực dịch vụ di động

• Mã hóa các thông tin trao đổi trên môi trường radio.

Xác thực người dùng bằng cách sử dụng khóa được chia sẻ trước, có cơ chế mã hóa qua mạng. GSM dễ bị tấn công bởi các loại tấn công khác nhau, mỗi loại đều nhắm vào một phần khác nhau của mạng, GSM chỉ xác thực người dùng với mạng. Mô hình cung cấp bảo mật và xác thực, nhưng không có khả năng chống chối bỏ. GSM sử dụng một số thuật toán A5/(1, 2, 3) chỉ để mã hóa đảm bảo bảo mật thoại trên kênh vô tuyến, trong khi giọng nói được truyền ở dạng rõ qua mạng lõi dưới dạng PCM và ADPCM [31]. A5 có một số hạn chế về mật mã và không thể liên lạc an toàn [2][6]. Do đó, A5 không thể cung cấp bảo mật cuộc gọi thoại hoàn toàn cho khách hàng GSM. Hơn nữa, người dùng không có quyền kiểm soát bảo mật mã hóa được kiểm soát bởi nhà cung cấp mạng và nhà sản xuất điện thoại di động

1.2.1. Nguyên lý xác thực và bảo mật trong mạng di động GSM

Một số chức năng xác thực và bảo mật đã được tích hợp vào GSM, bao gồm:

- Xác thực chủ thể thuê bao đăng ký

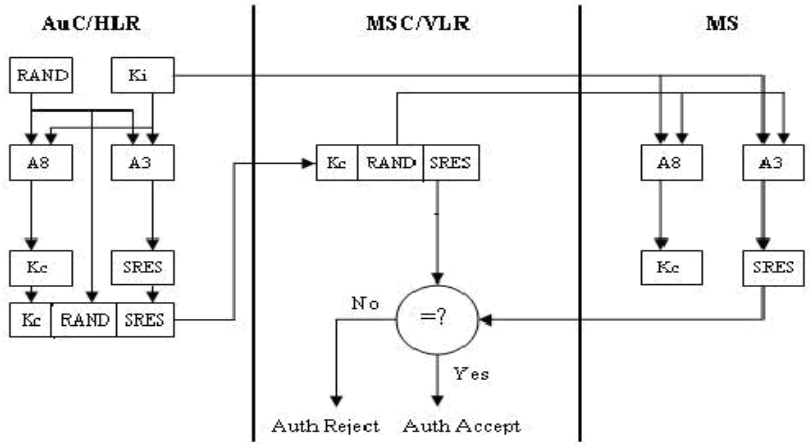
- Sử dụng mã hóa để đảm bảo bí mật thông tin trao đổi
- Bảo vệ định danh của thuê bao
- SIM (Mô dun nhận dạng thuê bao) được bảo vệ bằng mã số PIN
- SIM bị nhân bản không được cho phép gia nhập mạng đồng thời với SIM gốc
- Mã số bí mật Ki được bảo vệ an toàn.

Hệ thống GSM đảm bảo an toàn bảo mật bằng nhiều thuật toán với các loại thiết bị khác nhau. 2 nguyên lý chính là xác thực và mã hóa dữ liệu người dùng:

- ***Xác thực người dùng đăng nhập mạng (Au)***

Trong mạng GSM, việc xác thực thuê bao (xác thực người sử dụng) là đầu tiên, tiếp sau là xác thực các dịch vụ đã được chấp nhận. Với mỗi MS khi khởi tạo lần đầu, hoặc chuyển đến vùng BS mới, nó sẽ yêu cầu BTS cấp kênh truyền, khi có kênh truyền đã thống nhất với BTS, nó sẽ gửi yêu cầu cập nhật thông tin mạng vùng hiện tại đến MSC qua BSC. MSC sẽ trả lời nếu MS xác thực đúng.

Trong toàn bộ quá trình xác thực, có 3 nhân tố chính đó là: MS, MSC/VLR và HLR/AuC như trong ***Hình 1.2 [3]***.



Hình 1.2. Quá trình xác thực trong mạng GSM

Quá trình xác thực cụ thể như sau: đầu tiên MS gửi số IMSI (lấy từ SIM) gửi đến VLR qua trạm BTS gần nhất, để báo hiệu cần xác thực. VLR sẽ gửi IMSI đến trung tâm HLR/AuC. Trung tâm HLR/AuC dựa trên số IMSI sẽ tra cơ sở dữ liệu tìm ra khóa bí mật Ki của IMSI, tiếp đó sử dụng thuật toán xác thực (**A3**) và thuật toán sinh khóa mã (**A8**) để tạo ra khóa mã theo phiên (**Kc**) và kết quả ký được gọi là SRES.

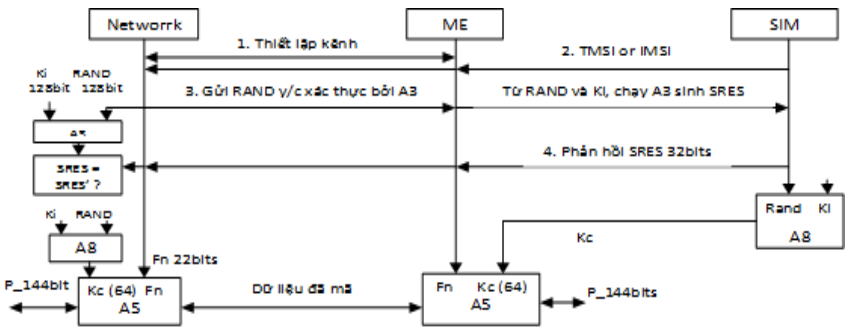
Tiếp theo, HLR sinh ra một số ngẫu nhiên RAND 128bit và gửi bộ 3 (Kc, SRES, RAND) về VLR. Tiếp theo, VLR lấy số ngẫu nhiên RAND 128bit gửi lại cho MS để yêu cầu MS tính ra số SRES và gửi quay trở lại. Cũng từ khóa bí mật Ki trên SIM và RAND nhận được, bằng các thuật toán **A3/8** MS tính ra Kc và SRES. MS sẽ dùng Kc làm khóa cho phiên làm việc và gửi SRES quay về VLR để xác thực. VLR sẽ so sánh 2 SRES của MS và HLR, nếu trùng nhau thì xác thực thuê bao thành công, và MS sẽ được cấp quyền truy nhập mạng.

Tuy nhiên nếu nhìn qua cơ chế xác thực trên, có thể thấy số IMSI được gửi trong bước một của quá trình xác thực, và nếu lấy được số này, hacker xem như sẽ có được 50% thông tin cần thiết để nhân bản SIM (số còn lại cần lấy là mã Ki).

Xác thực người dùng và các dịch vụ, mã hóa dữ liệu thoại, tin nhắn và Data trong các thế hệ với các công nghệ khác nhau của mạng GSM là có khác nhau, như xác thực và mã hóa trong chế độ UMTS là khác so với trong GSM/GPRS. Tuy nhiên trong khuôn khổ của Luận án chỉ tập trong vào công nghệ chuyển mạch kênh (CS) để truyền tín hiệu thoại trên kênh GSM, nên nội dung Luận án sẽ không phân tích các chế độ khác.

- ***Mã hóa dữ liệu người dùng như thoại, tin nhắn, Dữ liệu người dùng***

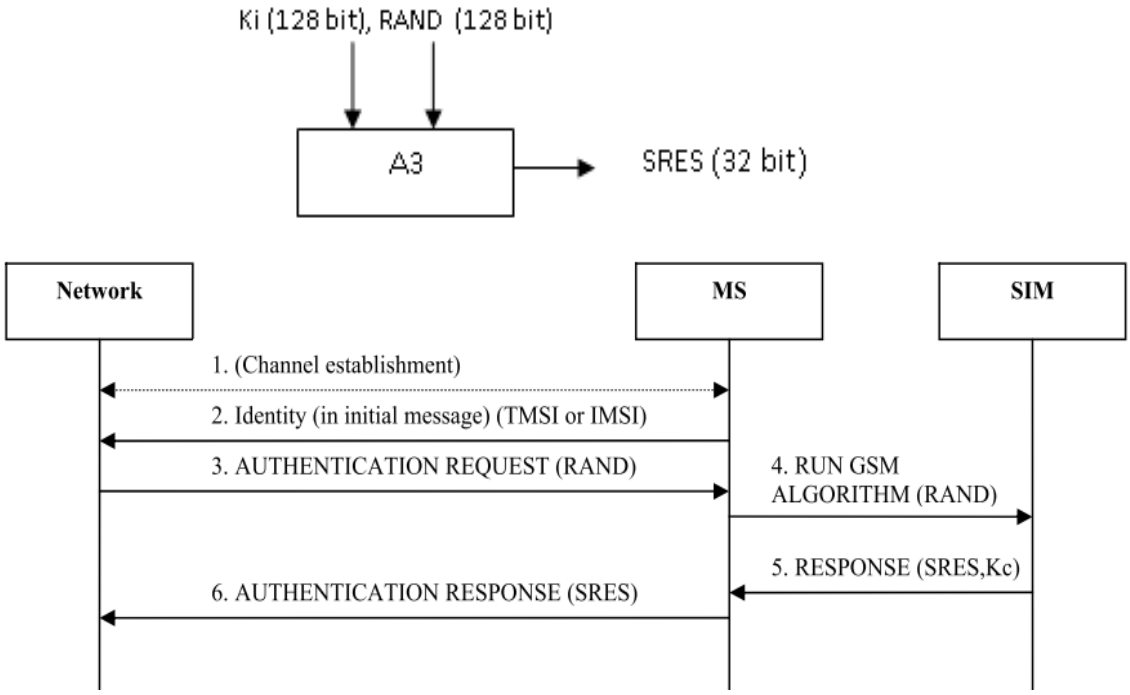
Như trên đã mô tả, khóa phiên mã hóa Kc được sử dụng cho cả MS và hệ thống mạng lõi sử dụng thuật toán mã hóa **A5** để mã/giải mã thông tin dữ liệu người dùng. Việc mã hóa này không được thực hiện bởi Module SIM vì không đủ năng lực xử lý, mà được thực hiện trên ME. **Hình 1.3** [**3**] dưới đây mô tả toàn bộ quá trình xác thực, sinh khóa và mã hóa:



Hình 1.3. Toàn bộ quá trình xác thực, sinh khóa và mã hóa trong mạng GSM

Trước hết tìm hiểu về các thuật toán A3, A8, A5:

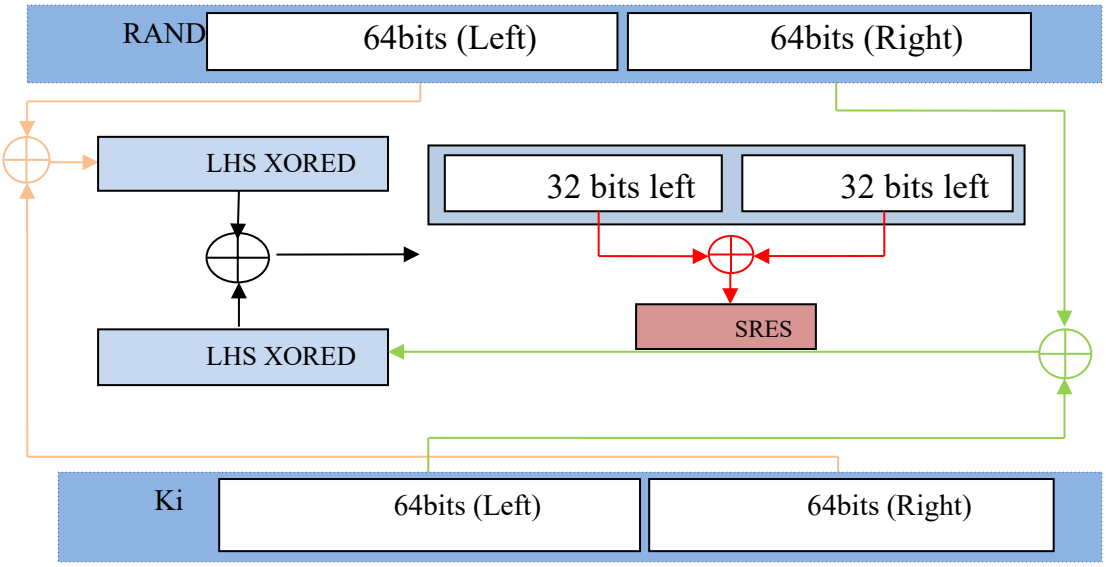
- Thuật toán A3 [3]: Sơ đồ khối thuật toán A3 như Hình 1.4



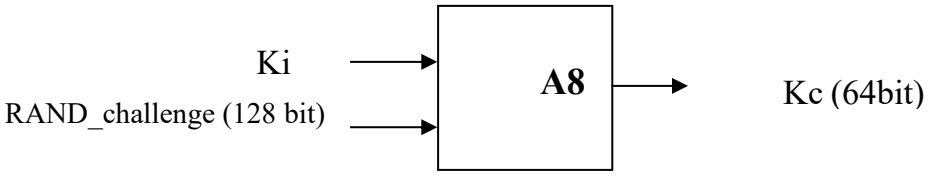
Hình 1.4. Mô hình thuật toán A3

Sơ đồ thực hiện các hàm chức năng trong thuật toán A3 để sinh ra SRES (32bit) được mô tả trong Hình 1.5, dưới đây:

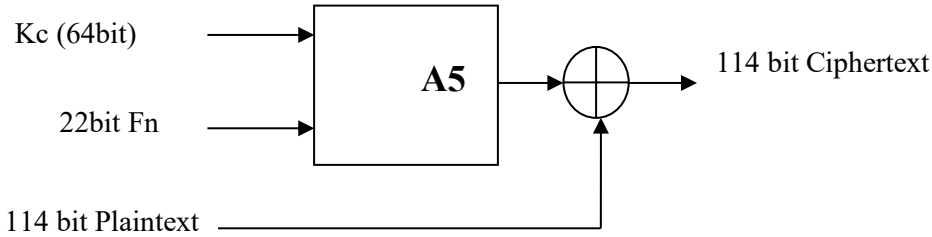
Nhìn vào lược đồ thuật toán trên thì thấy thuật toán rất dễ bị phá.



Hình 1.5. Sơ đồ khối các hàm thực hiện thuật toán A3



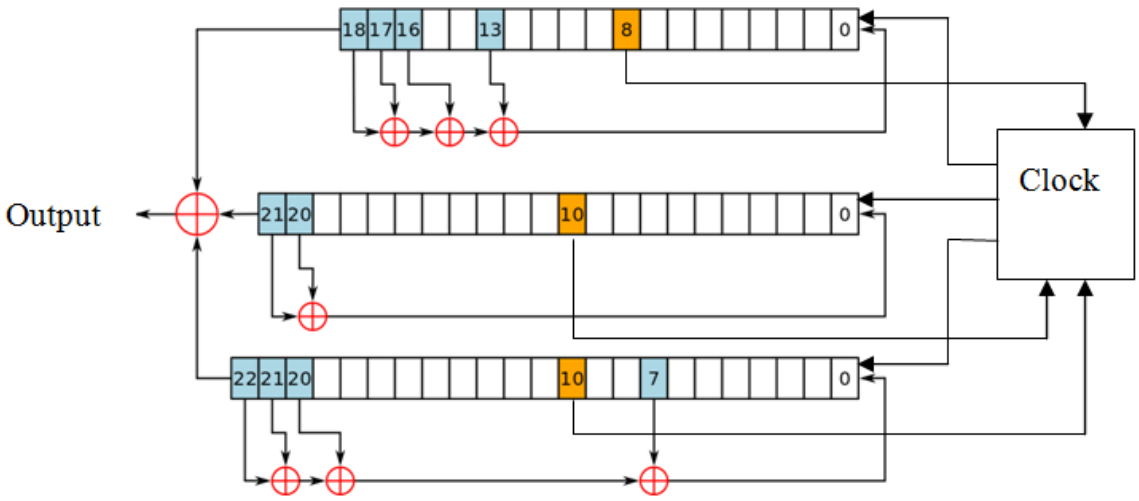
Hình 1.6. Sơ đồ khối thuật toán mã A8



Hình 1.7. Sơ đồ khối thuật toán A5

- Thuật toán A5: Sơ đồ khối thuật toán A5 như hình 1.7

A5 là một thuật toán mã dòng (*tạo ra dãy số giả ngẫu nhiên chất lượng tốt để XOR với dữ liệu cần mã – Sơ đồ khối thuật toán A5 như Hình 1.7*), sử dụng các thanh ghi dịch phản hồi tuyến tính. Có nhiều cách thực thi thuật toán A5, nhưng hầu như chỉ có 4 phiên bản A5/0, A5/1, A5/2 và (A5/3 sử dụng cho các hệ thống 3G), trong đó A5/1 được biết là mạnh nhất và sử dụng rộng ở Châu âu và châu mỹ, A5/2 được sử dụng ở Châu Á. Trong A5/1 sử dụng 3 thanh ghi dịch LFSR (R1, R2, R3 hay gọi là m_dãy) như Hình 1.8 dưới:



Hình 1.8. Sơ đồ khối thuật toán mã dòng A5 sử dụng 3 thanh ghi dịch phản hồi tuyến tính LFSR

Như hình trên, 3 thanh ghi dịch phản hồi tuyến tính trên có độ dài lần lượt là R1= 19, R2= 22 và R3=23 bits và tương ứng với 3 biểu thức trong **Bảng 1.1** sau:

Bảng 1.1. Các thanh ghi LFSR

LFSR	Độ dài (bit)	Đa thức phản hồi	Bít nhịp	Khai thác bit
1	19	$x^{19}+x^{18}+x^{17}+x^{14}+1$	8	13,16,17,18
2	22	$x^{22}+x^{21}+1$	10	20,21
3	23	$x^{23}+x^{22}+x^{21}+x^{8}+1$	10	7,20,21,22

Nguyên tắc hoạt động bộ 3 thanh ghi dịch phản hồi tuyến tính hoàn toàn theo nguyên tắc hoạt động của các m_dãy. Ở mỗi thanh ghi dịch sau mỗi chu kỳ xung nhịp, các bít có trọng số là '1' của đa thức (bít khai thác) được XOR với nhau và kết quả lưu vào bít có trọng số thấp nhất (bit Zero). Cả 3 thanh ghi đều có 1 bít cố định (*thanh ghi 1 là bít thứ 8 (c1), thanh ghi 2 và 3 là các bít thứ 10 (c2,c3), gọi là "Clocking bit")* để xác định dịch hay không dịch, phụ thuộc vào sự tính toán của hàm chức năng trong khối Clock, nếu dịch thanh ghi thì toàn bộ thanh ghi đó được dịch trái 1 bít, giá trị bít 0 được bù bằng bit từ khối Clock. Ở mỗi chu kỳ xung nhịp, từ giá trị các bit ở vị trí *c1, c2, c3,* hàm chức năng theo công thức (1.1) [7] để tìm ra giá trị bit (gọi là majority bit)

$$c1 * c2 \oplus c2 * c3 \oplus c1 * c3 \qquad (1.1)$$

Căn cứ vào *c1, c2, c3* và giá trị *Majority bit*, các thanh ghi r1, r2, r3 sẽ được dịch tương ứng với giá trị c_i = *Majority bit* [*7*]. Cụ thể các thanh ghi được dịch theo bảng trạng thái **Bảng 1.2** sau:

Bảng 1.2. Bảng trạng thái thực hiện công thức (1.1)

c1	c2	c3	*Majority bit*	r1	r2	r3
0	0	0	0	Y	Y	Y

0	0	1	0	Y	Y	N
0	1	0	0	Y	N	Y
0	1	1	1	N	Y	Y
1	0	0	0	N	Y	Y
1	0	1	1	Y	N	Y
1	1	0	1	Y	Y	N
1	1	1	1	Y	Y	Y

Quá trình thực thi A5 trong 4 bước như sau:

- Cả 3 thanh ghi (R1, R2, R3) được xóa về '0', tiếp theo xử lý song song ***64bit Kc*** từ LSB – MSB (Least Significant Bit – Most Significant Bit) được XOR với bit LSB của cả 3 thanh ghi (như vậy cần 64 chu kỳ xung nhịp, không sử dụng xung điều khiển ở pha này).
- Giá trị hiện tại của 3 thanh ghi ở bước trên, tiếp tục xứ lý song song với giá trị ***22bit Fn*** từ LSB – MSB được XOR với bít LSB của cả 3 thanh ghi (như vậy cần thêm 22 xung nhịp, không sử dụng xung điều khiển ở pha này). Giá trị các bít nhị phân hiện tại trên cả 3 thanh ghi này gọi là trạng thái khởi đầu của các thanh ghi dịch.
- 3 thanh ghi được kích hoạt thêm 100 xung nhịp và có các xung điều khiển, nhưng không sử dụng các giá trị đầu ra.
- 3 thanh ghi được kích hoạt thêm 228 xung nhịp với các xung điều khiển, tương ứng đầu ra sẽ có 228 bít. 228 bít này được chia làm 2 cho mỗi chiều/kênh (114bit cho mã hóa chiều MS-BTS, 114 bit cho giải mã chiều BTS-MS). Tín hiệu trên kênh GSM được tổ chức theo chuỗi, trong một kênh và theo một hướng, mỗi mẫu được gửi sau mỗi 4,615 mili giây chứa 114 bit thông tin.

1.2.2. Điểm yếu của bảo mật trong mạng di động GSM và một số tấn công phổ biến:

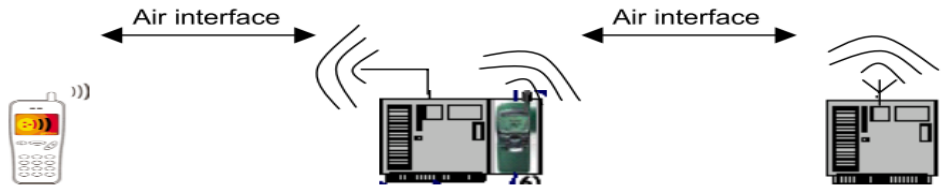
Như trên đã đề cập, mạng di động GSM không phải là một hệ thống an toàn hoàn hảo. Do đặc thù của cơ chế dùng sóng radio để liên lạc giữa thiết bị di động đầu cuối và trạm thu phát sóng, mạng GSM có những rủi ro bảo mật như:

- Tấn công giả mạo thiết bị di động đầu cuối
- Nghe lén cuộc gọi
- Tấn công dùng phương thức người thứ ba đứng giữa (man in the middle attack).

Bằng các thuật toán A3/A8/A5 để: xác thực, mã hóa các thông tin. Trong cơ chế bảo mật GSM, các thuật toán A3, A5, A8 đều được giấu kín. Tuy nhiên, phương thức bảo mật bằng cách giấu thuật toán này sẽ không an toàn. Lý do là một thuật toán cho dù tốt đến đâu cũng có thể mắc lỗi và nếu không được công khai để cộng đồng kiểm chứng thì hoàn toàn có thể bị mắc những lỗi nghiêm trọng. Thực tế đã chứng minh là dù được nhà sản xuất cố gắng giữ bí mật sau nhiều năm, hacker đã tìm được thông tin khá đầy đủ về các thuật toán A3, A5 và A8 (Cụ thể vào tháng 8 năm 1999, Một nhóm các nhà nghiên cứu Mỹ công bố khả năng phá thuật toán A5/2 bằng máy tính PC bình thường, thời gian phá mã là vài giây. Vào tháng 12 năm 1999, hai nhà nghiên cứu Israel công bố khả năng phá mã A5/1 trong vòng 2 phút sau khi lắng nghe cuộc gọi. Vào tháng 2 năm 2008, tại đại hội BlackHat (DEFCON-15), hai nhà nghiên cứu Hulton và Steve trình bày khả năng phá bảo mật GSM với giá rẻ! Hacker hiện nay có thể chế tạo thiết bị nghe lén GSM với giá chỉ vài chục ngàn đô la với khả năng giải mã cuộc gọi trong thời gian 30 giây). Như vậy với công nghệ, kỹ thuật hiện nay, thì người thứ 3 hoàn toàn có thể nghe xen giữa trên giao diện Um, hoặc các nhà mạng có thể xen vào qua các giao diện A hoặc Abit. Như trên đã nói, xét về tấn công mật mã thì các thuật toán A3,A8 là rất yếu và thậm trí cả thuật toán A5/1, cho dù các mạng có thể sử dụng phiên bản thuật toán mã tốt nhất (A5/1, khóa Kc cũng chỉ là 64bits). Ngoài ra, hệ thống cũng chỉ có xác thực một chiều BTS-MS mà không có xác thực chiều ngược lại, nghĩa là một trạm BTS giả thực hiện các thao tác giống như BTS thật, thì MS vẫn trả về các giá trị SRES như thường; việc xác thực ID (người gọi, người gửi) dữ liệu và ID được truyền trên các kênh khác nhau, IMSI gửi dưới dạng rõ ở phiên kết nối đầu tiên, đây là lỗi cơ bản của hầu hết các nhà mạng, nó cho phép một tấn công xen giữa. Trước đây để thiết kế một trạm BTS là rất khó và phức tạp, nhưng với công nghệ hiện nay thì giá rất vừa phải, dẫn đến những tấn công trên kênh vô tuyến là khả thi không phải chỉ các tổ chức đặc biệt.

Một số tấn công điển hình như:

a. Tấn công sử dụng trạm BTS giả mạo:



Máy đối tượng Trạm BTS giả Trạm BTS thật

Hình 1.9. Mô hình tấn công giả lập BTS

Hình thức này sử dụng một trạm BTS đã được sửa đổi để lấy thông tin từ máy di động bị tấn công khi thiết bị tấn công bị đánh lừa kết nối tới trạm BTS giả. Sau đó trạm BTS giả chuyển tiếp nội dung tới trạm BTS thật (như **Hình 1.9**).

b. Tấn công nặc danh người dùng

Lưu trong SIM ngoài IMSI còn có TMSI, nó như một biệt danh của thuê bao trong vùng. Kẻ tấn công có thể cần sự di chuyển của thuê bao và/hoặc theo vết các cuộc gọi để biết được IMSI và TMSI của MS, thông tin này cũng có thể sử dụng để xác định định danh một người cụ thể, khi đó giả mạo người sử dụng là rất tệ hại.

c. Tấn công dựa trên thuật toán xác thực

Nhiều nhà mạng GSM xử lý các thuật toán xác thực và sinh khoá MoU/COMP128 thay vì A3/A8. Nguyên nhân này bắt nguồn từ các SIM cũ, lý do khác là việc thay đổi hay sửa lại thuật toán là sự chi phí thay đổi phần mềm trong CSDL. Mặc dù thuật toán COMP128 không công khai, tuy nhiên đây là thuật toán yếu, đã bị dịch ngược và phân tích mã (ngày nay có thể dễ dàng tìm thấy trên Internet các phần mềm này). Bằng cách lấy được IMSI và Ki (qua dịch ngược được COMP128), việc Copy vào 1 SIM trống (mua ngoài thị trường) là khả thi, khi đó kẻ tấn công thực hiện các thủ tục xác thực với mạng như một thuê bao hợp lệ, do đó có thể thay đổi các cuộc gọi, thậm chí với khoá Ki đã có kẻ tấn công có thể giải mã và nghe toàn bộ các nội dung từ thuê bao và đến thuê bao.

Việc sao chép dữ liệu (IMSI và Ki) được thực hiện theo 2 cách: Clone Modul SIM vật lý và qua kênh vô tuyến.

d. Tấn công thuật toán mã (A5)

Kiểu này có 3 kiểu tấn công cơ bản là Tấn công phân tích mã; Tấn công phân tích không mã; Tấn công vét cạn.

- Tấn công vét cạn (brute-force): mặc dù khoá mã Kc là 64bit, nhưng 10bit cuối được điền toàn '0', do vậy không gian khoá giảm từ 2^{64} còn 2^{54}. Như đã nói ở trên, A5/1 là phiên bản mạnh nhất của thuật toán A5; A5/2 yếu hơn, và đã bị phá trong thời gian thực với một hệ số xấp xỉ 2^{16}, A5/1 mạnh nhưng cũng tấn công được với một hệ số xấp xỉ 2^{40} (với cấu hình máy tính Pentium 4, 3.2Ghz tấn công A5/1 trong 9 giờ

- Tấn công phân tích mã: Với các thuật toán mã hóa dữ liệu A5/1, A5/2 có một vài tấn công các thuật toán này, hầu hết các tấn công này yêu cầu kẻ tấn công phải biết các phần của khóa dòng, từ đó nó có thể biết các phần nhỏ bản tin rõ. Với phương pháp tấn công này thì kẻ tấn công phải biết được đủ số lượng các bản tin rõ theo yêu cầu.

Với thuật toán A5/1, các cuộc tấn công vào A5/1 gần đây chỉ tấn công dựa trên bản mã, điểm ấn tượng là việc tấn công chỉ yêu cầu biết một số lượng nhỏ các khung dữ liệu đã mã, tuy nhiên đây là một tấn công cần năng lực tính toán lớn. **Bảng 1.3** dưới đây đưa ra năng lực tính toán cần để tấn công chỉ dựa trên bản mã của thuật toán A5/1 [2]

Bảng 1.3. Năng lực tính toán cần để tấn công của thuật toán A5

Dữ liệu mã có được	**Các bước tiền xử lý**	**Số máy tính cần để xử lý trong 1 năm**	**Dung lượng ổ đĩa để lưu trữ**	**Chu kỳ**	**Số máy tính để tấn công trong thời gian thực**
2^{12} (≈ 15min)	2^{52}	140	22GB	2^{28}	1

$2^{6.7}$ (≈ 8s)	2^{41}	5000	176GB	$2^{32.6}$	1000
$2^{6.7}$ (≈ 8s)	2^{42}	5000	350GB	$2^{30.6}$	200
2^{14} (≈ 20min)	2^{35}	35	3GB	2^{30}	1

1.2.3. Một số phương pháp bảo mật thông tin thoại di động [6][16][28][29]

Để bảo mật cho thông tin thoại di động là tương đối đa dạng, về phạm vi tạm chia ra làm 2 đối tượng: thiết bị di động của người dùng và trách nhiệm của nhà mạng:

Đối với nhà mạng, có thể có các giải pháp sau: sử dụng thêm thuật toán an toàn cho thực thi các thuật toàn A3/A8 (SIM mới, cập nhật Software cho HLR); Sử dụng thuật toán mã hóa an toàn nhất (A5/1); đảm bảo an toàn các kênh truyền mạng lõi và cuối cùng là mã hóa đầu cuối – đầu cuối (E2E) độc lập với nhà mạng, đây là giải pháp an toàn nhất và nó thuộc phạm vi thiết bị đầu cuối.

Đối với phạm vi thiết bị di động người dùng, về công nghệ cũng tương đối nhiều phương pháp:

Bảo mật hoàn toàn bằng kỹ thuật phần cứng (chẳng hạn như các dòng điện thoại có bảo mật của hãng Motorola, Crypto AG; các dòng điện thoại di động sử dụng trong quân sự của nhiều nước như Nga, khối NATO) hoặc ở dạng một thiết bị bảo mật đường truyền, bảo mật dữ liệu âm thanh sử dụng kết hợp với điện thoại di động qua Bluetooth (của hãng R&S). Kết hợp giữa sử dụng phần cứng và phần mềm. Sử dụng hoàn toàn giải pháp bảo mật bằng phần mềm. Nhóm 2 và nhóm 3 chủ yếu áp dụng trong bảo mật điện thoại di động thông minh (smart-phone), phương thức truyền dữ liệu mật dựa trên nền tảng IP thông qua mạng 3G/4G, giải pháp phần mềm đơn giản, nhưng có độ an toàn, bảo mật rất kém.

Đối với các kỹ thuật bảo mật thông tin thoại và dữ liệu trên kênh thoại GSM, trên thế giới có rất nhiều mô hình giải pháp và công nghệ khác nhau. Tuy vậy có thể chia thành hai nhóm giải pháp công nghệ lớn đó là nhóm giải pháp dựa trên nền tảng tương tự (Scramblers) và nhóm giải pháp dựa trên nền tảng số (Digital Voice Protection – các tham số của tín hiệu thoại được lấy mẫu và biến đổi về dạng số thông qua bộ vocoder, sau đó tiến hành mã mật dữ liệu). Đối với các giải pháp bảo mật thoại trên

nền tảng tương tự Scramblers thường sử dụng các phương pháp mã tín hiệu thoại cơ bản như:

Dạng (1): Xáo trộn theo miền thời gian (TDS - Time Domain Scramblers) [32]

Dạng (2): Xáo trộn trong miền tần số (Frequency Domain Scramblers – FDS)

Dạng (3): Sự kết xáo trộn tần số và thời gian (Time Frequency Scrambling - TFS),

Dạng (4): Mã bằng phương pháp sử dụng các chuỗi Pseudo Noise (Encryption by using Pseudo Noise Sequences – ENS).

Bảo mật tín hiệu thoại bằng phương pháp tương tự ít được sử dụng trong các ứng dụng cần độ bảo mật cao, do tín hiệu sau khi mã rất rễ khôi phục. Để giải quyết triệt để vấn đề này, cần thực hiện mã hóa dữ liệu thoại số bằng thuật toán trao đổi khóa phiên và mã hóa đủ mạnh *(trong phần kết luận của Chương sẽ đặt ra và hướng giải quyết vấn đề này)*.

1.3. Các phương pháp nén tiếng nói trong mạng GSM [33, 34]

1.3.1. Một số đặc điểm tín hiệu tiếng nói cơ bản của mạng GSM [33].

Phần này chỉ xem xét các ảnh hưởng đến quá trình truyền dữ liệu qua kênh thoại GSM trên khía cạnh băng tần, kỹ thuật xử lý mã thoại, chuyển đổi mã, mà sẽ không xem xét những tác động khác thuộc về truyền thông trong nội tại mạng GSM hay GSM – PSTN.

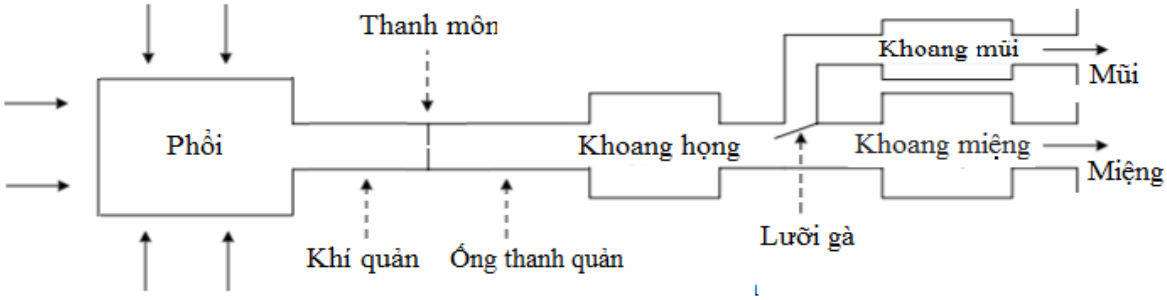
Kênh thoại mạng GSM được thiết kế để truyền tín hiệu tiếng nói với băng tần hẹp 300-3400Hz dẫn đến bị hạn chế tốc độ. Những bộ mã (codec) sử dụng trong GSM khai thác triệt để những thuộc tính của tín hiệu tiếng nói để thu được hiệu suất nén cao, trong khi vẫn giữ lại chất lượng tiếng nói nghe hiểu của người nghe (Điều này dẫn đến tín hiệu không phải tiếng nói bị lọc bỏ bớt bởi các bộ lọc được lập trong mã LPC), trong hầu hết các mô hình nén thoại, tín hiệu được tái tạo sẽ sai khác so với tín hiệu ban đầu. Để đảm bảo âm lượng tiếng nói trong cuộc đàm thoại, mạng GSM sử dụng bộ AGC (Automatic Gain Control) để điều khiển độ lớn biên độ đầu ra. Điều này dẫn đến biên độ của tín hiệu ra có thể khác so với tín hiệu vào; thông thường xen lẫn tín hiệu tiếng nói là những khoảng lặng, bộ phát hiện tiếng nói (VAD - Voice

Activity Detectors) có chức năng phát hiện tín hiệu tiếng nói và loại bỏ những khoảng lặng để tiết kiệm băng thông và năng lượng, vì vậy việc truyền dữ liệu có thể bỏ qua khoảng lặng.

1.3.2. Quá trình tạo và các tính chất cơ bản của tiếng nói

1.3.2.1. Mô hình hoá quá trình tạo tiếng nói [9][9b]

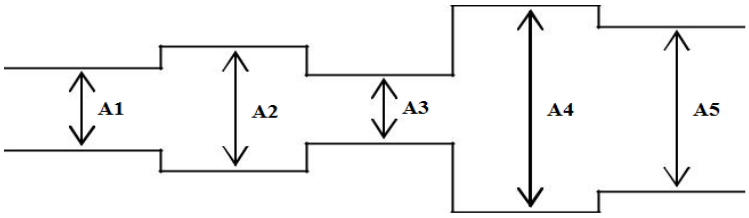
Không khí được ép (kích thích) từ phổi đi qua thanh quản bao gồm các dây thanh âm dao động (theo sự điều khiển của hệ thần kinh) rồi đi dọc theo cơ quan phát âm sẽ tạo ra tiếng nói. Sự dao động của các dây thanh âm tạo ra sự đóng mở tương tự như một cánh cửa (thanh môn). Sự đóng mở này sẽ làm cho luồng không khí từ phổi đi lên bị ngắt quãng khác nhau, làm cho tiếng nói tạo ra cũng khác nhau. Ngoài sự tác động của thanh quản tạo ra các dao động có tần số cơ bản, các thành phần hài bậc cao của tiếng nói phụ thuộc vào sự thay đổi của cơ quan phát âm gồm: họng, vòm họng, lưỡi, miệng, khoang mũi và mũi tương tự như sự thay đổi tham số của các hốc cộng hưởng. **Hình 1.10** biểu diễn mô hình cơ học của hệ thống phát âm của con người.



Hình 1.10. Biểu diễn mô hình cơ học của hệ thống phát âm

Từ mô hình cơ học trên Hình 1.10, có thể biểu diễn cơ quan phát âm bằng mô hình gần đúng gồm các ống hình trụ có độ dài bằng nhau nhưng có đường kính khác nhau (Hình 1.11). Các ống hình trụ này là các hốc cộng hưởng âm thanh với các tần số riêng gọi là tần số formant. Các tần số này tạo ra các âm vị khác nhau tuỳ theo hình dáng cơ quan phát âm. Mô hình này có thể được biểu diễn một cách khá chính xác bằng hệ phương trình vi phân. Trong quá trình phát âm người ta thấy rằng hình dáng cơ quan phát âm thay đổi rất chậm. Do đó trong khoảng thời gian ngắn (trong một

âm vị) có thể xem sự thay đổi là không đáng kể. Vì vậy người ta có thể biểu diễn hệ thống phát âm bằng một hệ thống tuyến tính bất biến theo thời gian. Nghĩa là trong thời gian một âm vị, các tham số của hệ thống sẽ gần như không đổi, chúng chỉ thay đổi từ âm vị này sang âm vị khác.



Hình 1.11. Mô hình dạng ống của cơ quan phát âm

Ngoài ra mô hình hoá quá trình kích thích của luồng không khí từ phổi đi qua thanh quản lên cơ quan phát âm cũng rất quan trọng. Tuỳ theo loại âm thanh mà có cách mô hình hoá thích hợp để tiếng nói sau khi tái tạo đạt được chất lượng theo yêu cầu.

1.3.2.2. Các tính chất cơ bản của tiếng nói

Trong kỹ thuật mã hoá tiếng nói, dựa vào dao động của các dây thanh âm có thể chia tiếng nói ra thành hai loại âm chính sau đây:

Âm hữu thanh (voiced sound): âm hữu thanh được tạo ra khi các dây thanh âm dao động đóng mở làm ngắt quãng luồng không khí và sự ngắt quãng này được xem gần như là tuần hoàn tác động lên cơ quan phát âm. Trong thực tế chu kì tuần hoàn này khoảng từ 2 -20ms. Do đó với âm hữu thanh, tín hiệu kích thích được mô hình hoá là các xung tuần hoàn.

Âm vô thanh (unvoiced sound): âm vô thanh được tạo ra khi luồng không khí đi qua thanh môn tác động lên cơ quan phát âm không theo một qui luật nào cả (không tuần hoàn). Do đó với âm vô thanh, tín hiệu kích thích được mô hình hoá tương tự như tín hiệu ngẫu nhiên (nhiễu).

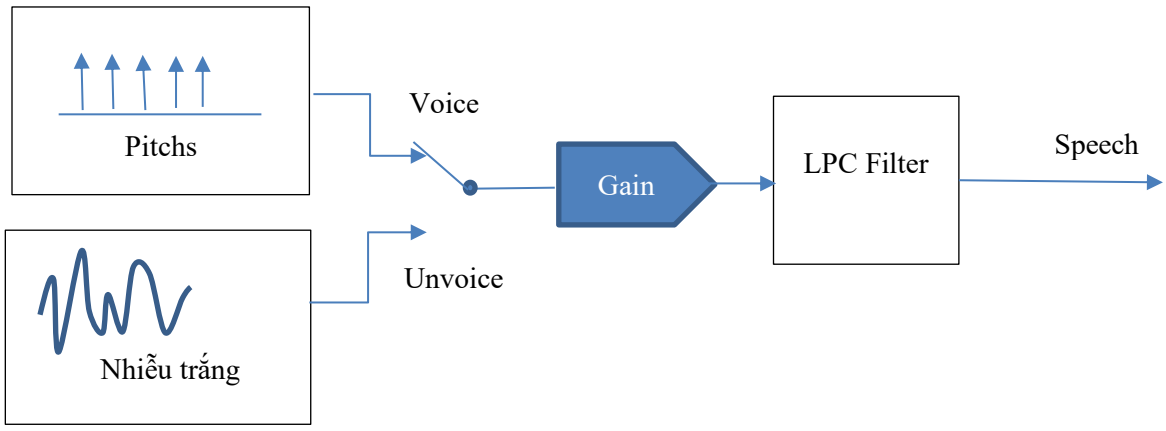
Nhìn chung, các âm của tiếng nói là một trong hai loại âm trên hoặc là sự kết hợp của chúng. Theo thống kê, với tiếng nói của các ngôn ngữ trên thế giới thì phần lớn là các âm là hữu thanh. Một số ngôn ngữ trong đó có tiếng Việt, hầu như chỉ toàn âm

hữu thanh. Thí nghiệm với tiếng Việt cho thấy có thể chỉ dùng hoàn toàn âm hữu thanh mà vẫn không làm ảnh hưởng đến ngữ nghĩa của lời nói.

Để mã hóa và tái tạo tiếng nói, có thể mô hình hóa các tham số thể hiện sự kích không khí từ phổi và giao động qua thanh quản bằng các tham số sau:

- Sự kích thích từ phổi tạo ra thay bằng *nhiễu ngẫu nhiên*.
- Dao động của thanh quản (Khoang họng) được mô hình bằng các bộ lọc tạo chu kỳ '*Pitch*'.
- Khoang tạo âm (Khoang miệng – Khoang mũi) được mô hình bằng bộ lọc LPC.

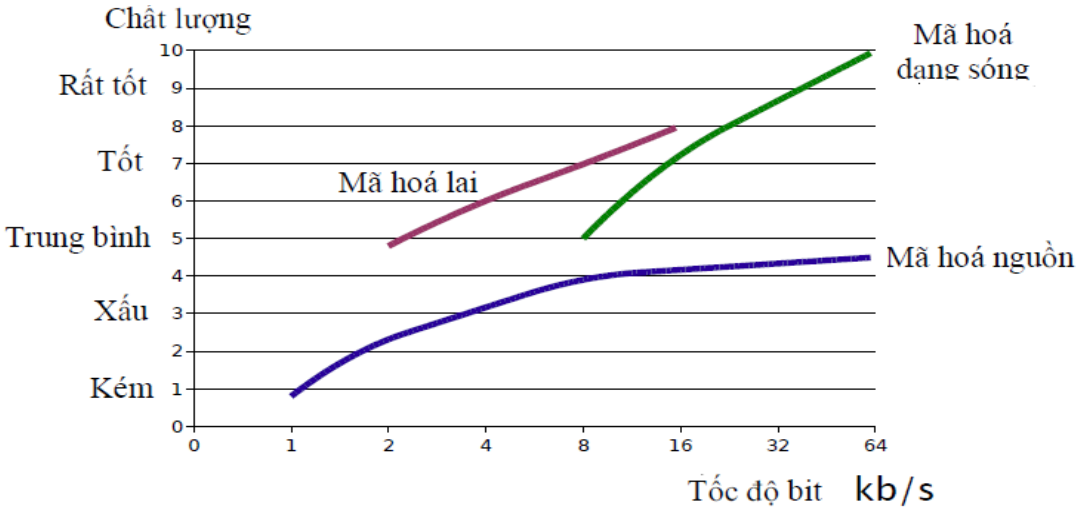
Hình 1.12 dưới đây mô hình hóa này:



Hình 1.12. Mô hình hóa quá trình tạo tiếng nói của con người [9b]

1.3.3. Các phương pháp mã hoá tiếng nói cơ bản

Mã hoá tiếng nói được chia ra thành ba loại chính là mã hoá dạng sóng, mã hoá nguồn và mã hoá lai. Tốc độ bit và chất lượng tiếng nói sau khi tổng hợp lại của các bộ mã hoá này được biểu diễn ở Hình 1.13.



Hình 1.13. Chất lượng tiếng nói với tốc độ bit của các bộ mã hoá

1.3.3.1. Mã hoá dạng sóng

Có thể chia mã hoá dạng sóng ra làm hai loại chính :

Trong miền thời gian: Mã hoá điều biến xung mã (PCM), điều biến xung mã sai lệch (DPCM) và điều biến xung mã sai lệch thích nghi (ADPCM).

Trong miền tần số: Mã hoá băng phụ hay còn gọi là băng con SBC (Subband Coding) và mã hoá biến đổi thích nghi ATC (Adaptive Transform Coding).

1.3.3.2. Mã hoá nguồn

Mã hoá nguồn sử dụng mô hình quá trình tạo ra nguồn tín hiệu và khai thác các thông số của mô hình này để mã hoá tín hiệu. Những thông số của mô hình sẽ được truyền đến bộ giải mã. Đối với tiếng nói, các bộ mã hoá nguồn được gọi là vocoder hoạt động dựa trên mô hình cơ quan phát âm như đã nói ở trên và được kích thích với một nguồn nhiễu trắng đối với các đoạn âm vô thanh hoặc được kích thích bằng một dãy xung có chu kì bằng chu kì pitch đối với đoạn âm hữu thanh. Do đó thông tin được gởi đến bộ giải mã là các thông số kỹ thuật của bộ lọc, một thông tin chỉ định đoạn tiếng nói là âm hữu thanh hay vô thanh, sự thay đổi cần thiết của tín hiệu kích thích và chu kì pitch nếu đó là đoạn tiếng nói hữu thanh.

Có nhiều kỹ thuật để mã hoá nguồn như: mã hoá kênh, mã hoá formant, mã hoá tham số và mã hoá đồng hình. Tuy nhiên, hiện nay chủ yếu tập trung vào nghiên cứu

và phát triển các bộ mã hoá tham số như mã hoá dự đoán tuyến tính kích thích bằng hai trạng thái (mã hoá LPC), mã hoá dự đoán tuyến tính có sự kích thích kết hợp MELP và mã hoá dự đoán tuyến tính kích thích bằng tín hiệu sau dự đoán RELP. Các bộ mã hoá tham số này thường được dùng trong điện thoại vệ tinh và trong an ninh, quốc phòng.

1.3.3.3. Mã hoá lai

Mã hóa lai có nhiều phương pháp nhưng phương pháp phổ biến nhất là mã hoá phân tích bằng cách tổng hợp AbS (Analysis-by-Synthesis). Bộ mã hoá này sử dụng mô hình cơ quan phát âm của người giống như mã hoá nguồn. Tuy nhiên, thay vì sử dụng các mô hình tín hiệu kích thích đơn giản như mã hoá nguồn thì ở đây tín hiệu kích thích được chọn sao cho cố gắng đạt được dạng sóng tiếng nói tái tạo càng giống với dạng sóng tiếng nói ban đầu càng tốt. Đây chính là đặc tính tạo nên sự khác biệt giữa các bộ mã hoá kiểu AbS. Thuật toán tìm ra dạng sóng kích thích này quyết định tới độ phức tạp của bộ mã hoá.

1.3.4. Kỹ thuật nén tiếng nói trong thông tin di động GSM

1.3.4.1. Các bộ mã Codec trong mạng GSM

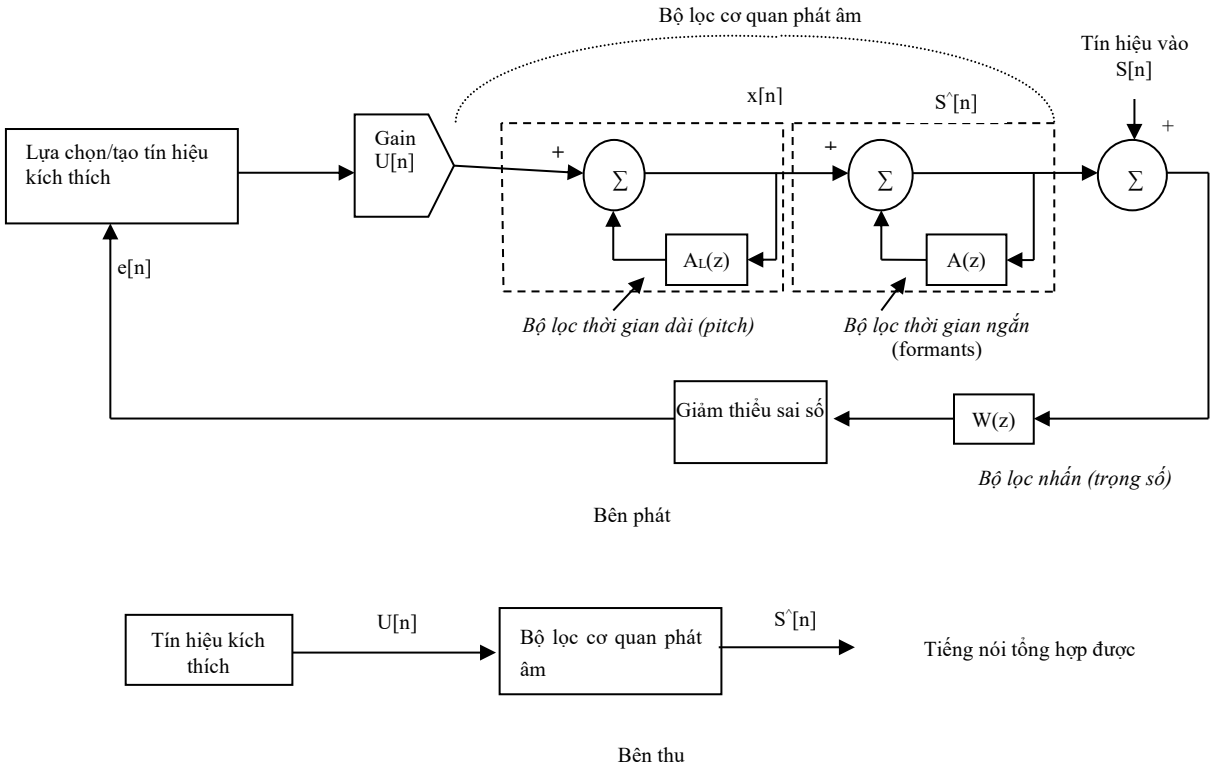
GSM đã sử dụng nhiều loại codec thoại để nén âm thanh 3,1 kHz vào băng thông từ 7-13kbit/s. Ban đầu, hai loại bộ nén thoại, được đặt tên theo các loại kênh dữ liệu được phân bổ, đã được sử dụng, được gọi là Half Rate (6,5 kbit/s) và Full Rate (13 kbit/s). Chúng sử dụng một hệ thống dựa trên mã dự đoán tuyến tính (LPC). GSM đã được cải tiến hơn nữa vào năm 1997 với codec tốc độ đầy đủ (EFR), codec 12,2 kbit/s sử dụng kênh tốc độ đầy đủ. Cuối cùng, với sự phát triển của UMTS, EFR đã được tái cấu trúc thành một codec tốc độ biến đổi được gọi là AMR-Narrowband, có chất lượng cao và mạnh mẽ chống nhiễu khi được sử dụng trên các kênh tốc độ đầy đủ, hoặc kém mạnh mẽ hơn nhưng vẫn có chất lượng tương đối cao trên kênh vô tuyến nửa tốc độ.

1.3.4.2. Cấu trúc một bộ mã hoá tiếng nói dùng phương pháp mã hoá lai AbS [16][8][10][30]

Hầu hết các tiêu chuẩn mã hoá tiếng nói trong thông tin di động GSM đều sử dụng phương pháp mã hoá lai AbS. Vì vậy trong phần này, xin được trình bày chi tiết về mã hoá lai AbS này.

Trong các bộ mã hoá lai, các thông số của hệ thống sẽ được xác định bằng kỹ thuật *dự đoán tuyến tính* như trong mã hoá tham số (ở trong phương pháp mã hoá nguồn) và tín hiệu kích thích được xác định bằng một vòng kín (*phân tích bằng cách tổng hợp*).

Hình 1.14 là sơ đồ khối của một bộ mã hoá lai điển hình [5][30]. Hệ thống này bao gồm một bộ lọc *dự đoán thời gian ngắn* (STP) A(z), một bộ lọc *dự đoán thời gian dài* (LTP) $A_L(z)$, một bộ lọc nhấn W(z), một bộ giảm thiểu sai số cung cấp thông tin cần thiết cho bộ tạo tín hiệu kích thích. Trong đó, quan trọng nhất là bộ tạo tín hiệu kích thích vì nó tạo ra hay chọn tín hiệu kích thích sao cho sai số bình phương trung bình sau khi qua W(z) là nhỏ nhất.



Hình 1.14. Sơ đồ khối của một bộ mã hoá lai [5][30]

Tuỳ theo mỗi loại mã hoá mà bộ tạo tín hiệu kích thích này khác nhau. Mặc dù sơ đồ trên là chung cho các bộ mã hoá lai nhưng một số loại không sử dụng bộ lọc LTP hoặc vị trí STP và LTP thay đổi.

Hàm $A_L(z)$, A(z) là đa thức thu được trực tiếp từ phép biến đổi z của phương trình sai phân tuyến tính hoặc từ phép biến đổi Laplace của phương trình vi phân tuyến tính liên tục chuyển qua gián đoạn với khoảng thời gian T và thay biến s bằng biến z (biến đổi song tuyến tính - bilinear transform) (1.2)

với: $z = e^{sT}$

Sau khi lấy loga cả hai vế, xấp xỉ bằng chuỗi sẽ thu được cặp biểu thức ở dưới. Đây là biến đổi song tuyến tính:

$$s \approx \frac{2}{T}\frac{z-1}{z+1} \text{ và } z \approx \frac{1+sT/2}{1-sT/2} \quad (1.3)$$

Đa thức $A_L(z)$, A(z) có dạng sau *[5,8,30]* :

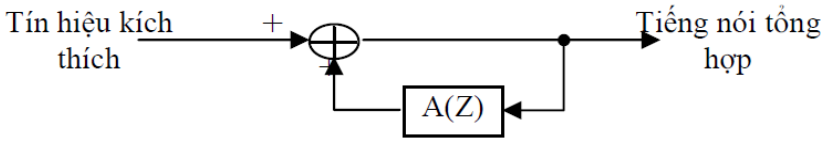
$$\frac{1}{A(z)} = \frac{1}{1-\sum_{i=1}^{p}\alpha_i z^{-i}} \quad (1.4)$$

$$A_L(z) = 1 - \sum_{i=-I}^{I}\alpha_i z^{-T-i} \quad (1.5)$$

Ở đây p, I là bậc của đa thức, α là hệ số bộ lọc LP, β là hệ số khuếch đại, i là chỉ số, T là độ trễ của bộ lọc Pitch, z là các mẫu sau đầu ra bộ lấy mẫu.

a. Dự đoán tuyến tính (LP) dựa trên mô hình phát âm

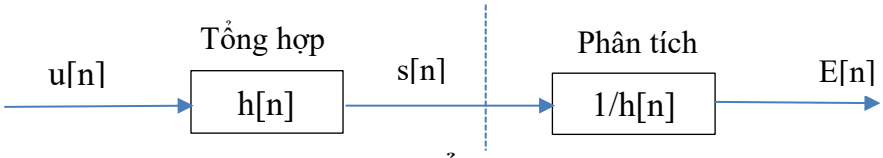
Dự đoán tuyến tính và hệ thống tuyến tính có vai trò rất quan trọng trong xử lí số nói chung và đặc biệt là xử lý tiếng nói. Nó là một công cụ kỹ thuật rất hiệu quả để ước lượng các thông số của tiếng nói như pitch, tần số formant, phổ ... khá chính xác với tốc độ tính toán nhanh.



Hình 1.15. Sơ đồ rút gọn của quá trình tái tạo tiếng nói

Dựa trên hàm truyền đạt người ta có thể biểu diễn mô hình cơ quan phát âm một cách gần đúng như **Hình 1.15**.

Quá trình tổng hợp và phân tích tín hiệu tiếng nói được mô tả trong Hình 1.16 dưới, với giả thiết tín hiệu kích thích u[n] là nhiễu trắng, thì tín hiệu ra e[n] cũng phải là nhiễu trắng nếu h[z] là một hàm truyền đạt toàn cực, không có điểm không.



Hình 1.16. Quá trình tổng hợp và phân tích tín hiệu tiếng nói

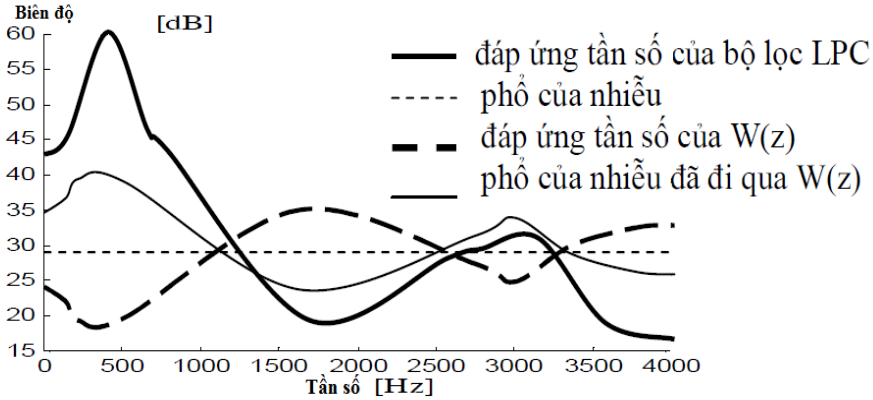
b. Dự đoán thời gian ngắn (STP) và dự đoán thời gian dài (LTP)

Bộ dự đoán thời gian ngắn thực chất là bộ lọc tổng hợp tiếng nói. Bộ lọc này sẽ thực hiện việc tổng hợp tiếng nói khi có tín hiệu kích thích đưa đến đầu vào của nó. Các hệ số của bộ lọc này sẽ được xác định bằng phương pháp dự đoán tuyến tính như đã đề cập ở trên. Các đoạn tiếng nói hữu thanh có dạng sóng tuần hoàn và tính chất tuần hoàn này có thể được khai thác để trợ giúp cho quá trình dự đoán tiếng nói. Từ những điều trên người ta đã đưa ra khái niệm về dự đoán thời gian dài hay dự đoán pitch. Cũng giống như các bộ STP, các bộ LTP cũng là các bộ dự đoán tuyến tính nhưng trong khi STP thực hiện việc dự đoán dựa trên các mẫu kề nhau thì LTP dựa trên các mẫu từ một hay nhiều chu kì pitch trước đó. Đây là lý do gọi nó là dự đoán thời gian dài.

Trong thực tế thay vì truyền đi các hệ số của $A_L(z)$ và A(z) người ta truyền đi các thành phần là LSF hoặc LSP cùng với biên độ (hay năng lượng) của tiếng nói. Bên thu tái tạo lại đa thức $A_L(z)$, A(z) cùng với các thông số khác và tổng hợp lại tiếng nói.

c. Bộ lọc nhấn (lọc trọng số W(z))

Ngoài việc khai thác các tính chất tiếng nói để mã hoá, người ta còn khai thác sự cảm nhận âm thanh của tai người (tai người không cảm nhận được những âm thanh bị che đi bởi các âm thanh khác có năng lượng lớn hơn một mức nhất định – hiệu ứng che lấp) trong mã hoá tiếng nói bằng khái niệm bộ lọc nhấn (bộ lọc tăng cường phổ thích nghi).



Hình 1.17. Biểu diễn hiệu của W(z)

Sự tác động của bộ lọc này được biểu diễn trong Hình 1.17. Ta thấy phổ của nhiễu có hai vùng nằm phía trên của đáp ứng tần số của bộ lọc LPC do đó các tần số nằm trong vùng này sẽ bị nhiễu che đi. Bộ lọc nhấn W(z) sẽ nâng biên độ của nhiễu trong vùng tần số formant (vùng đỉnh của đáp ứng tần số bộ lọc LPC) và nén biên độ của nhiễu trong các vùng trũng của đáp ứng tần số. Phổ của nhiễu sau khi đi qua W(z) sẽ có dạng là đường liền nét mảnh (có hình dạng phổ tương tự như phổ của bộ lọc LPC) và nhiễu sẽ dễ dàng bị các tần số formant che đi (năng lượng các tần số formant che năng lượng nhiễu). Tóm lại, W(z) sẽ định dạng nhiễu hay các sai số sao cho chúng bị che đi bởi các tần số formant năng lượng cao. Bộ lọc nhấn có thể được thực hiện qua hàm sau [12]:

$$W(z) = \frac{a(z)}{a(\frac{z}{\Upsilon})} = \frac{1+\sum_{i=1}^{p} a_i z^{-i}}{1+\sum_{i=1}^{p} a_i \Upsilon^i z^{-i}} \quad (1.6)$$

Với Υ được chọn như một hằng số giữa 0 và 1 (thường là 0.8 hoặc 0.9).

1.3.4.3. Một số loại mã hoá lai dùng trong liên lạc di động

Tuỳ theo cách tạo ra tín hiệu kích thích mà người ta đưa ra các loại mã hoá lai khác nhau như mã hoá đa xung MPE, mã hoá xung đều RPE, mã hoá kích thích bằng mã CELP, mã hoá kích thích vectơ tổng VSELP. Trong các tiêu chuẩn dùng cho liên lạc di động có ba loại sau đây thường được sử dụng là: RPE-LTP, ACELP và VSELP. Giới hạn tại Việt Nam nên chỉ quan tâm đến mã CELP và ACELP.

1.4. Kết luận chương 1

Với các số liệu và phân tích ở phần đầu Chương 1, với trình độ công nghệ và năng lực tính toán như hiện nay, rõ ràng với các thuật toán xác thực và bảo mật dữ liệu trong mạng di động GSM hiện nay thì vấn đề không an toàn và không bảo mật được thông tin cho người dùng có thông tin nhạy cảm chứ chưa nói đến thông tin bí mật quốc gia. Với cấu trúc và các phần tử mạng trong hệ thống mạng GSM hiện tại thì vấn đề xác thực và bảo mật thông tin thoại từ người dùng đến người dùng (End to End) là tối ưu nhất.

Với quá trình tạo và tổng hợp tiếng nói theo mô hình hóa trên (Hình 1.11), có thể biểu diễn hệ thống phát âm bằng một hệ thống tuyến tính bất biến theo thời gian, khi đó các bộ mã hoá thoại trong miền thời gian được xây dựng dựa trên kĩ thuật dự đoán tuyến tính. Do vậy các đặc tính thống kê của tín hiệu thoại được mô hình hoá một cách chính xác thông qua mô hình bộ lọc nguồn (*bộ lọc dự đoán thời gian ngắn, bộ lọc dự đoán thời gian dài*) với giả thiết tiếng nói là kết quả của quá trình kích thích một bộ lọc biến đổi theo thời gian bằng một dãy xung có chu kì đối với âm rung voiced và nguồn nhiễu ngẫu nghiên đối với âm câm.

Các bộ mã hoá sử dụng kĩ thuật phân tích trong miền thời gian có thể được kể đến như bộ mã hoá dự đoán thích nghi APC (Adaptive Predictive Coding), Mã dự đoán tuyến tính kích thích dư RELP (Residual Excited Linear Prediction), hay Mã dự đoán tuyến tính đa xung MPLPC (Multi Pulse LPC), và *đáng chú ý nhất là bộ Mã hoá dự đoán tuyến tính mã kích thích CELP* (Code-Excited LPC) *phù hợp với các ứng dụng dữ liệu tín hiệu thoại mã hóa qua kênh thoại analog đi qua các mạng khác nhau*.

CHƯƠNG 2: ĐỀ XUẤT THUẬT TOÁN NÉN VÀ ĐỀ XUẤT GIẢI PHÁP BẢO MẬT, TRUYỀN DỮ LIỆU QUA KÊNH THOẠI GSM

2.1. Lựa chọn giải pháp mã hóa mật cuộc gọi thoại di động trên kênh GSM

Thiết bị liên lạc GSM đã thực hiện các công việc: Vocoder chuyển tín hiệu tiếng nói sang dạng số hóa đã nén theo chuẩn GSM, sau đó gói tin số hóa được truyền qua kênh GSM ở dạng tín hiệu số (cùng với các tín hiệu điều khiển GSM khác). Dữ liệu GSM đã được mã hóa bảo mật với thuật toán mã A5/x, song như đã phân tích ở Chương 1 cho thấy rằng họ thuật toán này chưa đủ độ an toàn cần thiết.

Để mã hóa cuộc gọi thoại trên kênh voice GSM, có một giải pháp đơn giản là xáo trộn tần, đảo phổ tín hiệu tiếng nói theo một quy tắc xác định bởi khóa mã. Đây còn gọi là phương pháp mã hóa ở mức tương tự (trước khi tín hiệu được số hóa). Phương pháp này dễ áp dụng, song luôn có giải pháp phân tích tín hiệu đã xáo trộn để khôi phục lại tín hiệu ban đầu.

Nếu có thể can thiệp vào quá trình xử lý dữ liệu của Modem GSM, ta có thể thực hiện mã hóa dữ liệu dạng số hóa trước khi truyền trên kênh. Tuy nhiên tất cả các Modem GSM đều có tính đóng kín, không hỗ trợ can thiệp vào quá trình xử lý dữ liệu nội bộ của Modem.

Có thể sử dụng một giải pháp trung gian, đó là sử dụng chế độ truyền dữ liệu trên băng tần GSM (kênh CSD). Đây là một chuẩn truyền số liệu có sẵn trên kênh GSM được sử dụng để truyền tín hiệu Fax. Tuy nhiên việc hỗ trợ CSD tại Việt nam hiện nay còn hạn chế. Ta cũng không bàn tới việc truyền dữ liệu qua kênh IP (GPRS hoặc 3G/4G)

Đề xuất phương án bảo đảm tốt nhất để mã hóa và truyền dữ liệu cuộc gọi thoại mật qua kênh GSM là *xây dựng module thực hiện các công đoạn: tự thực hiện Vocoder với bitrate thấp; mã hóa dữ liệu thoại thu được sử dụng một thuật toán mã đủ mạnh, có thể sử dụng mã hóa khóa đối đối xứng; điều chế dữ liệu mã thành tín hiệu trong phổ tiếng nói, đưa tín hiệu đã điều chế này (dạng tương tự) vào đầu vào của thiết bị đầu cuối (ME) thuộc hệ thống GSM truyền qua kênh GSM thông thường*,

việc này như là phát triển một Modem làm việc trên kênh thoại 2G/3G, *nếu làm được Modem có tính năng này, thì Modem này không chỉ cho phép truyền dữ liệu qua kênh Voice GSM 2G/3G, mà nó còn có thể truyền dữ liệu qua tất cả các giao thức, các mạng cho phép truyền thông tin thoại như các mạng điện thoại chuyển mạch gói, mạng vô tuyến công nghệ SDR, OTT,..* Ở bên máy thu, ta sẽ thực hiện các bước theo thứ tự ngược lại để thu được tín hiệu tiếng nói ban đầu. Việc phát triển một Modem như vậy, nó liên quan đến một loạt các kỹ thuật, cụ thể sẽ được đề cập dưới đây.

Có hai vấn đề cần quan tâm khi thực hiện phương án này: cần xử lý điều chế ở mức thời gian thực; sử dụng một giải pháp Vocoder có Bitrate đủ thấp để có thể điều chế lại thành tín hiệu trong phổ tần và giống tín hiệu tiếng nói (vì kênh voice GSM được nén xuống rất thấp (hệ số nén cao) và không được phá vỡ cấu trúc khung thoại qua giao diện Um với 20ms/slot, với chuẩn nén 13kbps sẽ tương ứng 260 bits/slot, cần phải quan tâm đến cấu trúc khung này để tạo ra một dạng sóng sẽ truyền trên khung này mà không bị biến dạng đáng kể). Cần chú ý là tín hiệu này lại thông qua tầng Vocoder của GSM một lần nữa. Cũng do tầng Vocoder của GSM làm nảy sinh vấn đề thứ hai, đó là tầng Vocoder của GSM có sử dụng bộ phát hiện tiếng nói tích cực VAD để phát hiện các khoảng không có âm thanh. Với tín hiệu điều chế trong phổ tiếng nói, có nhiều phần không có đặc trưng giống hệt tiếng nói thông thường, vì thế được VAD xác định là không phải tiếng nói. Dữ liệu điều chế trong khoảng thời gian VAD xác định là không phải tiếng nói sẽ không được truyền sang máy thu đầu xa để tiết kiệm công suất phát. Ta cần chỉnh sửa bộ điều chế để tránh việc bị VAD xác định là không phải tiếng nói mà vẫn bảo đảm truyền dữ liệu đủ hiệu quả

Việc lựa chọn bộ nén thoại (*Vocoder*) lý tưởng cần đáp ứng các tiêu chí: có Bitrate thấp, chất lượng thoại tốt, có khả năng chống nhiễu (chống lỗi kênh) tốt, thích ứng với ngôn ngữ người dùng, hiệu suất cao với tín hiệu câm (*unvoiced*), độ trễ ngắn (*real time*), phù hợp với các tài nguyên xử lý hạn chế.

2.2. So sánh ba thuật toán nén dùng kỹ thuật dự đoán tuyến tính (LP Specch Model)

Ba thuật toán nén sử dụng LP là: LPC10e, CELP và MELP. Cả 3 thuật toán này đề là dùng kỹ thuật dự đoán tuyến tính và đề được dung phổ biến đặc biệt là trong thông tin vô tuyến, sự khác nhau cơ bản của ba thuật toán này là nằm ở sự phức tạp của chế độ kích thích. Mô hình kích thích được sử dụng bởi vocoder CELP là tinh vi nhất trong ba mô hình, nhưng mô hình kích thích trong vocoder MELP là đơn giản hóa và đã được chứng minh [12]. Thuật toán nén CELP và MELP đạt chất lượng tương đương, nhưng MELP đạt được tỷ lệ nén rất cao (bằng ½ CELP) và có nhiều chế độ, nhiều phương pháp cải thiện kích thích (pitch) và nhiều tốc độ rất thấp, các **Bảng 2.1, 2.2** dưới đây so sánh việc cấp phát bit giữa vocoder CELP và MELP:

Các tham số	***Số bit được cấp phát***
Vocing	4
Energy	11
Pitch	7
Spectrum	38

Bảng 2.1. Số bit được cấp phát cho MELP 600bit [12]

Tham số	***Tổng số bit mỗi Frame***
LPC	34
Chu kỳ Pitch (chỉ số bảng mã thích nghi)	28
Độ khuyết đại bảng mã thích nghi	20
Chỉ số phức tạp bảng mã Codebook	36
Độ phức tạp bảng mã khuyết đại	20
Đồng bộ	1
Mã sửa lỗi	4
Dự phòng mở rộng	1

Bảng 2.2. Cấp phát các bit cho CELP (FS1061)

2.3. Mô hình và đề xuất bộ mã hoá dự đoán tuyến tính kích thích hỗn hợp MELP

2.3.1. Đặt vấn đề

Như mục 2.1 đã đặt ra, việc lựa chọn bộ nén thoại (Vocoder) trong mục tiêu của luận án này là hết sức quan trọng, nó phải đáp ứng các tiêu trí như trình bày ở trên. Có rất nhiều phương pháp nén thoại, nhưng tại sao lại chọn thuật toán MELP. Không phải ngẫu nhiên mà Mỹ lại đưa MELP (các phiên bản từ MELP đến MELPe) thành chuẩn US MIL-STD-3005 và các nước NATO là STANAG-4591 (MELPe), vì các Model thuật toán nén thoại MELP (MELP, MELPe, MELPe++) bao gồm các bít đồng bộ hóa vô cùng hiệu quả trong các ứng dụng bảo mật vô tuyến băng tần hẹp (Narrowband) trên các kênh hay lỗi bít, mất gói, mất đồng bộ và nó rất hiệu quả trong việc cân đối giữa băng thông và chất lượng tín hiệu thoại, đặc biệt độ phức tạp tính toán có thể thực hiện trên các Chíp ARM hay DSP, phù hợp cho bài toán bảo mật thông tin thoại.

Ngày nay, chuẩn mã thoại MELPe gồm 3 tốc độ 2400bps, 1200bps và 600bps [11b,13,14].

Các vocoder dựa trên phương pháp mã hoá mô hình, các tham số đặc trưng cần được xác định đối với mỗi khung tín hiệu là:

- ***Các tham số kích thích*** gồm quyết định voice/unvoice và chu kì pitch
- ***Các tham số hệ thống*** gồm thông tin đường bao phổ hay đáp ứng xung của hệ thống.

Bên giải mã, để tổng hợp lại tiếng nói, các tham số kích thích được sử dụng để tổng hợp thành tín hiệu kích thích, tín hiệu này gồm một chuỗi xung tuần hoàn trong vùng voice và nhiễu ngẫu nhiên trong vùng unvoice. Tiếp đó, tín hiệu kích thích này sẽ được cho qua một bộ lọc mà các tham số của nó dựa trên các tham số hệ thống đã ước lượng. Mô hình hay được sử dụng cho hệ thống là LPC (*Linear Prediction Coding*), trong đó mã hoá dự đoán tuyến tính được sử dụng để mô hình hoá đường bao phổ.

Cải tiến quan trọng trong MELP về mô hình là sử dụng mô hình kích thích hỗn hợp tuần hoàn/nhiễu và quyết định voice/unvoice trong từng dải tần riêng làm tăng rất đáng kể chất lượng tiếng nói so với LPC-10 tuy nhiên nó vẫn có một số khuyết điểm về chất lượng tiếng nói, đặc biệt là ở các vùng không ổn định của tiếng nói và các vùng không tuần hoàn.

Các vocoder truyền thống cho tiếng nói tổng hợp về phương diện nghe hiểu là khá tốt nhưng chúng lại không có chất lượng tiếng cao. Chúng ta có thể phân các lý do theo hai loại sau:

- Do hạn chế cơ bản trong mô hình tiếng nói
- Do ước lượng các tham số mô hình tiếng nói không chính xác.

Các nhược điểm của các vocoder truyền thống về suy giảm chất lượng có thể thấy và các cơ sở của các biện pháp để giải quyết như sau:

Một trong những suy giảm chất lượng chủ yếu trong các vocoder sử dụng mô hình voice/unvoice đơn giản là : trong một số vùng của tiếng nói, nghe được hỗn độn, các âm ù, đặc biệt trong các vùng voice hỗn hợp hay các vùng voice có nhiễu. Quan sát phổ thời gian ngắn của các đoạn tín hiệu này cho thấy nổi trội chủ yếu ở các vùng voice hỗn hợp là các hài của tần số cơ bản, ở các vùng voice có nhiễu là năng lượng giống nhiễu. Nguyên nhân quan trọng của hiện tượng này là do mô hình kích thích voice/unvoice đơn giản chỉ đơn thuần tạo ra các tín hiệu kích thích có phổ hoặc là gồm các hài của tần số cơ bản hoặc là phổ nhiễu.

Để giải quyết vấn đề này cần sử dụng mô hình kích thích hỗn hợp, các kích thích tuần hoàn và giống nhiễu sẽ được trộn lại, sau đó được tạo dạng phổ bởi một bộ lọc, các tham số của bộ lọc này dựa trên các tham số hệ thống trong quá trình phân tích.

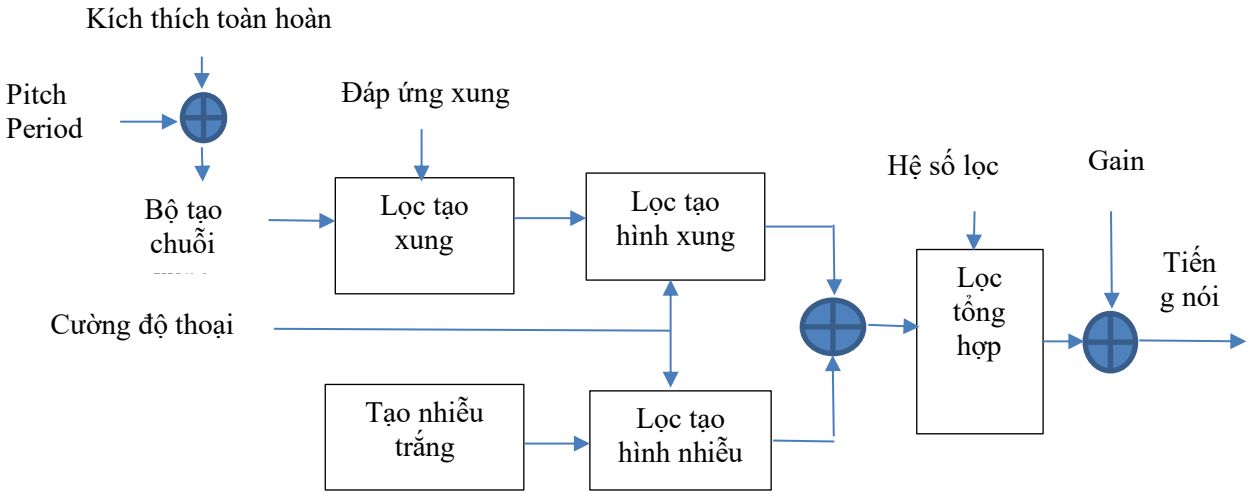
Cơ sở của cách giải quyết này là do con người có thể phân biệt giữa các vùng tần số đặc trưng bởi hài của tần số cơ bản và các vùng tần số khác đặc trưng bởi nhiễu nên để nâng cao chất lượng tiếng nói tổng hợp, mô hình voice/unvoice phải có thêm yếu tố biến thiên theo tần số. Hình vẽ (Hình 1.14) thể hiện khái niệm mô hình kích thích hỗn hợp đa băng nói trên.

Bên cạnh yếu tố mô hình, chất lượng xấu của tiếng nói tổng hợp của các vocoder cũng có một phần lớn là do sự ước lượng không chính xác các tham số mô hình tiếng nói [35]. Ví dụ như ước lượng pitch hay quyết định voice/unvoice không chính xác (do việc xác định pitch cùng với quyết định voice/unvoice không chính xác) thường làm cho tiếng nói tổng hợp suy giảm chất lượng rất nhiều, đặc biệt là trong tín hiệu tiếng nói có nhiễu, sự suy giảm này xuống đến mức nghiêm trọng.

Tóm lại, một hệ thống vocoder chất lượng cao cần phải cải tiến mô hình tiếng nói và có các phương pháp tin cậy ước lượng chính xác các tham số của mô hình tiếng nói.

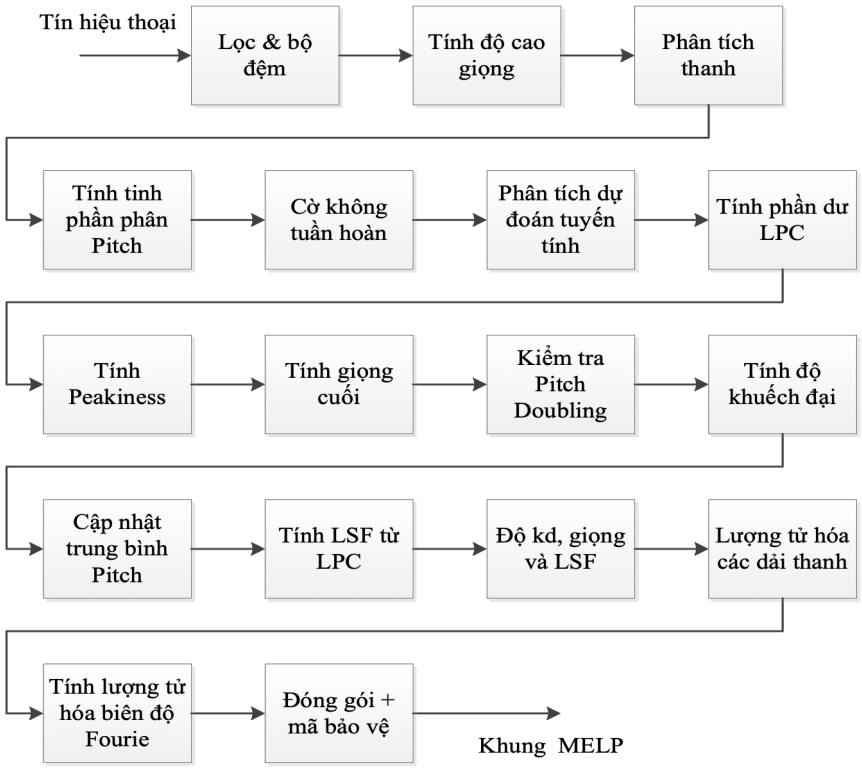
2.3.2. Mô hình thuật toán mã thoại MELP

Sơ đồ khối bộ mã hoá tiếng nói MELP (*Mixed Excitation Linear Prediction*) được thể hiện trong ***hình 2.1*** [*17*][*8*] dưới đây:



Hình 2.1. Mô hình mã hóa tiếng nói Melp

2.3.2.1. Quá trình mã thoại MELP được biểu diễn trên Hình 2.2 [13][18]:



Hình 2.2. Quy trình thực hiện mã thoại Melp.

Một số bước cơ bản trong qui trình này được thực hiện như sau:

Bước 1: Loại bỏ thành phần tần số thấp (DC)

Bước đầu tiền của quá trình nén tiếng nói là loại bỏ thành phần năng lượng tần số thấp trong tín hiệu đầu vào. Để thực hiện bước này, người ta sử dụng một bộ lọc thông cao Chebychev kiểu 2 bậc bốn có tần số cắt 60 Hz và suy giảm 30 dB ở dải chắn. Đầu ra của bộ lọc này được sử dụng làm tín hiệu đầu vào cho tất cả các khối tiếp theo trong quá trình nén tiếng nói.

Bước 2: Tìm kiếm pitch tổng thể

Để tìm kiếm pitch tổng thể, đầu tiên tín hiệu đầu vào được lọc thông thấp với bộ lọc Butterword bậc 6 với tần số cắt 1 KHz. Giá trị pitch được tìm trong khoảng $\tau = 40 \div 160$ mẫu và là giá trị có tự tương quan chuẩn hoá (τr) lớn nhất, trong đó:

$$r(\tau) = \frac{c_\tau(0,\tau)}{\sqrt{c_\tau(0,0)c_\tau(\tau,\tau)}} \quad (2.1)$$

Ở đây:

$$c_\tau(m,n) = \sum_{k=-\lfloor \tau/2 \rfloor-80}^{-\lfloor \tau/2 \rfloor+79} s_k + m^S k + n \quad (2.2)$$

Và τ là cờ cho số nguyên của các mẫu, [τ/2] là giá trị phần nguyên của τ/2, như thể hiện trong công thức (2.2), cửa sổ phân tích pitch được đặt tại mẫu cuối cùng của khung hiện tại. Việc tính toán pitch cuối cùng sẽ được mở rộng đến 20 mẫu (qua việc nội suy và kiểm tra bội pitch).

Bước 3: Phân tích voice tại các dải tần [12]

Khối chức năng này xác định năm cường độ voice ở 5 dải thông Vbp_i, i = 1÷5. Nó cũng xác định chính xác hơn pitch và giá trị hàm tự tương quan chuẩn hoá đã tìm được.

Tín hiệu tiếng nói đầu vào sẽ được lọc ở năm dải tần: 0÷500 Hz, 500÷1000 Hz, 1000÷2000 Hz, 2000÷3000 Hz, 3000÷4000 Hz bằng các bộ lọc Butterword bậc 6. Việc xác định pitch chính xác thực hiện trên tín hiệu đầu ra của bộ lọc 0÷500 Hz. Có hai giá trị pitch được xem xét là pitch tổng thể của khung hiện tại và khung trước đó, đối với mỗi giá trị pitch được xét, hàm tự tương quan chuẩn hoá (phương trình 3.1) được tính trên khoảng ± 5 của giá trị pitch đó, tiếp theo pitch phân sẽ được tìm tiếp xung quanh giá trị nguyên tối ưu tìm được. Nếu pitch phân nào có giá trị hàm tự tương quan chuẩn hoá lớn hơn sẽ được chọn là pitch mới P_2. Giá trị hàm tự tương quan chuẩn hoá $r(P_2)$ là cường độ voice dải tần 0÷500 Hz. Giá trị P_2 sẽ được dùng để xác định cường độ voice ở các dải tần khác, xác định pitch cuối cùng và tính toán độ khuếch đại.

Đối với các dải tần còn lại, cường độ voice trong dải đó là giá trị lớn hơn trong hai giá trị sau:

+ r(P_2) trong thủ tục tìm pitch phân trong dải tần này

+ r(P_2) trong thủ tục tìm pitch phân của đường bao miền thời gian của tín hiệu trong dải này.

Giá trị này sẽ được giảm đi 0.1 để bù lại sai số do việc làm trơn tín hiệu đường bao miền thời gian. Việc tính toán đường bao này do một khối tách toàn dạng sóng và một bộ lọc làm trơn. Bộ lọc này gồm một điểm không một chiều và một cặp điểm cực phức tại tần số 150 Hz, bán kính 0.97.

Bước 4: Xác định chính xác pitch phân

Thủ tục này dựa trên cơ sở nội suy để tăng độ chính xác của giá trị pitch đầu vào. Nếu giá trị đầu vào là pitch phân thì sẽ được làm tròn tới số nguyên gần nhất. Giả sử giá trị pitch nguyên là T mẫu, công thức nội suy giả thiết rằng hàm r(τ) đạt giá trị lớn nhất trong đoạn [T, T+1]. Trong thủ tục này $c_T(0, T-1)$ và $c_T(0, T+1)$ được so sánh để xác định giá trị r(τ) lớn nhất đó nằm trong đoạn [T-1, T] hay [T, T+1], lưu ý rằng khi $c_T(0, T-1) > c_T(0, T+1)$ (tức là pitch nằm trong đoạn [T-1, T]) thì pitch sẽ được trừ đi 1 để đúng với giả thiết của công thức nội suy.

Độ lệch phân Δ được tính bởi công thức nội suy sau:

$$\Delta = \frac{c_T(0,T+1)c_T(T,T) - c_T(0,T)c_T(T,T+1)}{c_T(0,T+1)[c_T(T,T+1)] + c_T(0,T)[c_T(T+1,T+1) - c_T(T,T+1)]} \qquad (2.3)$$

trong đó $c_T(m, n)$ được định nghĩa ở công thức (2.2).

Trong một số trường hợp, Δ vượt ngoài dải 0.0÷1.0, vì vậy nó được giới hạn trong dải -1÷2.Giá trị pitch phân là T + Δ và nằm trong dải 20÷160.

Giá trị tự tương quan chuẩn hoá tại giá trị pitch phân tìm được bằng:

$$r(T+\Delta) = \frac{(1-\Delta)c_T(0,T) + \Delta c_T(0,T+1)}{\sqrt{c_T(0,0)[(1-\Delta)^2 c_T(T,T) + 2\Delta(1-\Delta)c_T(T,T+1) + \Delta^2 c_T(T+1,T+1)]}} \qquad (2.4)$$

Các phương trình trên dựa trên việc nội suy tuyến tính các mẫu tín hiệu đầu vào.

Bước 5: Cờ không tuần hoàn

Cờ không tuần hoàn được đặt bằng 1 nếu $Vbp_1 < 0.5$ và bằng 0 nếu ngược lại. Khi được đặt bằng 1, cờ này thông báo cho khối giải mã rằng thành phần xung kích thích mang tính không tuần hoàn nhiều hơn là tuần hoàn.

Bước 6: Phân tích dự đoán tuyến tính.

Tín hiệu tiếng nói đầu vào được lấy bởi một cửa sổ Hamming 200 mẫu (25ms) sẽ được bộ phân tích dự đoán tuyến tính bậc 10 có tâm đặt tại mẫu cuối cùng của khung hiện tại. Bước đầu tiên là tìm các hệ số tự tương quan bậc 0÷10, sau đó dựa trên thuật toán Levison – Durbin để tìm các hệ số dự đoán tuyến tính a_i, i = 1÷10. Cuối cùng, các hệ số dự đoán tuyến tính được mở rộng dải tần 15 Hz với hệ số 0.934 ($a_i = a_i \times 0.934^i$)

Bước 7: Tính toán thặng dư dự đoán tuyến tính

Để có tín hiệu thặng dư dự đoán tuyến tính, chúng ta cho tín hiệu tiếng nói đầu vào qua bộ lọc dự đoán tuyến tính (với các hệ số được xác định ở phần trên). Cửa sổ thặng dư có tâm đặt tại mẫu cuối cùng của khung hiện tại và có độ dài đủ lớn để sử dụng trong tìm pitch cuối cùng.

Bước 8: Tính toán độ nhọn của tín hiệu thặng dư

Độ nhọn của tín hiệu thặng dư (*Peakiness)* được tính toán trên cửa sổ 160 mẫu có tâm đặt tại mẫu cuỗi cùng của khung hiện tại. Độ nhọn của tín hiệu là tỉ số của hàm chuẩn bậc hai trên hàm chuẩn bậc nhất:

$$Peakiness = \frac{\sqrt{\frac{1}{160}\sum_{n=1}^{160} r_n^2}}{\frac{1}{160}\sum_{n=1}^{160}|r_n|} \qquad (2.5)$$

Nếu độ nhọn lớn hơn 1.34 thì cường độ voice dải tần thấp nhất được đặt bằng 1 và nếu lớn hơn 1.6 thì cường độ voice ba dải tần thấp nhất sẽ được đặt bằng 1.

Bước 9: Tính toán pitch cuối cùng

Giá trị pitch cuối cùng được tính trên tín hiệu thặng dư sau khi đã được lọc thông thấp bằng bộ lọc Butterword bậc 6 có tần số cắt 1 KHz.

Đầu tiên, P_2 được làm tròn đến số nguyên gần nhất và giá trị pitch nguyên sẽ được chọn trong lân cận ± 5 để cực đại hoá hàm tương quan chuẩn hoá (công thức 2.1). Tiếp theo là bước làm tinh pitch phân xung quanh giá trị nguyên tối ưu tìm được, kết quả là ta được giá trị pitch phân P_3 và tự tương quan chuẩn hoá tương ứng là $r(P_3)$. Cuối cùng, chúng ta sẽ tìm pitch cuối cùng bằng thủ tục kiểm tra bội pitch, thuật toán như sau:

```
inputs: the input speech signal; the residual signal; P2; Pavg
outputs: P3, cor_P3
fresid buffer = filter the residual with a 1 kHz Butterworth
P3 = best integer pitch on fresid over the range P2-5 to P2+5
P3, cor_P3 = frac_pitch(fresid, P3)
if (cor_P3 >= 0.6)
Dth = 0.5
if (P3 <= 100) Dth = 0.75
P3, cor_P3 = double_ck(fresid, P3, Dth)
else
P3, cor_P3 = frac_pitch(input, P2)
if (cor_P3 < 0.55)
P3 = Pavg
else
Dth = 0.7
if (P3 <= 100) Dth = 0.9
P3, cor_P3 = double_ck(input, P3, Dth)
endif
```

```
endif
if (cor_P3 < 0.55) P3 = Pavg
```

Bước 10: Kiểm tra bội pitch

Thủ tục kiểm tra bội pitch tìm kiếm và sửa lại giá trị pitch là bội của pitch thực sự. ý tưởng của thuật toán kiểm tra này là xét các pitch là ước số của pitch đầu vào và tính tự tương quan tương ứng với chúng, sau đó so sánh các tự tương quan này với nhau, giá trị pitch có tự tương quan tương ứng lớn nhất là pitch cần tìm. Tự tương quan ứng với pitch nhỏ sẽ được ưu tiên so với tự tương quan ứng với pitch lớn hơn bởi một hệ số D_{th}. Đoạn mã giả sau thể hiện thuật toán này:

```
inputs: signal; P; Dth
outputs: Pc, cor_Pc
Pc, cor_Pc = frac_pitch(signal, P)
for (k=8; k>=2; k--)
Pk = Pc/k
if (Pk >= 20)
Pk, cor_Pk = frac_pitch(signal, Pk)
if (Pk < 30) cor_Pk = double_ver(Pk, cor_Pk)
if (cor_Pk > Dth * cor_Pc)
Pc, cor_Pc = frac_pitch(signal, Pk)
break
endif
endif
endfor
if (Pc < 30) cor_Pc = double_ver(Pc, cor_Pc)
```

Với đầu vào P và r(P), thủ tục xác minh bội pitch trả về giá trị nhỏ hơn giữa r(P) và r(2P), trong đó r(2P) được xác định bằng thủ tục tìm pitch phân xung quanh 2P. Việc xác minh bội trong thủ tục kiểm tra bội pitch có tác dụng loại trừ các pitch giả có giá trị nhỏ.

Bước 11: Tính toán độ khuếch đại

Độ khuếch đại tín hiệu tiếng nói đầu vào được đo hai lần trong một khung. Độ dài cửa sổ tính toán cho hai độ khuếch đại là bằng nhau và thay đổi theo pitch.

Khi Vbp_1 lớn hơn 0.6, độ dài cửa sổ là giá trị nhỏ nhất trong các bội của P_2 lớn hơn 120 mẫu. Nếu độ dài này lớn hơn 320, nó được chia cho 2.

Khi Vbp_1 nhỏ hơn hoặc bằng 0.6, độ dài cửa sổ là 120 mẫu.

G_1 được tính toán trên cửa sổ có tâm ở trước mẫu cuối cùng của khung hiện tại 90 mẫu và cửa sổ để tính toán G_2 có tâm tại mẫu cuối cùng của khung hiện tại. Các độ khuếch đại là các giá trị RMS, được tính theo dB:

$$G_i = 10\log_{10}(0.01 + \frac{1}{L}\sum_{n=1}^{L} S_n^2 \qquad (2.6)$$

Tín hiệu đầu vào nằm trong dải -32768÷32767, L là độ dài cửa sổ đã đề cập ở trên.

Thành phần 0.01 tránh để tham số của hàm log quá gần 0. Nếu G_i nhỏ hơn 0 thì G_i bằng 0.

Bước 12: Cập nhật pitch trung bình

Pitch trung bình thời gian dài P_{avg} được cập nhật bằng một thủ tục làm trơn đơn giản sau:

Nếu $r(P_3)$ lớn hơn 0.8 và G_2 lớn hơn 30dB thì P_3 được đặt vào một bộ đệm gồm ba giá trị pitch mạnh gần nhất p_i, i = 1÷3. Nếu không, cả ba giá trị pitch trong bộ đệm được thay đổi hướng tới giá trị pitch ngầm định, $P_{default}$ = 50 mẫu theo công thức sau:

$$p_i = 0.95\, p_i + 0.05\, P_{default},\ i = 1 \div 3 \qquad (2.7)$$

Giá trị pitch trung bình sau đó được chọn là giá trị ở giữa của ba giá trị trong bộ đệm. P_{avg} được sử dụng trong tính toán pitch cuối cùng.

Bước 13: Lượng tử hoá các hệ số dự đoán

Đầu tiên, các hệ số dự đoán tuyến tính a_i, i = 1÷10 được chuyển đổi thành các tần số phổ đường (Line spectrum frequency-LSF). Tiếp theo, các thành phần LSF được sắp xếp theo thứ tự tăng dần với khoảng cách nhỏ nhất là 50 Hz. Tiêu chuẩn khoảng cách nhỏ nhất này được thực hiện bằng cách sửa từng cặp tần số không thoả mãn. Đoạn mã giả sau thể hiện thuật toán này:

```
dmin = 50
for (i=1; i<10; i++)
d = f[i+1] - f[i]
if (d < dmin)
s1 = s2 = (dmin-d)/2
if (i == 1 and f[i] < dmin) s1 = f[i]/2
else if (i > 1)
tmp = f[i] - f[i-1]
if (tmp < dmin) s1 = 0
else if (tmp < 2*dmin) s1 = (tmp-dmin)/2
endif
if (i == 9 and f[i+1] > 4000-dmin) s2 = (4000-f[i+1])/2
else if (i < 9)
tmp = f[i+2] - f[i+1]
if (tmp < dmin) s2 = 0
else if (tmp < 2*dmin) s2 = (tmp-dmin)/2
endif
f[i] = f[i] - s1
f[i+1] = f[i+1] + s2
endif
endfor
```

Vector LSF nhận được f sẽ được lượng tử hoá sử dụng bộ lượng tử vector nhiều tầng (Multi-stage vector quantizer – MSVQ). Sách mã MSVQ gồm 4 tầng với số mức tương ứng là 128, 64, 64 và 64.

Vector được lượng tử hoá $\hat{f}$ là tổng của các vector được lựa chọn ở mỗi tầng. Vector tìm được có bình phương khoảng cách Euclidean được lấy trọng số giữa các vector trước lượng tử hoá và sau lượng tử hoá là nhỏ nhất:

$$d^2(f,\hat{f}) = \sum_{i=1}^{10} w_i (f_i - \hat{f}_i)^2 \tag{2.8}$$

Trong đó:

$$w_i = \begin{cases} P(f_i)^{0.3}, 1 \le i \le 8 \\ 0.64P(f_i)^{0.3}, i = 9 \\ 0.16P(f_i)^{0.3}, i = 10 \end{cases} \tag{2.9}$$

f_i là thành phần thứ i của vector LSF chưa lượng tử hoá và $P(f_i)$ là phổ công suất bộ lọc dự đoán ngược được đánh giá tại tần số f_i. Thủ tục tìm kiếm các cector lượng tử hoá trên là một xấp xỉ tối ưu M (M = 8) với một tìm kiếm đầy đủ. Vector mã tối ưu tìm được ở mỗi tầng được lưu lại để sử dụng với tầng tiếp theo.

Sau đó, các vector LSF đã lượng tử hoá được sắp xếp theo thứ tự tăng dần và đảm bảo khoảng cách nhỏ nhất 50 Hz như đã mô tả ở trên. Các vector tìm được sẽ được sử dụng trong tính toán biên độ Fourier.

Bước 14: Lượng tử hoá pitch

Giá trị pitch cuối cùng P_3 trong dải 20÷160 mẫu được lượng tử hoá logarit với bộ lượng tử hoá đều 99 mức. Sau đó, giá trị pitch được ánh xạ sang một từ mã 7bit sử dụng bảng tra. Từ mã toàn các bit 0 thể hiện trạng thái unvoice, khi Vbp_1 nhỏ hơn 0.6. Tất cả 28 từ mã với trọng số Hamming 1 hay 2 được dành để chống lỗi.

Bước 15: Lượng tử hoá độ khuếch đại

Có hai giá trị độ khuếch đại cần lượng tử háo là G_1 và G_2.

Để lượng tử hoá G_2, người ta sử dụng một bộ lượng tử hoá đều 5bit trong dải từ 10÷77dB.

Đối với G_1, người ta sử dụng một bộ lượng tử hoá 3bit theo thuật toán thích nghi sau:

Một khung sẽ nằm trong đoạn tiếng nói ổn định nếu thoả mãn hai điều kiện sau:

1) G_2 của khung đó nằm trong khoảng 5 dB xung quanh G_2 của khung trước

2) G_1 của khung đó nằm trong khoảng 3 dB xung quanh giá trị trung bình của 2 giá trị G_2 khung đó và khung trước.

Khi đó, một từ mã toàn bit 0 được gửi đi để chỉ thị cho khối giải mã đặt G_1 bằng giá trị trung bình G_2 của khung hiện tại và khung trước.

Nếu không, khung hiện tại nằm trong đoạn chuyển tiếp của tiếng nói và G_1 sẽ được lượng tử bằng một bộ lượng tử đều 7bit trong dải giữa hai giá trị G_2 của khung hiện tại và khung trước được mở rộng 2x6 dB theo hai phía và nằm trong dải từ 10÷77 dB. Đoạn mã giả sau mô tả thuật toán lượng tử thích nghi G_1:

```
if (|G2 - G2p| < 5.0 and |G1 - 0.5 *(G2 + G2p)| < 3.0)
quantizer_index = 0
else
gain_max = max(G2p, G2) + 6.0
gain_min = min(G2p, G2) - 6.0
if (gain_min < 10.0) gain_min = 10.0
if (gain_max > 77.0) gain_max = 77.0
quantizer_index values 1 to 7 are determined by quantizing G1 with a 7-level, uniform quantizer ranging from gain_min to gain_max
endif
```

Bước 16: Lượng tử hoá cường độ voice các dải tần

Khi Vbp_1 nhỏ hơn hay bằng 0.6 (unvoice) thì Vbp_i, i = 1÷5 được lượng tử hoá bằng 0.

Khi Vbp_1 lớn hơn 0.6 thì cường độ voice các dải còn lại được lượng tử hoá bằng 1 nếu chúng

Có một ngoại lệ là khi Vbp_i, i = 2÷5 lần lượt là 0001 thì Vbp_5 cũng được lượng tử hoá bằng 0.

Bước 17: Tính toán và lượng tử hoá biên độ Fourier

Khối chức năng này xác định biên độ Fourier của mười hài pitch đầu tiên của tín hiệu thặng dự dự đoán. Tín hiệu thặng dư này là tín hiệu tiếng nói đầu vào sau khi đã qua một bộ lọc ngược LPC với các hệ số đã được lượng tử hoá gián tiếp thông qua miền LSF.Việc phân tích sử dụng FFT 512 điểm trên một cửa sổ 200 mẫu.Đầu tiên, một cửa sổ Hamming 200 mẫu sẽ được lấy trên tín hiệu thặng dư nói trên, thêm vào các mẫu không và thực hiện FFT phức.Đầu ra FFT sẽ được chuyển thành dạng thành dạng biên độ-pha, các hài sẽ được xác định bằng thuật toán xác định đỉnh phổ.

Khối xác định đỉnh tìm kiếm cực đại trong đoạn có độ dài $[512/\hat{P}_3]$ mẫu tần số (P_3 là giá trị pitch đã lượng tử). Đoạn tần số này có tâm là ước lượng ban đầu của hài pitch cần tìm cực đại, ước lượng ban đầu đối với hài thứ i là $512i/\hat{P}_3$. Giới hạn số lượng biên độ hài được được tìm kiếm là $\min(10, \hat{P}_3/4)$. Những biên độ này được chuẩn hoá để có RMS = 1.0. Nếu có ít hơn 10 hài được tìm thấy thì các biên độ còn lại được đặt bằng 1.

10 biên độ được xác định ở trên sẽ được lượng tử hoá bằng một bộ lượng tử vector 8bit. Việc tìm kiếm trong sách mã dựa trên tiêu chuẩn khoảng cách Euclidean có lấy trọng số cảm quan nhỏ nhất với các trọng số cảm quan cố định, chúng có tác dụng nhấn mạnh thành phần tần số thấp đối với các thành phần tần số cao:

$$w_i = \left[\frac{117}{25 + 75\left(1 + 1.4\left(\frac{f_i}{1000}\right)^2\right)^{0.69}} \right]^2, i = 1, 2, \ldots, 10 \qquad (2.10)$$

Trong đó $f_i = 8000i/60$ là tần số (Hz) tương ứng với hài thứ i khi chu kì pitch ngầm định bằng 60 mẫu.

2.3.2.2. Quá trình giải mã MELP

Bộ giải mã tiếng nói MELP dựa trên mô hình tham số mã hoá dự đoán tuyến tính LPC truyền thống và được bổ xung 5 đặc điểm sau:

1) Kích thích hỗn hợp

2) Xung không tuần hoàn

3) Tăng cường phổ thích nghi

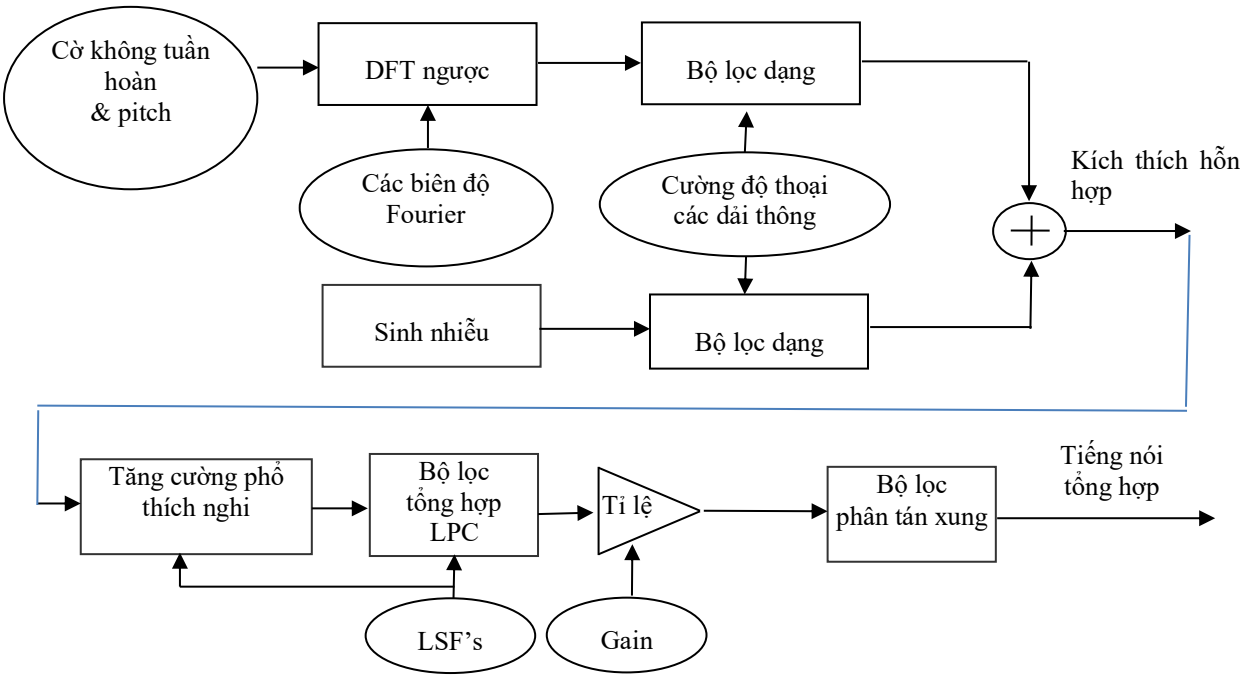
4) Phân tán xung

5) Mô hình hoá biên độ phổ Fourier

Các đặc điểm này được thể hiện ở sơ đồ khối bộ giải mã MELP trong Hình 2.3.

Kích thích hỗn hợp: đây là đặc điểm chủ yếu nâng cao chất lượng của MELP so với vocoder LPC truyền thống như đã phân tích ở trên. Điều này được thực hiện bằng cách sử dụng mô hình trộn đa băng. Mô hình này có thể tái tạo cường độ voice phụ thuộc tần số bằng cách sử dụng một cấu trúc lọc thích nghi theo từng khung tín hiệu. Cấu trúc này gồm hai bộ lọc dạng cho hai nguồn kích thích voice và unvoice, hai các bộ lọc này được xây dựng từ một dàn lọc thông dải cố định. Sự góp phần của mỗi bộ lọc thông dải vào bộ lọc dạng phụ thuộc quyết định voice/unvoice trong dải thông tương ứng, tức là bộ lọc dạng cho kích thích voice được xây dựng. Tác dụng chính của kích thích trộn là giảm các âm ồn ù của vocoder LPC truyền thống.

Xung không chu kỳ: khi tiếng nói đầu vào là voice, bộ mã hoá MELP có thể dùng các xung tuần hoàn hay không tuần hoàn để tái tạo lại tiếng nói. Các xung không tuần hoànđược sử dụng trong các vùng chuyển tiếp giữa voice và unvoice của tiếng nói. Đặc điểm này cho phép các bộ giải nén tái tạo lại các nguyên âm không ổn định mà không đưa vào giọng nói tổng hợp các âm giọng nói khác.



Hình 2.3. Sơ đồ khối giải mã MELP [11]

Bộ lọc tăng cường phổ thích nghi: dựa trên các đặc điểm của bộ lọc tổng hợp dự đoán tuyến tính. Bộ lọc này được sử dụng để tăng cường cấu trúc formant của tiếng nói tổng hợp và làm cho dạng sóng tiếng nói tổng hợp gần hơn với tiếng nói tự nhiên ban đầu hơn, tức là làm cho tiếng nói tổng hợp giống với tiếng nói tự nhiên hơn.

Phân tán xung: được thực hiện sử dụng một bộ lọc cố định. Bộ lọc này có các hệ số suy ra từ một xung tam giác làm phẳng phổ, nó có tác dụng trải năng lượng kích thích trong một chu kì pitch, giảm các âm chói gây khó chịu của tiếng nói tổng hợp.

Mười biên độ Fourier đầu tiên: được xác định từ các đỉnh của biến đổi Fourier trên tín hiệu thặng dư dự đoán. Các thông tin chứa trong các hệ số này cải thiện tính chính xác của mô hình tạo tiếng nói tại các tần số thấp đóng vai trò quan trọng trong cảm thụ tiếng nói. Điều này làm tăng chất lượng tiếng nói tổng hợp, đặc biệt đối với giọng nói nam và khi có nhiễu nền.

Quá trình giải mã Melp được thực hiện qua các bước cơ bản được mô tả dưới đây:

Bước 1: Lấy các bit thông tin và sửa lỗi.

Các bit thu được qua kênh truyền dẫn sẽ được lấy ra và tổ chức thanh các từ mã tham số. Quá trình giải mã tham số đối với hai mode voice và unvoice là khác nhau.

Pitch sẽ được giải mã đầu tiên bởi nó chứa các thông tin về các mode này. Nếu từ mã pitch chỉ gồm toàn bit 0 hay chỉ duy nhất một bit 1 thì mode unvoice sẽ được sử dụng. Nếu có hai bit 1 thì cờ chỉ thị xoá khung sẽ được lập. Trong các trương hợp khác thì giá trị pitch sẽ được giải mã và mode voice sẽ được sử dụng.

Trong mode unvoice, mã Hamming (8, 4) được giải mã để sửa các lỗi một bit và phát hiện các lỗi 2 bit. Nếu phát hiện một lỗi khổng thể sửa, cờ xoá khung sẽ được lập. Nếu không, các mã Hamming (7, 4) được giải mã, sửa các lỗi một bit (khi không có lỗi hai bit nào).

Nếu cờ xoá khung được lập trong khung hiện tại do mã Hamming, từ mã pitch không hợp lệ hay được chỉ ra từ kênh truyền dẫn thì một cơ chế lặp lại khung được thực hiện. Tất các các tham số cho khung hiện tại sẽ được thay thế bằng các tham số của khung trước. Ngoài ra, dộ khuếch đại đầu tiên được đặt bằng dộ khuếch đại thứ hai để loại trừ sự chuyển tiếp mức khuếch đại.

Nếu cờ xoá khung không được lập, các tham số còn lại sẽ được giải mã. Các tần số phổ đường LSF được kiểm tra theo điều kiện thứ tự tăng dần và khoảng cách tối thiểu như mô tả ở phần 3.5 Trong mode unvoice, các giá trị tham số ngầm định được sử dung cho pitch, độ dịch (jitter), cường độ voice các dải tần và các biên độ Fourier. Giá trị pitch được đặt bằng 50 mẫu, jitter bằng 25%, tất cả các cường độ voice dải tần đều được đựat bằng 0 và các biên dộ Fourier được đặt bằng 1. trong mode voice, Vbp_1 được đặt bằng 1, jitter được đặt bằng 25% nếu cờ không tuần hoàn được đặt bằng 1, nếu không jitter được đặt bằng 0%. Cường độ voice đối với bốn dải tần được đặt bằng 1 nếu các bit tương ứng là 1 và ngược lại. Có một ngoại lệ là khi các bit tương ứng Vbp_i, i = 2, 3, 4, 5 là 0001 thì Vbp_5 được đặt bằng 0.

Khi mã của tham số khuếch đại thứ nhất G_1 nhận được gồm toàn bit 0 thì một số lỗi trong tham số khuếch đại thứ hai G_2 có thể được phát hiện và sửa lại. Quá trình sửa lỗi này cải thiện chất lượng trong điều kiện lỗi kênh. Quá trình này được minh hoạ bằng đoạn mã giả sau:

```
inputs: G1_index, G2_index
outputs: G1, G2
internal: G2p, G2p_error
G2 = decode(G2_index)          32 levels; range: 10 to 77 dB
if (G1_index == 0)             special G1 code: use mean of G2 and G2p
if (|G2 - G2p| > 5)            G2_index probably in error
if (G2p_error == 0)            G2p is correct
G2 = G2p                       replace the erroneous G2 with past value
endif
G2p_error = 1
else
    G2_index probably correct
G2p_error = 0
endif
G1 = 0.5 * (G2 + G2p)          mean of G2 and G2p
else
G1 = decode(G1_index) 7 levels; range: min(G2, G2p)-6 to max(G2,G2p)+6
G2p_error = 0      (above range is clamped to 10 to 77 dB)
endif
G2p = G2                       save for use as past value
```

Bước 2: Suy giảm nhiễu

Đối với tín hiệu đầu vào không có nhiễu nền, cả hai độ khuếch đại sau khi giải mã sẽ được suy giảm một giá trị nhỏ sẻ dụng luật trừ công suất. Đây là một trường hợp đơn giản hoá, bất biến theo tần số của phương pháp triệt nhiễu trừ phổ làm trơn (Smoothed Spectral Subtration noise suppression).

Trước khi xác định độ suy giảm cho độ khuếch đại thứ nhất G_1, người ta sẽ cập nhật lại ước lượng nhiễu nền G_n từ giá trị trước của nó theo thủ tục sau:

Nếu $G_1 > G_n + C_{up}$ thì $G_n = G_n + C_{up}$

Nếu$G_1 < G_n - C_{down}$ thì $G_n = G_n - C_{down}$

Nếu không $G_n = G_1$

$C_{up} = 0.0337435$

$C_{down} = 0.135418$

Bộ ước hượng nhiễu này tăng 3 dB/s và giảm 12 dB/s với tốc độ cập nhật độ khuếch đại là 88.9 lần/s. giá trị ước lượng giới hạn trong khoảng 10÷80. Ước lượng nhiễu nền cũng được sử dụgn trong tính toán tăng cường phổ thích nghi (phần 6.4.5).

Độ khuếch đại G_1 được điều chỉnh bằng cách trừ đi một đại lượng hiệu chỉnh dương G_{att} theo dB:

$$G_{att} = -10\log_{10}\left(1-10^{0.1[G_n+3-G_1]}\right) \quad (2.11)$$

trong đó G_n là ước lượng nhiễu nền (theo db) và G_1 là thành phần khuếch đại đầu tiên (theo dB).

Đại lượng hiệu chỉnh này có giới hạn trên là 6 dB để tránh sự thay đổi bất thường là nhiễu tín hiệu. Để đảm bảo chỉ những tín hiệu không bị nhiễu nền mới bị suy giảm, giá trị G_n được sử dụng trong phương trình trên bị cắt tại cần trên là 20 dB.

Ước lượng nhiễu và các bước điều chỉnh độ khuếch đại được lặp lại đối với dộ khuếch đại G_2. Chú ý rằng các khung lặp lại sẽ không được ước lượng nhiễu và suy giảm độ khuếch đại.

Bước 3: Nội suy tham số

Tất cả các tham số tổng hợp MELP đều được nội suy đồng bộ với pitch trong mỗi chu kỳ pitch được tổng hợp. Các tham số được nội suy bao gồm: độ khuếch đại (dB), các tần số phổ đường LSF, pitch, jitter, các biên độ Fourier, xung và các hệ số nhiễu cho kích thích hỗn hợp, và các hệ số điều chỉnh phổ của bộ lọc tăng cường phổ thích nghi.

Độ khuếch đại được nội suy tuyến tính và tuỳ theo thời điểm bắt đầu t_0 ($t_0 = 0$, 1,..179) của chu kỳ pitch mới. Nếu t_0 nhỏ hơn 90 thì độ khuếch đại được nội suy giữa độ khuyếch đại thứ hai của khung trước G_{2p} và độ khuếch đại thứ nhất của khung hiện tại G_1, trong trường hợp khác là giữa G_1 và G_2. Các tham số khác nói chung sẽ

được nội suy tuyến tính giữa các giá trị của khung hiện tại và khung quá khứ. Hệ số nội suy int đối với các tham số này dựa trên điểm bắt đầu của chu kỳ pitch mới:

$$\text{int} = t_0/180 \quad (2.12)$$

Có hai ngoại lệ trong thủ tục nội suy liên quan đến sự thay đổi lớn trong các tham số khi bắt đầu một chu kì pitch là:

Thứ nhất, nếu có sự thay đổi mạnh với tần số pitch cao thì giá trị pitch mới sẽ được dùng ngay mà không qua nội suy. Điều kiện là G_1 lớn hơn G_{2p} 6 dB và chu kỳ pitch của khung hiện tại nhỏ hơn một nửa chu kỳ pitch của khung trước.

Thứ hai, nếu G_2 thay đổi lớn hơn 6 dB thì các tần số phổ đường LSF và pitch được nội suy dựa trên việc nội suy các hệ số khuyếch đại. Đó là do hệ số khuếch đại được truyền đi hai lần đối với mỗi khung, vì vậy sẽ nội suy chính xác hơn. Trong trường hợp này hệ số nội suy là:

$$\text{int} = \frac{G_{\text{int}} - G_{2p}}{G_2 - G_{2p}} \quad (2.13)$$

Bước 4: Tạo kích thích hỗn hợp.

Kích thích hỗn hợp được tạo ra là tổng của các kích thích nhiễu và xung sau khi đã lọc. Kích thích xung $e_p(n)$, n = 0, 1,..T-1 được tính qua IDFT có độ dài một chu kỳ pitch:

$$e_p(n) = \frac{1}{T}\sum_{k=0}^{T-1} M(k) e^{j2\pi nk/T} \quad (2.14)$$

Chu kỳ pitch T là giá trị pitch nội suy cộng với tích của hệ số jitter và giá trị pitch đó. Trong đó jitter bằng giá trị jitter đã nội suy nhân với một số ngẫu nhiên trong đoạn [-1,1]. Chu kỳ pitch được làm tròn tới số nguyên gần nhất và giới hạn trong khoảng 20÷160.

Pha của tất cả các kích thích xung được đặt bằng 0 do M(k) thực. Vì $e_p(n)$ thực nên

$$M(T-k) = M(k) \quad k= 1, 2,\ldots L \quad (2.15)$$

Trong đó, L = [T/2]

Thành phần một chiều M(0) được đặt bằng 0. Các thành phần biên độ M(k), k = 1, 2,…10 được đặt bằng giá trị nội suy của biên độ Fourier, các thành phần biên độ khác được đặt bằng 1.

Để tránh sự thay đổi đột ngột khi bắt đầu chu kì pitch, kích thích xung được dịch vòng 10 mẫu để xung kích thích chính có tại mẫu thứ 10 của chu kì. Xung này được nhân với căn bình phương của chu kì pitch để có tính hiệu có RMS đều và sau đó được nhân với 1000 để có mức tín hiệu danh định.

Nhiễu được một bộ sinh số ngẫu nhiên đều có RMS bằng 1000 và nằm trong dải từ -1732÷1732.

Các tín hiệu kích thích xung và nhiễu sau đó được lọc và lấy tổng lại tạo thành kích thích trộn. Bộ lọc cho kích thích xung có các hệ số được lấy tổng từ tất cả các hệ số tương ứng của các bộ lọc thông dải ở các dải tần đã quyết định là voice. Tương tự, bộ lọc cho kích thích nhiễu có các hệ số được lấy tổng từ tất cả các hệ số tương ứng của các bộ lọc thông dải ở các dải tần đã quyết định là unvoice. Các hệ số bộ lọc đều được nội suy đồng bộ với chu kì pitch.

Bước 5: Tăng cường phổ thích nghi

Tín hiệu kích thích hỗn hợp được tạo ra ở trên tiếp theo sẽ được cho qua bộ lọc tăng cường phổ thích nghi. Bộ lọc này là bộ lọc cực/không bậc 10 cộng với một thành phần bù bậc 1. Các hệ số của nó được tính từ hàm truyền được mở rộng dải thông của bộ lọc dự đoán tuyến tính.

Từ các tần số phổ đường sau khi nội suy, ta có hàm truyền của bộ lọc dự đoán tuyến tính A(z). Tiếp theo, bàm truyền của bộ lọc tăng cường $H_{ase}(z)$ được tính theo công thức:

$$H_{ase}(z) = \frac{A(\alpha z^{-1})}{A(\beta z^{-1})}(1 + \mu z^{-1}) \qquad (2.16)$$

trong đó

$\alpha = 0.5\ p$

$\beta = 0.8\ p$

và hệ số điều chỉnh μ bằng giá trị của *max*(0.5k_1, 0), nội suy và nhân với xác suất p của tín hiệu. Hệ số phản xạ thứ nhất k_1 được tính từ các tham số tần số phổ đường LSF đã giải mã. Do quy ước về dấu của các hệ số của bộ dự đoán MELP, k_1 thường âm đối với các phổ voice. Xác suất tín hiệu p được ước lượng bằng cách so sanh độ khuếch đại nội suy hiện tại G_{int} với ước lượng nhiễu nền $G_n$theo công thức:

$$p = \frac{G_{\text{int}} - G_{\text{n}} - 12}{18} \tag{2.17}$$

Xác suất tín hiệu nằm trong đoạn [0, 1].

Bước 6: Tổng hợp dự đoán tuyến tính

Bộ tổng hợp sử dụng một bộ lọc dạng trực tiếp với các hệ số tương ứng với các tham số tần số phổ đường LSF đã được nội suy.

Bước 7: Điều chỉnh độ khuếch đại

Do tín hiệu kích thích được tạo ra tại một ngưỡng bất kì nên cần phải dưa độ khuếch đại tiếng nói vào tiếng nói tổng hợp. Hệ số diều chỉnh tỉ lệ s_{gain} được tính cho mỗi chu kì pitch được tổng hợp độ dài T băng cách chia giá trị RMS mong muốn ($G_{int\ càn}$ phải được chuyển từ dB) cho giá trị RMS của tín hiệu tiếng nói tổng hợp chưa được điều chỉnh tỉ lệ $\hat{s}_n$:

$$S_{gain} = \frac{10^{G_{\text{int}}/20}}{\sqrt{\frac{1}{T}\sum_{n=1}^{T}\hat{S}_n^2}} \tag{2.18}$$

Để tránh hiện tượng không liên tục trong tiếng nói tổng hợp, hệ số điều chỉnh tỉ lệ này được nội suy giữa các giá trị trước và hiện tại đối với 10 mẫu đầu tiên của chu kì pitch.

Bước 8: Phân tán xung

Bộ lọc phân tán xung là một bộ lọc FIR bậc 65. Các hệ số của bộ lọc này được tính từ một xung tam giác làm phẳng phổ.

Bước 9: Điều khiển vòng lặp tổng hợp

Sau khi xử lý mỗi chu kì pitch, bộ giải mã cập nhật t_0 băng cách cộng vào số mẫu T trong chu kì vừa được tổng hợp. Nếu t_0 nhỏ hơn 180, quá trình tổng hợp đối với

khung hiện tại tiếp tục từ bước nội suy tham số. Nếu không, bộ giải mã giữ lại phần còn lại của chu kì pitch hiện tại nằm sau khung hiện tại trong bộ đệm và lấy giá trị khởi tạo đầu cho khung tiếp theo là $t_0 - 180$.

2.3.3. Đề xuất bộ mã hoá MELP cải tiến tốc độ thấp

Điểm mấu chốt trong thỏa hiệp giữa chất lượng và tốc độ của bộ mã thoại là độ chính xác của xác định pitch vì pitch không chỉ xác định chính xác tần số cơ bản mà còn ảnh hưởng đến việc nội suy tất cả các tham số khác [*19*]. Phương pháp xác định pitch cải tiến thực hiện với giá trị tối ưu của chu ki pitch và các tham số LPC tối ưu là các giá trị sao cho sai số nhỏ nhất giữa phổ gốc $|S_w(\omega)|$ và phổ tổng hợp $|\hat{S}_w(\omega)|$:

$$\varepsilon = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left[\left|S_w(\omega)\right| - \left|\hat{S}_w(\omega)\right|\right]^2 d\omega \tag{2.19}$$

Trong đó biến đổi Fourier thời gian ngắn $S_w(\omega)$ của một đoạn tín hiệu $s_w(n)$ được lấy bởi cửa sổ $w(n)$ sẽ được mô hình hoá là tích của đường bao phổ $H_w(\omega)$ và phổ kích thích $E_w(\omega)$:

$$\hat{S}_w(\omega) = H_w(\omega).|E_w(\omega)| \tag{2.20}$$

Đường bao phổ được biểu diễn bằng các hệ số dự đoán tuyến tính LPC. Phổ kích thích biên độ $|E_w(\omega)|$ là phổ của kích thích hỗn hợp các xung tuần hoàn và nhiễu được ước lượng ở năm dải tần của tín hiệu tiếng nói đầu vào.

Để thực hiện, phổ tín hiệu được chia đều vào các dải tần có tâm tại các hài của tần số cơ bản ứng với pitch đang xét. Để đơn giản, mô hình hoá đường bao phổ là một hằng số trong khoảng hài thứ m với giá trị A_m. Khi đó, sai số phổ (2.19) trong khoảng tần số xung quanh hài thứ *m* bằng:

$$\tilde{\varepsilon} = \frac{1}{2\pi}\int_{a_m}^{b_m}\left[\left|S_w(\omega)\right| - \left|A_m\right|.\left|E_w(\omega)\right|\right]^2 d\omega \tag{2.21}$$

và sai số phổ trên toàn dải tần ứng với một chu kì pitch đang xét là:

$$\tilde{\varepsilon} = \sum_m \tilde{\varepsilon}_m \tag{2.22}$$

Một vấn đề khác, là hiện tượng thay đổi pitch đột ngột do ảnh hưởng của cấu trúc hài hay nhiễu. Để giải quyết vấn đề này, phương pháp bám pitch được thực hiện theo quy hoạch động, khắc phục được hiện tượng trên nhưng không làm mất độ chính xác của pitch như các phương pháp làm trơn thông thường khác.

Để thực hiện phương pháp quy hoạch động, các tham số pitch và sai số phổ tương ứng của bốn khung được sử dụng bao gồm: khung quá khứ, khung hiện tại, và hai khung tương lai. Trong mỗi khung, chọn ra năm pitch $P_j[i]$ (i = 0÷4) có sai số phổ tương ứng nhỏ nhất $V_j[i]$, j là chỉ số của khung. Mục đích cuối cùng là tìm là pitch thực sự của khung hiện tại.

Một đường pitch qua các khung được đánh giá bằng một hàm trọng số theo phương pháp quy hoạch động gồm hai thành phần:

(1) Các trọng số của các node mà đường pitch đó đi qua: là sai số phổ tương ứng;

(2) Khoảng cách giữa hai node liên tiếp (thuộc hai khung liên tiếp) trên đường pitch: là chênh lệch giữa các pitch.

Công thức xác định trọng số của một đường pitch qua bốn khung (j = 0÷3) là:

$$W_{path} = \sum_{j=0}^{3} cV_j + \sum_{j=1}^{3} \left| P_j - P_{j-1} \right| \tag{2.23}$$

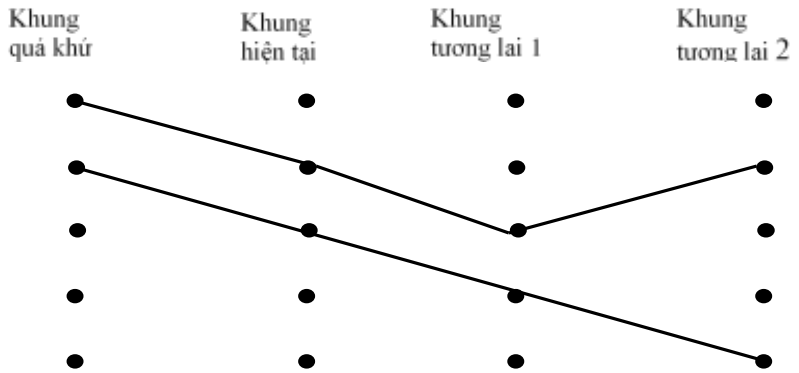
Trong đó, c là hằng số để cân bằng trọng số giữa các node và các khoảng cách có dải động khác nhau.

Theo phương pháp quy hoạch động, pitch cần tìm trong khung hiện tại sẽ nằm trong đường pitch có trọng số nhỏ nhất. Hai bước thực hiện chính như sau:

Bước 1: Với mỗi pitch trong năm pitch của khung hiện tại, ta tìm đường pitch qua nó có trọng số nhỏ nhất;

Bước 2: So sánh trọng số của năm đường pitch, pitch cần tìm sẽ nằm trên đường pitch có trọng số nhỏ nhất.

Lưu ý là do chỉ bám pitch trên các khung voice liên tục nên nếu đường pitch gặp một khung unvoice thì sẽ kết thúc ngay ở đó.



Hình 2.4. Bám pitch theo phương pháp quy hoạch động

Chi tiết các bước được thể hiện như sau:

1. Chọn năm pitch có sai số phổ nhỏ nhất của khung tương lai thứ hai. Giả thiết rằng ta cũng đã có các dãy pitch ở các khung quá khứ, hiện tại và tương lai thứ nhất;

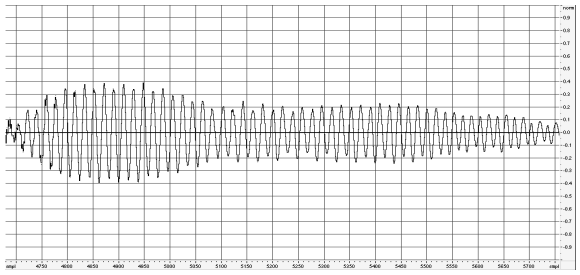
2. Tìm node ở khung hiện tại có đường pitch đi qua nó có trọng số nhỏ nhất;

3. Cập nhật lại giá trị pitch và sai số phổ các khung sẵn sàng cho khung tiếp theo.

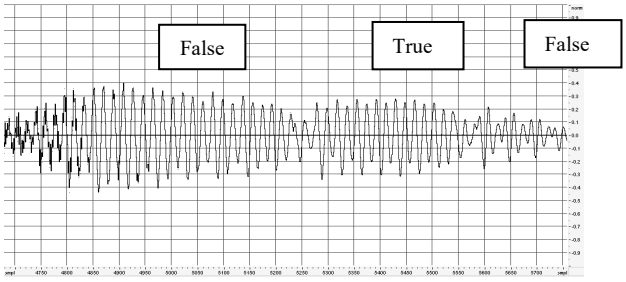
Kết quả mô phỏng

So sánh tiếng nói tổng hợp của bộ mã hoá iMELP cải tiến sử dụng thuật toán xác định pitch mới nói trên so với MELP chuẩn cho thấy tiếng nói tổng hợp nghe rõ và tự nhiên hơn. Có thể nhận thấy sự khác biệt rõ khi nghe so sánh hai tiếng nói tổng hợp dùng MELP và IMELP. Hình 2.5 thể hiện ưu điểm của MELP cải tiến so với MELP. Các âm đột biến khó chịu do xác định sai pitch (thể hiện rõ nhất tại vùng cạnh đường chấm, tức là các vùng mà MELP bắt đầu xác định sai pitch) trong MELP đã được loại trừ trong MELP cải tiến.

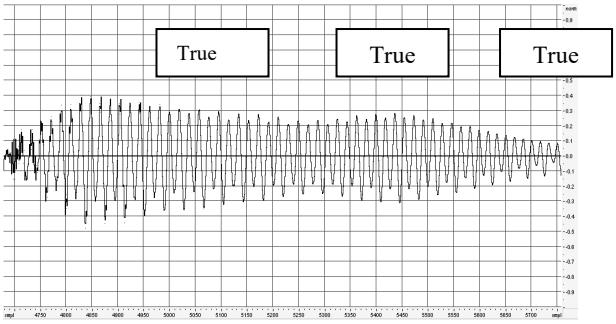
Bước tìm pitch nguyên không yêu cầu tìm ngay được pitch thực sự, giá trị pitch tìm được trong bước này là bội hay chỉ gần pitch thực sự cũng là đạt yêu cầu. Thủ tục tìm pitch phân và tìm pitch cuối cùng sẽ xác định làm tinh và xác định pitch thực sự



(a)



(b)

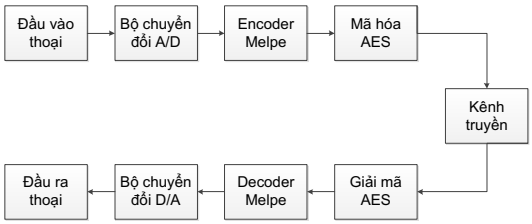


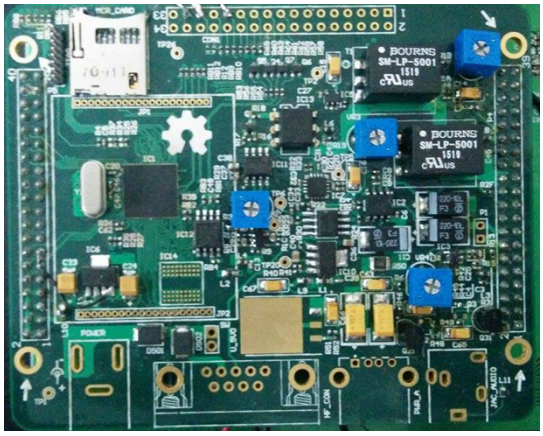
(c)

Hình 2.5. So sánh chất lượng MELP chuẩn và iMELP cải tiến; (a) Tín hiệu gốc; (b) Tín hiệu MELP chuẩn; (c) Tín hiệu iMELP cải tiến ở tốc độ 1200bps

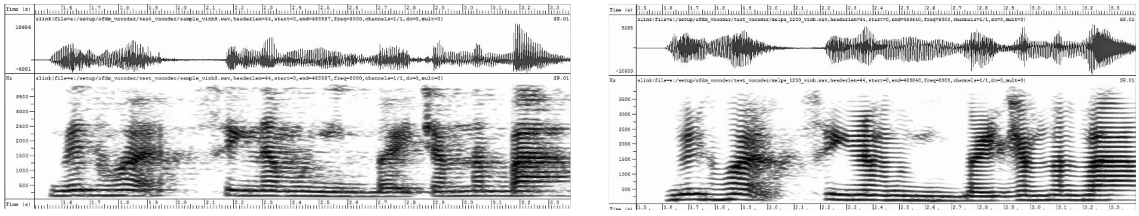
Kiến trúc phần cứng thực nghiệm

Bộ nén thoại vocoder iMELP được thực hiện trên nền tảng ARM. Quá trình thực thực thi nén thoại và bộ giải nén được thực hiện trên chip của hãng ST là STM32F437. Chip STM32F437 sử dụng công nghệ ARM Cortex-M4 nền tảng RISC 32bit có tốc độ Clock 180MHz, hỗ trợ dấu chấm động, bộ nhớ trong 2MB. Toàn bộ quá trình được thực hiện trên vi xử ARM theo sơ đồ khối sau:



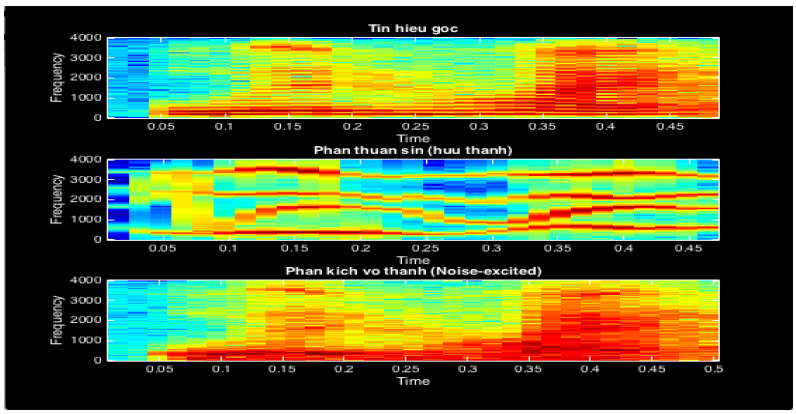


Hình 2.6. Phần cứng thực hiện nén và mã thoại Melpe



(a) (b)

Hình 2.7. (a) Dữ liệu thoại đầu vào trong 3,3s; (b) Dữ liệu thoại sau khi nén bằng iMELP cải tiến tốc độ 1200bps



Hình 2.8. Phân tích phổ tín hiệu vô thanh và hữu thanh

Dựa trên tiêu chuẩn đánh giá chất lượng thoại theo PESQ của ITU, đánh giá xem bộ nén thoại MELP cùng với LPC10 (trong bộ Matlab) được như bảng sau:

Thoại gốc	Đã nén	MOS
ORG_nguyen_hue.wav	iMelp_nguyen_hue.wav	2.066
org_nh.wav	iMelp_nh.wav	2.545
sample_vinh8.wav	iMelp_1200_vinh.wav	2.614
ORG_nguyen_hue.wav	LPC_nguyen_hue.wav	1.437
org_nh.wav	LPC_org_nh.wav	1.459

2.4. Giải pháp điều chế và giải điều chế để truyền dữ liệu qua kênh thoại GSM

2.4.1. Phương pháp điều chế tín hiệu tựa tiếng nói

Phương pháp điều chế tín hiệu tựa tiếng nói (speech-like waveform) đã được thử nghiệm và mô tả trong một số bài báo của các tác giả khác nhau [4][25][26][27][30][36],... Đây là phương pháp truyền dữ liệu dưới dạng tổng hợp thành tiếng nói và cơ bản sử dụng 3 đặc tính chính:

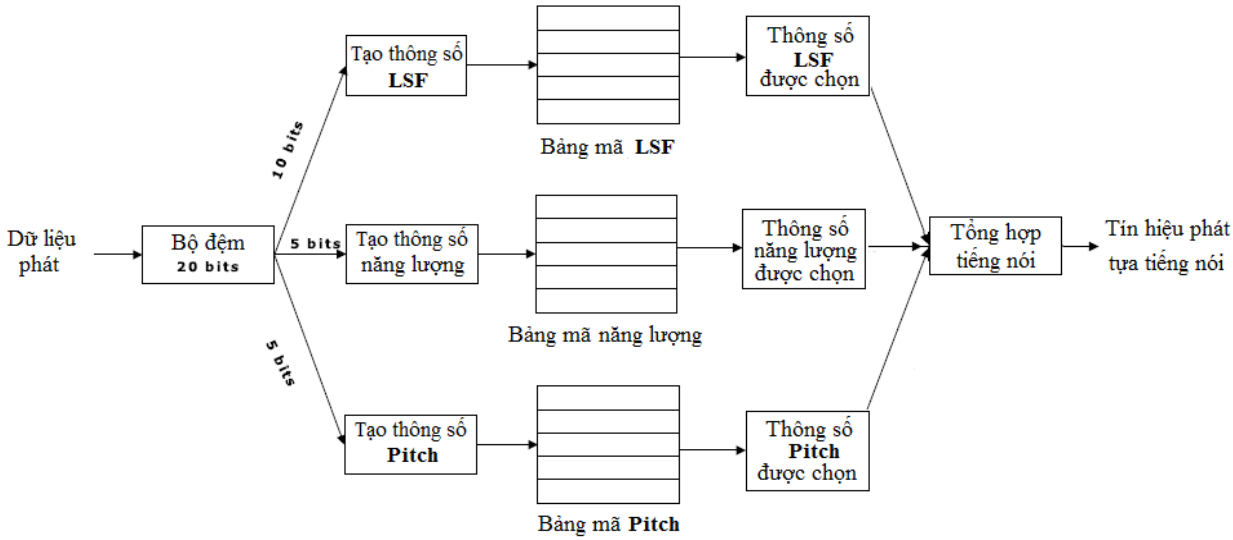
1) Đường bao của phổ tiếng nói được biểu diễn bởi các tần số phổ vạch (LSF).

2) Tần số cơ bản hoặc cao độ của giọng nói (pitch)

3) Hình dạng và năng lượng kích thích ACELP (hoặc CELP)

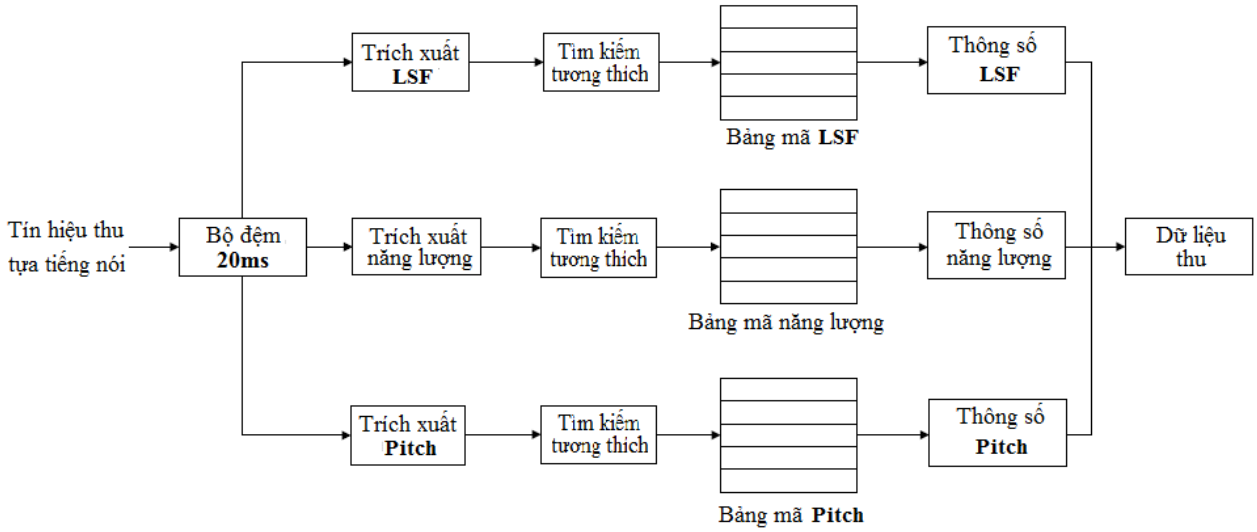
Các thông số nêu trên được bảo tồn khi truyền qua kênh thoại GSM và PSTN.

Dữ liệu đầu vào được ánh xạ tới các thông số trên bằng 3 bảng mã - codebook và sau đó được nhập vào bộ tổng hợp (Hình 2.9). Tiếng nói tổng hợp này không phải là ngôn ngữ của bất kỳ cư dân nào trên thế giới mà nó chỉ có cùng tính chất của tiếng nói trên phương diện nén và giải nén mà thôi.



Hình 2.9. Sơ đồ khối của phương pháp điều chế tín hiệu tựa tiếng nói [5][30]

Tiếng nói với các thông số chủ yếu được tổng hợp và phát đi. Bên thu có bộ phân tích tiếng nói sẽ tách ra các thông số, kiểm tra tính tương thích rồi tra trong bảng mã để lấy ra dữ liệu (Hình 2.10).



Hình 2.10. Sơ đồ khối của phương pháp giải điều chế tín hiệu tựa tiếng nói [30]

Dữ liệu được gán như sau: 10 bit cho LSF, 5 bit cho cao độ và 5 bit cho năng lượng. Tổng cộng là 20 bit được truyền trong 20 ms. Điều này sẽ cho tốc độ bit là 1 kbps. Tốc độ bit cao hơn sẽ đạt được bởi các bảng mã lớn hơn.

Có hai nhiệm vụ chính cần thực hiện trong phương pháp này. Một là chọn loại mã hóa tiếng nói nào sẽ được sử dụng và hai là từ đó thiết kế các bảng mã. Do hệ thống GSM dùng mã nén tiếng nói theo thuật toán CELP – ACELP nên loại mã hóa cùng loại sẽ được chọn. Tại sao nên chọn mã nén cùng loại với hệ thống kênh truyền, là vì các thông số mà dữ liệu ánh xạ vào khi truyền qua hệ thống ít bị sai lệch hơn. Có thể chọn một trong số các mã GSM-HR (VSELP), GSM-EFR, GSM-ARM (ACELP), CELP... hoặc Speedx. GSM-EFR là loại được ưu tiên lựa chọn vì việc triển khai đơn giản hơn. Từ loại mã nén được chọn sẽ quyết định việc thực hiện nhiệm vụ thứ hai là thiết kế bảng mã như thế nào.

Thiết kế bảng mã là công việc phức tạp và tốn nhiều thời gian nhất. Bảng mã thực hiện ánh xạ dữ liệu vào các thông số và sau đó nhập chúng vào bộ tổng hợp tiếng nói.

Có hai phương pháp được sử dụng để điền vào các bảng mã.

Phương pháp biểu đồ.

Ý tưởng ở đây là tạo ra các bảng mã với các tham số được sử dụng thường xuyên nhất và ít sai lệch nhất khi truyền qua hệ từ một đoạn ghi âm mẫu tiếng nói. Giả định ở đây là các tham số được sử dụng thường xuyên nhất và ít sai lệch nhất khi truyền

sẽ dễ dàng bảo toàn khi truyền qua hệ thống. Sau khi phân tích lời nói mẫu từ máy phân tích EFR các tham số được trich ra và thống kê. Ví dụ, tham số LSF có 5 chỉ số và lần đầu tiên được lượng tử hóa thành 7 bit. Nếu chúng ta muốn truyền 2 bit trên tham số đó, thì biểu đồ được chia thành 4 khoảng và tối đa được tìm thấy trong mỗi khoảng và đưa vào bảng mã. Điều này được thực hiện để tránh rằng tất cả các lựa chọn rất gần nhau. Điều tương tự cũng được thực hiện cho các chỉ số LSF khác. Do đó, bảng mã là sự kết hợp của tất cả các giá trị tối đa được tìm thấy trong biểu đồ. Điều tương tự được thực hiện cho pitch và năng lượng.

Giải thuật di truyền GA.

Giải thuật di truyền là một phương pháp để tối ưu hóa quá trình dò tìm. Đầu tiên, vấn đề được định nghĩa cẩn thận và theo thuật ngữ của giải thuật di truyền gọi là GA, và được gọi là bộ gen. Một bộ gen có thể được xác định theo nhiều cách và hiệu quả của phương pháp này phụ thuộc vào mức độ giải quyết vấn đề như thế nào. Một tập hợp con của bộ gen sẽ được chọn và được gọi là quần thể và quần thể này sẽ tiến hóa bằng cách sử dụng đột biến và trao đổi chéo. Bộ gen thích nghi sẽ tồn tại. Sau đó, chức năng luyện tập sẽ đánh giá bộ gen nào là tốt nhất và được chọn để phát triển. Phương pháp này không đảm bảo sự hội tụ cho giải pháp tối ưu. Có thể sử dụng bộ công cụ GA trong Matlab để thực hiện các mô phỏng.

Yêu cầu là chọn các mục cho bảng mã có BER thấp nhất. Tuy nhiên, thật khó khăn khi thực hiện vấn đề này về mặt GA. Khi tối ưu hóa LSF thì các tham số khác đươc giữ cố định. Bộ gen sau đó là một bảng mã của các tham số LSF vì bên thu phải có một bảng mã để có thể nhận được. Hàm luyện tập dùng để tính toán BER cho mỗi mục trong bộ gen rồi tính trung bình kết quả. Chúng ta chỉ có thể tối ưu hóa một tham số nhưng để có kết quả tốt nhất, nó phải được tối ưu hóa các tham số đồng thời. Đây là nhiệm vụ vô cùng khó khăn.

Tính ổn định của hệ thống.

Bộ lọc dự đoán thời gian ngắn STP và bộ lọc dự đoán thời gian dài LTP là bộ lọc đáp ứng xung vô hạn IIR và là mô hình toàn cực (All-Pole Model). Việc chọn lựa

các hệ số có thể gây ra sự mất ổn định. Để kiểm tra tính ổn định khi chọn các hệ số có thể áp dụng các tiêu chuẩn sau:

1) Trong miền tần số (biến đổi Fourier): Nyquist

2) Trong miền Z (biến đổi Z): Schur-Cohn

3) Trong miền S (biến đổi Laplace): Routh Hurwitz

Với bậc lọc 10 và bảng mã 1000 phần tử việc kiểm tra mọi tổ hợp các hệ số xem có ổn định không dù là sử dụng máy tính cũng là công việc khổng lồ. Ngoài ra đối với hệ thống nén tiếng nói các hệ số của bộ lọc thu được từ hệ vật lý thật tức là từ lời nói có tính chất thay đổi chậm từ mẫu này đến mẫu khác. Đối với quá trình điều chế tựa tiếng nói, dữ liệu là số bất kỳ nên dữ liệu hiện tại và dữ liệu kế tiếp có thể không liên quan đến nhau. Khi tổng hợp thành tiếng nói thì mẫu tiếng nói trước và mẫu tiếng nói sau có tham số thay đổi đột ngột. Tiếng nói giả này truyền trên GSM có thể bị sai lệch nhiều khi tái tạo lại. Đây cũng là một trở ngại lớn.

2.4.2. Đề xuất phương pháp điều chế tín hiệu kiểu viễn thông truyền thống có cấu trúc phổ gần giống phổ của tiếng nói

2.4.2.1. Điều chế tín hiệu kiểu viễn thông truyền thống

Điều chế theo phương thức viễn thông truyền thống được nhiều tác giả nghiên cứu. Thực nghiệm cho thấy điều chế (số) khóa pha (dịch pha) PSK tốt hơn so với điều chế (số) khóa biên độ (dịch biên) ASK và điều chế (số) khóa tần số (dịch tần) FSK. ASK thay đổi biên độ, trong trường hợp này, bộ mã hóa tiếng nói của GSM có AGC và nó phát hiện các thay đổi về biên độ này sẽ thực hiện việc bù, điều này sẽ gây ra lỗi trong máy thu. FSK cũng không phải là một lựa chọn tốt ở đây vì băng thông rất hạn chế (4 kHz). Điều chế (số) dịch pha vi sai DPSK thường được chọn vì tính đơn giản khi thực hiện và không cần bộ thu kết hợp.

Kênh bị giới hạn băng tần 4 kHz vì tần số lấy mẫu là 8 kHz. Trong hệ thống điện thoại thường có các bộ lọc thông thấp và thông cao, vì vậy tốt nhất trong thực tế chọn tần số sóng mang là tần số trung tâm của băng thông và có thể được xác định bằng

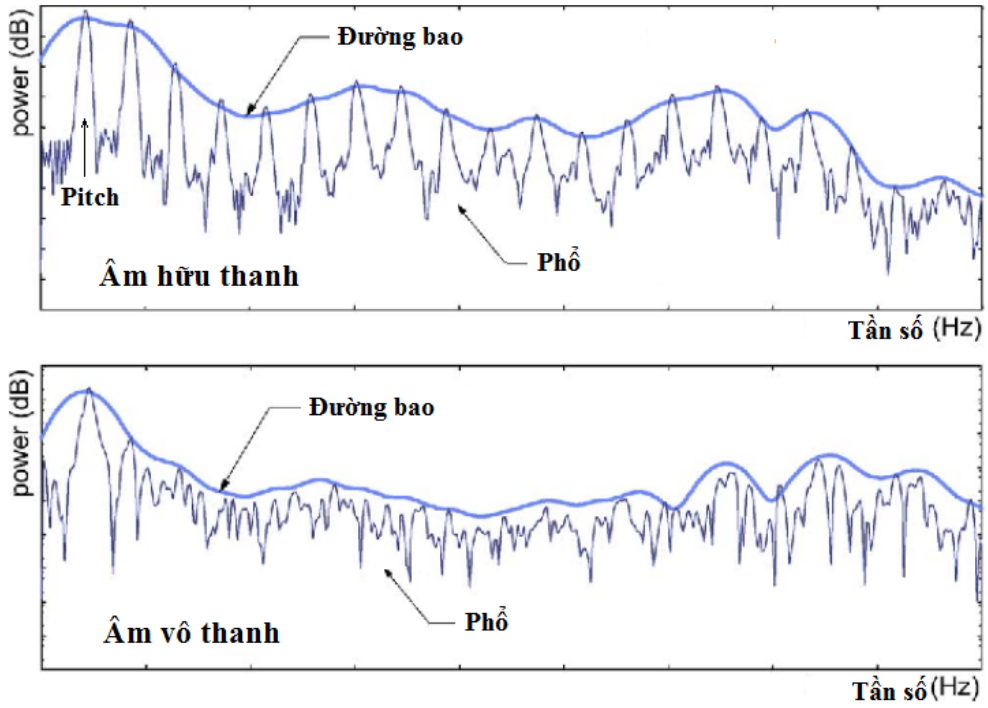
các thiết bị đo lường. Với dải tần 300-3400Hz tần số sóng mang được chọn là 1.8 kHz.

Hạn chế của phương pháp điều chế tín hiệu kiểu viễn thông truyền thống là tốc độ truyền thấp và hiện tượng mất tín hiệu do VAD. Với GSM thời gian đáp ứng của bộ lọc dự đoán thời gian ngắn STP là 5 ms, bộ lọc dự đoán thời gian dài LTP là 20 ms. Như thế thời gian truyền một ký hiệu – symbol không dưới 5 ms. Tần số truyền ký hiệu cực đại sẽ là 200 Hz (1/5ms). Nếu dùng điều chế DPSK thì tốc độ truyền chỉ là 200 bps. Để tăng tốc độ truyền phải tăng mức điều chế và khi đó sai số BER sẽ tăng. Tác động của VAD cũng cần phải được xem xét. Đối với truyền dữ liệu không yêu cầu thời gian thực thì khoảng lặng xuất hiện không thành vấn đề, nhưng với yêu cầu truyền dữ liệu thời gian thực như mật mã thoại chẳng hạn thì không thể chấp nhận được. Để khắc phục người ta thường chèn những đoạn tín hiệu có tính xung để "đánh lừa" bộ VAD. Khi đó phải trả giá bằng tốc độ truyền giảm và không phải lúc nào cũng ổn.

Như phân tích ở trên, kỹ thuật phát hiện tiếng nói tích cực VAD được sử dụng trong GSM. Để tín hiệu modem truyền qua kênh GSM không bị gián đoạn (mất) do VAD tác động thì tín hiệu modem phải có đặc tính sao cho VAD nhận diện như tín hiệu voice. Phương pháp thứ nhất là điều chế tín hiệu tựa tiếng nói speech-like waveform. Phương pháp thứ hai là điều chế theo phương thức viễn thông truyền thống có chèn những đoạn tín hiệu có tính xung để "đánh lừa" bộ VAD. Phương pháp thứ ba là điều chế tín hiệu kiểu viễn thông truyền thống có cấu trúc phổ gần giống phổ của tiếng nói, cụ thể là OFDM. Phổ của OFDM giống phổ của âm hữu thanh nên không cần chèn tín hiệu để đánh lừa bộ VAD. Hơn nữa, nếu lựa chọn số vạch phổ, khoảng cách giữa các vạch phổ nằm trong dải của âm hữu thanh thì dữ liệu sau khi điều chế thành tín hiệu điều chế truyền qua kênh GSM đến máy thu, được máy thu giải điều chế sẽ bảo toàn ít bị sai lệch.

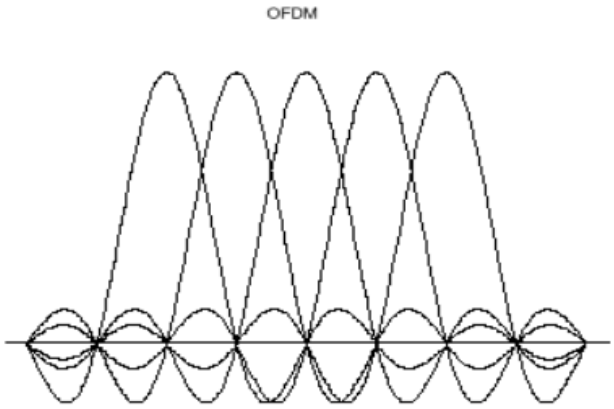
2.4.2.2. Điều chế tín hiệu kiểu viễn thông truyền thống có cấu trúc phổ gần giống phổ của tiếng nói

Hình 2.11 dưới cho thấy phổ của đoạn tiếng nói với âm hữu thanh là phổ có hình răng lược với tần số là bội nguyên lần của tần số của giọng nói pitch. Với đoạn âm vô thanh phổ là phổ của nhiễu. Đường bao trong cả hai trường hợp, là đặc tuyến tần số của các hốc cộng hưởng của cơ quan phát âm (bộ lọc cơ quan phát âm).



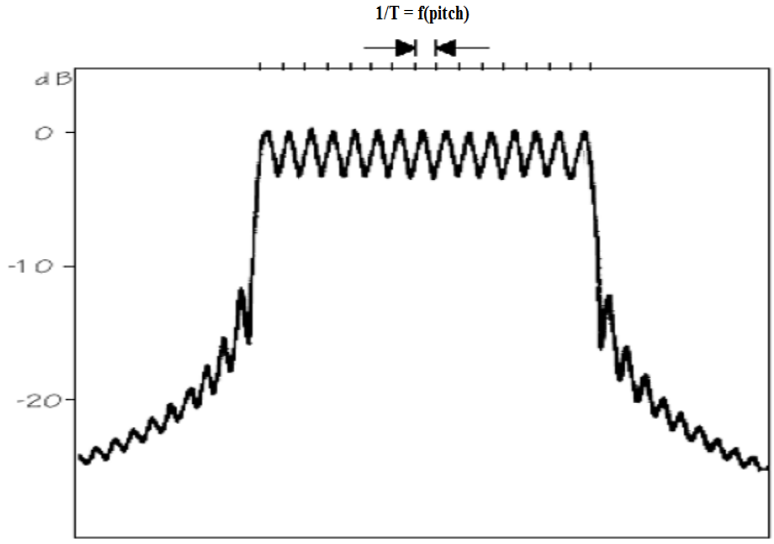
Hình 2.11. Phổ của âm hữu thanh và âm vô thanh

Kỹ thuật điều chế ghép kênh phân chia theo tần số trực giao – OFDM do R.W Chang phát minh năm 1966. OFDM là một trường hợp đặc biệt của phương pháp điều chế đa sóng mang, trong đó các sóng mang phụ trực giao với nhau, nhờ vậy phổ tính hiệu ở các sóng mang phụ cho phép chồng lấn lên nhau mà phía thu vẫn có thể khôi phục lại tín hiệu ban đầu. Sự chồng lấn phổ tín hiệu làm cho hệ thống OFDM có hiệu suất sử dụng phổ lớn hơn nhiều so với kỹ thuật điều chế thông thường.



Hình 2.12. OFDM là một trường hợp đặc biệt của phương pháp điều chế đa sóng mang

Hình 2.12 phổ răng lược của điều chế ghép kênh theo tần số trực giao – OFDM tương tự phổ âm hữu thanh.



Hình 2.13. Phổ điều chế OFDM

Từ phân tích ở trên, công trình này lựa chọn phương pháp điều chế ghép kênh phân chia theo tần số trực giao - OFDM để truyền dữ liệu qua kênh GSM. OFDM đã được nhiều công trình trong và ngoài nước nghiên cứu khá kỹ càng cả trên phương diện lý thuyết và ứng dụng. Dưới đây tác giả xin trình bày nội dung liên quan đến việc lựa chọn các thông số và thực hiện điều chế OFDM sao cho có thể truyền dữ liệu qua kênh thoại GSM. Nhiều vấn đề về chi tiết lý thuyết và kỹ thuật của OFDM đã được trình bày trong các tài liệu khác sẽ không được trình bày ở đây.

Những vấn đề mới được đề xuất là:

1)Lựa chọn các thông số

Như đã nêu ở phần trên để tín hiệu truyền qua kênh thoại GSM được bảo toàn thì tín hiệu phải có cấu trúc phổ giống phổ của tiếng nói và trên phương diện khác cần phải xem xét các đặc điểm xử lý tiếng nói của hệ thống GSM.

Thứ nhất dải phổ của OFDM phải nằm trong dải thoại 300 – 3400 Hz (khi truyền qua GSM/PSTN)

Thứ hai số vạch phổ không nên nhiều quá vì GSM sử dụng kỹ thuật nén dựa trên cơ sở LPC với bậc lọc cố định. Về mặt toán học người ta có thể biểu diễn một hàm đi qua n điểm cho trước bằng một đa thức có (n – 1) bậc. LPC trong GSM có bậc lọc là 10 nên số vạch phổ tốt nhất sẽ là 11. Tuy nhiên ở đây mỗi sóng mang con được điều chế số về pha hoặc biên độ với số mức là 2 (hoặc 4) nên số điểm có thể tăng lên mà ít ảnh hưởng đến chất lượng tín hiệu sau khi hồi phục. *Ở đây số sóng mang con được chọn là 16.*

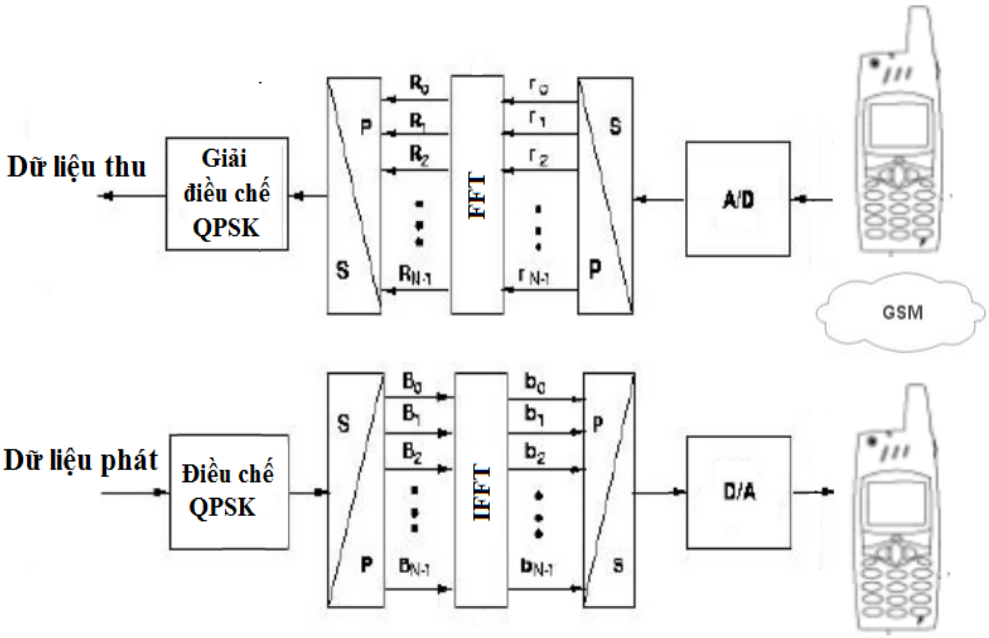
Thứ ba khoảng thời gian truyền một ký hiệu (symbol) không được ngắn hơn thời gian giữa 2 superframe (bao gồm 4 frame) trong GSM là 20ms, tương ứng với tốc độ truyền symbol là 50Hz. Như vậy khoảng cách ngắn nhất giữa các vạch phổ sóng mang phụ của OFDM là 50Hz. *Trong trường hợp này chọn là 75Hz và như vậy dư 25Hz làm khoảng bảo vệ.*

Dải tần của OFDM sẽ là : 75Hz x 16 = 1200Hz ta chọn tần số trung tâm là 1500 Hz như vậy dải phổ của OFDM từ 900Hz đến 2100Hz thỏa mãn điều kiện thứ nhất là dải phổ nằm trong dải phổ của thoại từ 300 đến 3400 Hz.

Thứ tư chọn phương thức điều chế. Như phân tích ở trên, điều biên ảnh hưởng của bộ AGC, còn điều tần phổ quá rộng vì thế chỉ còn điều pha. *Ở đây điều pha QPSK được chọn cho điều chế OFDM truyền qua kênh thoại GSM.*

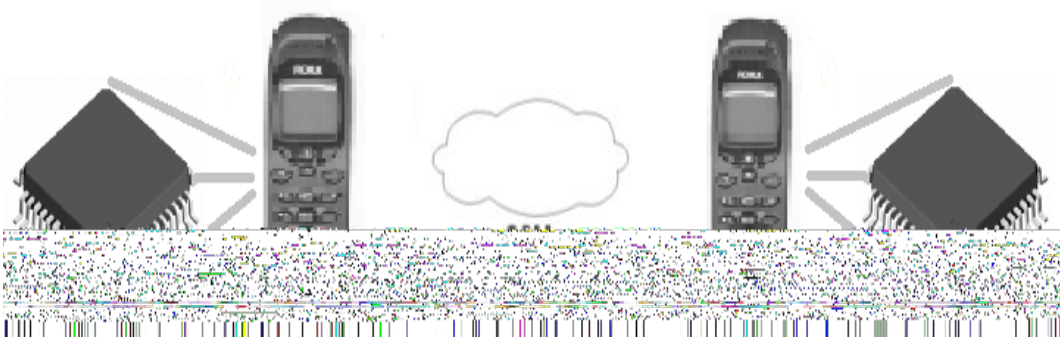
2) *Thực hiện điều chế OFDM với QPSK :*

Việc thiết kế, chế tạo modem OFDM với QPSK có thể được thực hiện ngay trên PC theo sơ đồ Hình 2.14 dưới đây. Phần còn lại là bài toán kinh điển của OFDM, nó đã được nhiều tài liệu khác nhau thể hiện nên không được trình bày ở đây.



Hình 2.14. Sơ đồ nguyên lý modem QPSK – OFDM

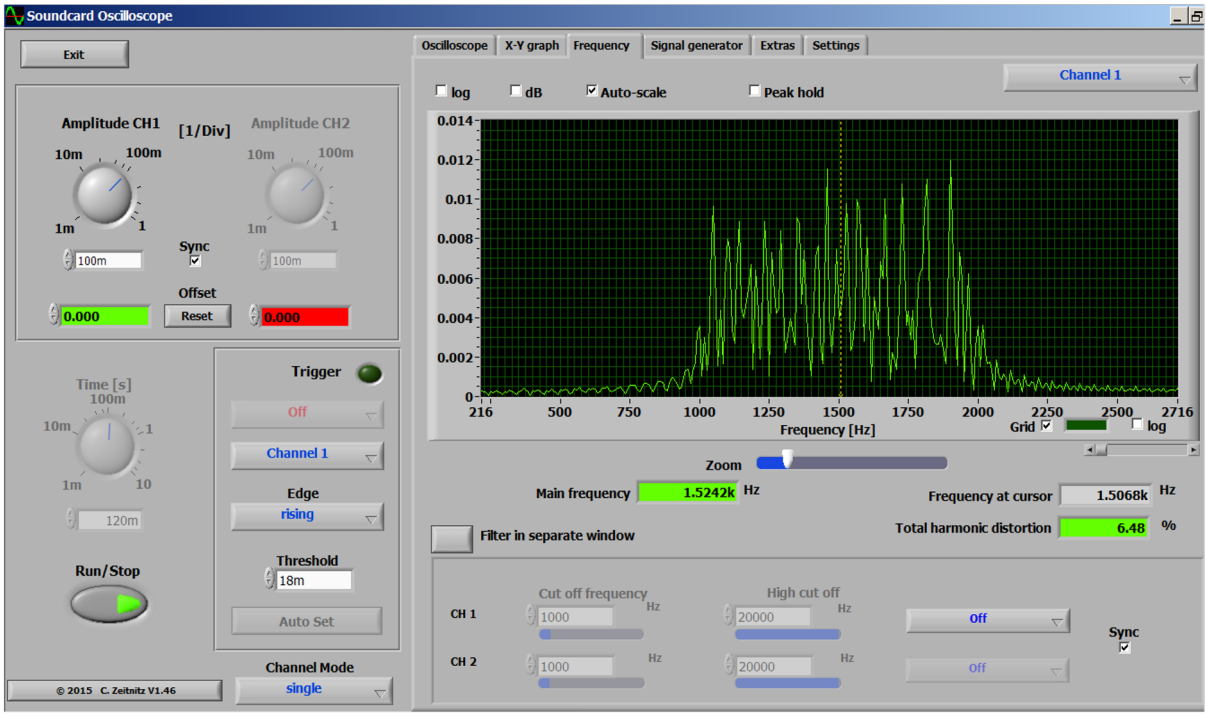
Về mặt lý thuyết mà nói thì phương pháp điều chế tín hiệu tựa tiếng nói sẽ cho kết quả tốt nhất (như Hình 2.16). Tuy nhiên trong thực tế rất khó thực hiện và có thực hiện được thì chất lượng cũng không cao như đã phân tích ở trên. *Thực nghiệm cho thấy điều chế tín hiệu OFDM có cấu trúc phổ gần giống phổ của tiếng nói có ưu điểm không bị VAD chặn, dễ thực hiện, kết quả khá tốt. Trường hợp kênh truyền có băng thông tối đa BER < 0.05%, trường hợp kênh truyền xấu BER không quá vài %.*



Hình 2.15. Tích hợp modem GSM vào phần cứng và phần mềm trên di động

Hướng tiếp theo là tích hợp toàn bộ modem này vào chip để có thể lắp vào điện thoại di động, không cần máy tính nữa (Hình 2.15). Đây là công việc gian nan và cần có thời gian. Lập trình trên chip với không gian chật hẹp, tài nguyên hạn chế nên yêu cầu phải tối ưu hóa về tốc độ, về kích thước mã chương trình, về không gian

vùng nhớ dữ liệu và vùng nhớ phục vụ thao tác tính toán. Hướng khác là tích hợp chức năng modem vào phần mềm của điện thoại di động thông minh. Công việc này cũng khó khăn không kém công việc tích hợp vào chip



Hình 2.16. Phổ tín hiệu thu được từ điều chế OFDM bằng QPSK

2.5. Kết luận chương 2

Chương này đã nêu ra một số phương pháp có thể mã hóa tín hiệu thoại dựa trên các đặc tính kênh để truyền qua kênh thoại GSM, như mã hóa xáo trộn phổ tín hiệu, can thiệp vào mã nguồn phần điều chế Modem GSM, sử dụng chế độ truyền dữ liệu trên băng tần GSM (kênh CSD) và đề xuất phương án nghiên cứu của Luận án để mã hóa và truyền dữ liệu cuộc gọi thoại mật qua kênh GSM. Cũng đã chỉ ra được kết quả của phương pháp này không chỉ để truyền dữ liệu qua kênh thoại GSM mà nó còn có thể được ứng dụng để truyền dữ liệu qua các kênh truyền thoại khác, như các kênh thoại vệ tinh, thoại VoIP, PSTN.

Chương này cũng đã làm rõ them về những trở ngại của đặc kỹ thuật của mạng GSM và kênh Voice GSM, đó là vấn đề tính thời gian thực, phương pháp nén tín hiệu thoại, điều chế và truyền trên bang tần hẹp, đặc tính cấu trúc khung truyền dữ liệu,

chức năng nhận diện tín hiệu thoại (VAD) và phân tích, lựa chọn thuật toán nén MELP, đề xuất và thực hiện *cải tiến thuật toán MELP thành MELPe* để phù hợp với các tính chất kênh truyền (*băng tần hẹp, trên kênh hay lỗi bít, mất gói, mất đồng bộ và phải hiệu quả trong việc cân đối giữa băng thông và chất lượng tín hiệu thoại, đặc biệt độ phức tạp tính toán có thể thực hiện trên các Chíp ARM hay DSP*) mà đề tài hướng đến.

Trong chương 2 trình bày một số phương pháp điều chế dữ liệu tựa ngẫu nhiên thành tín hiệu tựa tiếng nói mà một số nghiên cứu trên thế giới đã làm, từ đó đề xuất phương pháp điều chế, kỹ thuật lựa chọn các thông số và thực hiện điều chế OFDM với QPSK. *Đây là hướng nghiên cứu, mục tiêu chính và kết quả thực nghiệm đạt được của Luận án nghiên cứu*.

CHƯƠNG 3: BẢO MẬT DỮ LIỆU SỬ DỤNG THUẬT TOÁN SINH SỐ GIẢ NGẪU NHIÊN DỰA TRÊN DÃY PHI TUYẾN HAI CHIỀU LỒNG GHÉP

3.1. Giới thiệu m-dãy

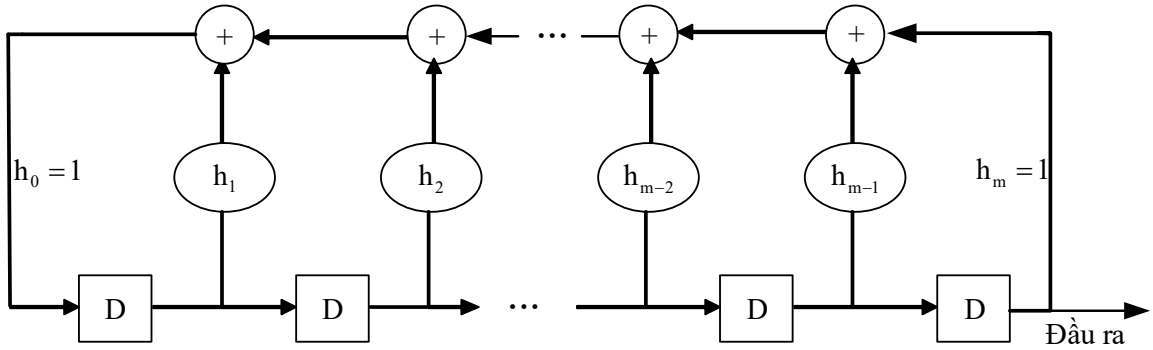
3.1.1. Thanh ghi dịch và đa thức nguyên thủy

Để dựng một m-dãy có độ dài $N = 2^m - 1$, ta biểu diễn đa thức nguyên thủy (prime polynomial) h(d) bậc m có dạng như sau:

$$h(d) = h_0 + h_1 d + h_2 d^2 + \ldots + h_{m-1} d^{m-1} + h_m d^m = \sum_{i=0}^{m} h_i d^i \qquad (3.1)$$

Trong đó $h_0 = h_m = 1$. Đa thức này được sử dụng để xây dựng thanh ghi dịch phản hồi tuyến tính (LFSR) như biểu diễn trong Hình 3.1, bao gồm m nút biểu diễn cho các phần tử nhớ hay các thành phần flip-flops (các phần tử tạo trễ 1 chu kì xung nhịp), mỗi phần tử nhớ có thể ghi nhớ giá trị '0' hoặc '1'. Tại mỗi thời điểm có sườn lên của xung nhịp Clock, giá trị trong các phần tử nhớ được dịch sang phần tử bên phải, đồng thời giá trị phản hồi được tính toán theo các hệ số của đa thức h(d) sau đó phản hồi đến phần tử đầu cùng bên trái. Phép tính phản hồi là phép nhân và cộng theo mô-đun 2, tương đương với phép tính AND và XOR trong mạch điện tử.

Từ đặc tính này, ta có thể thấy rằng một dãy có thể được tạo bởi một bộ LFSR 2 trạng thái, và có các hệ số phản hồi được kết nối đến đầu ra của phần tử nhớ thứ i nếu như $h_i = 1$ và không có phản hồi nếu $h_i = 0$.



Hình 3.1. Thanh ghi dịch phản hồi tương đương h(d)

Đa thức h(d) là đa thức nguyên thủy bậc m nếu số nguyên nhỏ nhất n, mà đối với số này $d^n + 1$ chia hết cho đa thức g(d) với $n = 2^m - 1$.

Ví dụ 3.1: $h(d) = d^5 + d^4 + d^3 + d + 1$ là một đa thức nguyên thủy bậc m = 5 vì số nguyên n nhỏ nhất mà $d^n + 1$ chia hết cho đa thức g(d) là $n = 2^5 - 1 = 31$. Trái lại, $h(d) = d^5 + d^4 + d^3 + d^2 + d + 1$ không phải là nguyên tố vì: $d^6 + 1 = (d + 1)(d^5 + d^4 + d^3 + d^2 + d + 1)$, nên số n nhỏ nhất bằng 6.

Số đa thức nguyên thủy bậc m bằng:

$$N_p = \frac{1}{m}\Phi(2^m - 1) \quad (3.2)$$

Trong đó Φ(n) là hàm Euler xác định bởi:

$$\Phi(n) = n.\prod_{p|n}\left(1 - \frac{1}{p}\right) \quad (3.3)$$

Ở đây p|n là ký hiệu "tất cả các ước số nguyên tố của n". Hàm Euler Φ(n) bằng số các số nguyên dương nhỏ hơn n và là các số nguyên tố cùng nhau so với n.

Ví dụ 3.2: $\Phi(15) = 15(1 - 1/3)(1 - 1/5) = 8$, ta thấy rằng {1, 2, 4, 7, 8, 11, 13, 14} là các số nguyên tố cùng nhau so với 15. Ngoài ra, $\Phi(31) = 30$, ta thấy rằng $\Phi(p) = p - 1$ cho mọi số nguyên tố $p \geq 1$ vì tất cả các số dương nhỏ hơn p đều là số nguyên tố cùng nhau so với p.

Bảng 3.1. Thống kê số lượng đa thức nguyên thủy có bậc m.

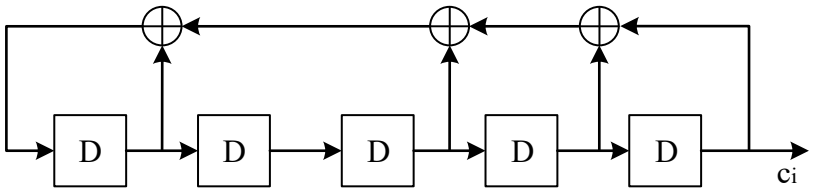
Bậc m	**Độ dài chuỗi $N = 2^m - 1$**	**Số đa thức**
3	7	2
4	15	2
5	31	6
6	63	6
7	127	18
8	255	16
9	511	48

Bậc m	Độ dài chuỗi $N = 2^m - 1$	Số đa thức
10	1023	60
11	2047	176
12	4095	144
13	8191	630
14	16383	756
15	32767	1800
16	65535	2048
17	131071	7710
18	262143	7776
19	524287	27594
20	1048575	24000

3.1.2. Dãy có độ dài cực đại

Trong hình 3.2, từ m nút chứa các giá trị 0 hoặc 1, ta có thể đưa ra 2^m trạng thái khác nhau cho thanh ghi dịch. Nhưng trạng thái gồm m giá trị 0 thì không thể xuất hiện (trong trường hợp này m-dãy sẽ sinh ra một dãy chứa toàn bit 0). Vậy *chu kì cực đại có thể là 2^m-1*.

Trong tài liệu [20] đã chứng minh rằng, nếu h(d) là một đa thức nguyên thủy bậc m, thanh ghi dịch sinh bởi đa thức h(d) sẽ sinh ra dãy đầu ra có chu kỳ kì 2^m-1. Ta gọi dãy đó là m-dãy.



Hình 3.2. Mạch thanh ghi dịch với hàm h(d) = d5 + d4 + d3 + d + 1

Hình 3.2 là mô hình mạch LFSR với hàm $h(d) = d^5 + d^4 + d^3 + d + 1$ là một đa thức nguyên thủy và nó tạo ra một m-dãy có chu kì $N = 2^5 - 1 = 31$ như biểu diễn trong bảng 3.1 với dãy đầu ra c_i = 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 1 1 0 1.

Bảng 3.2. Bảng các trạng thái thanh ghi dịch với hàm h(d) = d5 + d4 + d3 + d+1.

Xung nhịp i	Trạng thái	Xung nhịp i	Trạng thái
0	1 1 1 1 1	16	1 1 1 0 0
1	1 0 1 0 0	17	0 1 1 1 0
2	0 1 0 1 0	18	0 0 1 1 1
3	0 0 1 0 1	19	1 1 0 0 0
4	1 1 0 0 1	20	0 1 1 0 0
5	1 0 1 1 1	21	0 0 1 1 0
6	1 0 0 0 0	22	0 0 0 1 1
7	0 1 0 0 0	23	1 1 0 1 0
8	0 0 1 0 0	24	0 1 1 0 1
9	0 0 0 1 0	25	1 1 1 0 1
10	0 0 0 0 1	26	1 0 1 0 1
11	1 1 0 1 1	27	1 0 0 0 1
12	1 0 1 1 0	28	1 0 0 1 1
13	0 1 0 1 1	29	1 0 0 1 0
14	1 1 1 1 0	30	0 1 0 0 1
15	0 1 1 1 1	31	1 1 1 1 1

3.1.3. Các thuộc tính của m-dãy

Theo [*20*] một m-dãy $\{a_n\}$ với đa thức sinh h(d) có các thuộc tính:

1) Dãy có bậc m thì chu kì của dãy là $N = 2^m - 1$.

2) Có chính xác $N = 2^m - 1$ dãy không toàn '0' được tạo bởi h(d): chúng chỉ đơn giản là tập hợp N pha khác nhau (dịch vòng) của $a = \{a_n\}$ được gọi: a, Ta, T^2a,…, $T^{N-1}a$. Trong đó, T^m kí hiệu cho phép dịch là dịch một vector sang bên trái m vị trí. Nếu $a = \{a_n\} = (a_0, a_1, a_2, \dots, a_{N-1})$ thì $Ta = (a_1, a_2, a_3,\dots, a_{N-1}, a_0)$, $T^2a = (a_2, a_3,\dots, a_{N-2}, a_{N-1}, a_0, a_1)$, …, $T^Sa = (a_S, a_{S+1},\dots,a_{N-1}, a_0,\dots, a_{S-1})$.

3) Trong số N dãy được tạo bởi h(x) có chính xác một dãy $\{a_n\}$ có tính chất: $a_n = a_{2n}$ cho tất cả n = 0,1,2,…

4) Mỗi dãy không toàn '0', có chu kì $N = 2^m - 1$ thì bước chạy có độ dài m là sự xuất hiện của số ký tự '1' liên tiếp là một lần trong một chu kỳ của dãy, một bước chạy có độ dài m-1 cho '0', 2 bước chạy có độ dài m-1 cho '1', 2 bước chạy m-2 cho '0,…, 2^{m-r} bước chạy có độ dài r cho '1',…

5) Thuộc tính cộng và dịch, các trạng thái nó cho tất cả $0 \le r,s \le N-1$ có tồn tại một giá trị t sao cho: $T^ra + T^sa = T^ta$. Điều này chính là tổng của hai m-dãy là một m-dãy khác được tạo bởi cùng h(x). Một dãy có chu kì N là một m-dãy nếu và chỉ nếu nó có các thuộc tính cộng và dịch.

6) Số giá trị 1 trong mỗi chu kỳ là 2^{m-1}; số giá trị 0 là $2^{m-1} - 1$.

7) Hàm tự tương quan (ACF) có hai mức:

$$R(\tau) = \sum_{n=0}^{N-1}(-1)^{a_n + a_{n+\tau}} = \begin{cases} N, & \tau \equiv 0(\text{mod N}) \\ \sum_{n=0}^{N-1}(-1)^{a_t} = -1, & \tau \neq 0(\text{mod N}) \end{cases} \quad (3.4)$$

8) Nếu dãy nhị phân m được lấy mẫu với f bằng mũ 2, thì cùng dãy trả về.

9) Lấy mẫu một m-dãy với mỗi phép quay f, gcd(f,N) = 1, $1 \le f \le N-1$, $N = 2^m - 1$, sẽ đưa ra $\Phi(2^m-1)/m$ m-dãy có chu kì $2^m - 1$.

10) Khoảng tuyến tính bằng bậc m. Nó cho ta biết rằng có thể xác định được m-dãy đó nếu như xác định đúng 2m giá trị trong dãy.

3.2. Dãy có cấu trúc lồng ghép

3.2.1. Xây dựng dãy lồng ghép và dãy phi tuyến lồng ghép

Theo [21], ý tưởng cơ bản của kỹ thuật lồng ghép là dựa vào các m-dãy có độ dài có thể phân tích được thành tích và có ít nhất một nhân tử dạng $2^m - 1$. Thứ tự lồng ghép và các dãy con sẽ được xác định và quyết định cấu trúc của mã. Sau đó, chuyển đổi cấu trúc đó thành phi tuyến để tăng tổ hợp mã và độ phức tạp, có thể theo các phương pháp sau:

- Phương pháp 1: Giữ nguyên thứ tự lồng ghép nhưng thay m-dãy con thành phần bằng m-dãy khác có cùng độ dài.
- Phương pháp 2: Giữ nguyên thứ tự lồng ghép nhưng thay m-dãy con thành phần bằng dãy phân bố tựa ngẫu nhiên cùng độ dài.
- Phương pháp 3: Dùng dãy tích của T m-dãy con thành phần tạo dãy lớn.
- Phương pháp 4: Dùng dãy tích của T dãy con thành phần là các m-dãy khác nhau tạo dãy lớn.

Các phương pháp trên có thể được chia làm hai nhóm phương pháp chính là: nhóm thứ nhất là nhóm sử dụng cấu trúc lồng ghép đa cấp (thứ tự lồng ghép I_p^T) và lồng dãy con có đặc tính ngẫu nhiên được tạo từ m-dãy khác thay thế để tạo dãy phi tuyến (phương pháp 1 và 2) và nhóm thứ 2 là nhóm trực tiếp tạo dãy phi tuyến bằng cách tạo dãy tích T bậc và thực hiện nhiều cấp (phương pháp 3 và 4). Khi tạo dãy tích thông thường ta sẽ tạo được dãy có độ dài k.L (k nguyên dương tùy ý, $L = 2^m - 1$), còn khi tạo dãy lồng ghép đa cấp phi tuyến được trình bày trong luận án này có giá trị độ dài của dãy là $T.L = 2^n - 1$ (T là chu kì lồng ghép, L là độ dài m-dãy $L = 2^m - 1$).

Về mặt toán học, có thể biểu diễn và phân tích dãy có cấu trúc lồng ghép bằng hai công cụ trên trường hữu hạn là hàm Vết hoặc biến đổi D. Hai công cụ toán học này là tương đương và đều là cách biểu diễn các phần tử trong trường hữu hạn. Đánh giá về 2 phương pháp này như sau:

Phương pháp dùng hàm Vết là rất thích hợp cho việc nghiên cứu tạo và lồng ghép dãy m, tuy nhiên hàm Vết chỉ có thể thực hiện được với dãy có độ dài $L = p^m - 1$, nên nó không thể được dùng cho cấu trúc lồng ghép dãy có độ dài tùy ý ($L \neq p^m - 1$), hơn nữa nó không thể cho biết thông tin về trạng thái LFSR.

Phép biến đổi D là ngắn và dễ dàng thực hiện được, hơn nữa nó còn chứa đầy đủ thông tin về trạng thái LFSR. Phép biến đổi D có thể được áp dụng cho bất kì dãy tuần hoàn nào có độ dài có thể phân tích thành dạng L = T.N. Phép biến đổi D là phương pháp gần với phần cứng nhất, phương pháp này có thể được dùng để tính độ phức tạp (ELS) và hàm tương quan (ACF) của dãy. Do đó phép biến đổi D sẽ được để thực hiện phân tích, đánh giá và tạo dãy.

Kiến trúc dãy lồng ghép [1b]

Ta quan tâm tới m-dãy tam phân $\{b_n\}$ có độ dài $L = q^n$-l với q là một số nguyên tố nhận các giá trị trong tập {2, 3, 5, 7…} sao cho n=m.l.

Gọi

$N = p^m$-1

S = L/N

Trong bài báo [1b] đã chỉ ra rằng trong trường hợp này, ta có thể xây dựng lên dãy $\{b_n\}$ bằng cách lồng ghép (S-1) dãy con thành phần, mỗi dãy có độ dài N. Các dãy con có được bằng cách áp dụng phép nhảy bước (decimation) trên dãy $\{b_n\}$ với bước nhảy bằng S

Khi phép nhảy bước bắt đầu từ bit đầu tiên của $\{b_n\}$, ta thu được dãy con:

$$\{a_0, a_S, \ldots, a_{(3^m-2)S}\}$$

Tương tự như vậy, với vị trí bắt đầu nhảy bước là t, ta thu được dãy con

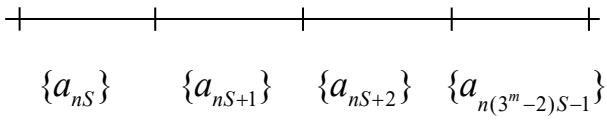
$$\{a_t, a_{S+t}, \ldots, a_{(3^m-2)S+t}\}$$

Do đó, xét trên miền thời gian, các dãy con này (sắp xếp theo cột) có thể được coi là ghép kênh theo bước thời gian S $\{a_{nS}\}\{a_{nS+1}\}\ldots\{a_{n(3^m-2)S-1}\}$ để đặt vào S khe thời gian như trong sơ đồ dưới đây

$$M = \begin{vmatrix} a_0 & a_1 & \dots & a_{S-1} \\ a_S & a_{S+1} & \dots & a_{2S-1} \\ \dots & \dots & \dots & \dots \\ a_{(3^m-2)S} & a_{(3^m-2)S+1} & \dots & a_{(3^m-1)S-1} \end{vmatrix}$$

$$= \{a_{nS}\}\{a_{nS+1}\}\dots\{a_{n(3^m-2)S-1}\}$$

T khe thời gian



***Sơ đồ 1**: Ghép các dãy con theo thời gian.*

Thứ tự mà các chuỗi con được ghép vào trong thực tế là thứ tự lồng ghép I_P^S đã xét ở trên.

Bây giờ ta chỉ cần tìm kiếm trong Bảng 3.4 để tìm ra biển diễn theo biến đổi d $S_i(d^S)$ của dãy con (theo cột) và từ đó có được I_P^S chính là thứ tự của $S_i(d^S)$ trong bảng 1. Riêng trường hợp dãy con chứa toàn giá trị con, ta coi thứ tự lồng ghép là ∞.

Trong bài báo [1b] cũng chỉ ra một tính chất quan trọng của dãy lồng ghép, đó là mỗi dãy con của dãy lồng ghép là một m-dãy với bậc m, nhưng lệch pha với nhau một khoảng xác định bằng phần tử tương ứng trong I_P^S. *Nhiệm vụ chính để xây dựng dãy lồng ghép là tìm cách tính toán trước các giá trị của* I_P^S, từ đó có thể xác định ngay được các dãy con để ghép thành dãy đầu ra

Xây dựng dãy lồng ghép phi tuyến

Có rất nhiều cách thức để tạo dãy phi tuyến, ở đây sẽ thực hiện theo cách thức lồng ghép phi tuyến dựa vào các giá trị pha tìm được và chỉ thay đổi dãy con để tạo dãy mới có khoảng tuyến tính lớn hơn. Để tạo dãy phi tuyến ta thực hiện theo những bước như sau, theo bài báo [1b]:

Ta sẽ sử dụng hai m-dãy đầu vào $\{a_n\}$ và $\{b_n\}$ với cùng bậc n và bộ tham số n, m, S giống nhau. Sử dụng kiến trúc lồng ghép, ta xây dựng nên các dãy lồng ghép với thứ tự lồng ghép I_P^S và I'^S_P tương ứng.

Trong bước này, bằng cách giữ nguyên các giá trị pha lồng ghép I_p^T và thay đổi các dãy con (dãy lồng ghép) bằng các dãy con tương ưng với dãy đầu vào thứ hai. Ở đây, thực hiện chọn m-dãy với các tính chất được tạo từ hai đa thức nguyên tố khác nhau để lồng ghép với các giá trị pha lồng ghép I_p^T khác nhau được tính từ các đa thức tạo m-dãy có bậc là bội của đa thức tạo dãy lồng ghép.

Trong phần 3.2.2.3 sau đây có các đánh giá chi tiết về độ phức tạp tuyến tính của một dãy lồng ghép phi tuyến cụ thể (sử dụng khoảng tương đương tuyến tính – ELS), từ đó kết luận rằng dãy lồng ghép phi tuyến có độ phức tạp cao hơn so với dãy lồng ghép thông thường

3.2.2. Các tính chất của dãy lồng ghép

3.2.2.1. Tính ngẫu nhiên

Bất kì dãy ngẫu nhiên nào được sử dụng trong các hệ thống hiện nay thì đều được tạo bởi máy tính hay một thiết bị nào đó. Tất nhiên, nó phải dựa vào một thuật toán nào đó để có thể tạo ra dãy giá trị đảm bảo được tính chất ngẫu nhiên. Vì vấn đề đó mà dãy này được gọi là dãy giả ngẫu nhiên.

Các phép đo cho dãy giả ngẫu nhiên 2 mức:

1) Thuộc tính cân bằng: xác suất xuất hiện bit ‘0’ và bit ‘1’ là gần bằng nhau (mỗi loại bằng 1/2). Trong thực tế, số lượng bit ‘0’ và ‘1’ sai khác nhau là một đơn vị, sở dĩ là như vậy là do bộ tạo dãy loại trừ trạng thái tất cả các bit bằng ‘0’, và sự sai khác nhau 1 bit này không có ý nghĩa khi dãy cực dài.

2) Thuộc tính chạy: sự kéo dài của bit ‘0’ hoặc bit ‘1’ liên tiếp trong dãy được định nghĩa là một bước chạy. Một bước chạy được định nghĩa là một nhóm bit cùng loại (‘1’ hoặc ‘0’) liên tiếp tồn tại trong dãy. Trường hợp có một bit ‘0’ hay bit ‘1’ xen giữa các bit ‘1’ hay ‘0’ liên tiếp cũng được xem là kết thúc một bước chạy và bắt đầu một bước chạy mới. Độ dài bước chạy là số bit trong mỗi bước chạy. Đặc tính chạy trong dãy giả ngẫu nhiên phải thỏa mãn là trong một chu kỳ của dãy số tổng quát có: $1/2^n$ số bước chạy có độ dài là n.

3) Thuộc tính tự tương quan: Từ một dãy giả ngẫu nhiên đã có, nếu ta dịch chuyển theo cách dịch đi lần lượt từng vị trí bit sang phải hoặc sang trái, ta sẽ thu được m-dãy mới có số phần tử trùng hợp và không trùng hợp vơi dãy ban đầu. Hàm tự tương quan của dãy giả ngẫu nhiên có dạng gần như là 2 mức, với giá trị đỉnh tại trễ là 0 là đỉnh và tại các vị trí khác là rất thấp. Đây là thước đo quan trọng nhất để đánh giá độ giống nhau của dãy với các bước dịch của nó, với hàm tự tương quan được xác định như sau:

$$R(\tau) = \sum_{n=0}^{N-1} \hat{a}_n \hat{a}_{n+\tau} = \begin{cases} N, & \tau \equiv 0 \\ c, & \tau \neq 0 \end{cases} \quad (3.5)$$

Trong đó $\hat{a}_n = (-1)^{a_n} \in \{+1, -1\}$, $a_n \in \{1,0\}$, c là một giá trị nhỏ.

3.2.2.2. Hàm tự tương quan

Để đánh giá đặc tính tự tương quan của dãy phi tuyến ta dựa vào dãy lồng ghép pha I_p^T của dãy phi tuyến được cho trong [*22*]. Trong thực tế, việc tính ACF của một dãy lồng ghép phụ thuộc vào tính duy nhất trong các vị trí giống nhau giữa các giá trị lồng ghép I_p^T, và nó được định nghĩa như sau:

$$r_{K+T} = r_K (mod\ N)$$

trong đó, r_k là các giá trị của $[I_p^T]$ và $\infty + T = \infty$.

Bằng việc kết hợp với I_p^T và tăng dãy đó cùng với cấu trúc ma trận TxT.N, ta có thể tìm ra tất cả các vị trí trùng khớp. Trong các đường chéo của ma trận TxT.N chỉ thông tin về trùng khớp là được đưa ra.

Gọi i, j là các chỉ số hàng, cột của ma trận. Ta có các chỉ số trong ma trận TxT.N:

$$S(i,j) = \begin{cases} 0, (I_p^{T(1)}(j) - I_p^T(i))\ mod(N) \neq 0 \\ 1, (I_p^{T(1)}(j) - I_p^T(i))\ mod(N) = 0 \end{cases} \quad (3.6)$$

Trong đó:

1) N là độ dài dãy con, $N = 2^n - 1$.
2) $j = d.T + e$, d và e là số nguyên bất kì.
3) $0 \leq i < T$, $0 \leq j < T.N$.

4) $I_p^{T(1)}(j) = \left(I_p^T(j \bmod T) + d\right) mod(N)$.

Ta có công thức tính số vị trí trùng khớp trên đường chéo k của ma trận TxT.N như sau:

$$C(k) = \sum_{i=0}^{T-1} S(i,j), \quad \text{j=(i+k)mod(T.N)} \tag{3.7}$$

Với dãy đảm bảo các thuộc tính ngẫu nhiên thì đường chéo đầu tiên đưa ra một số lượng lớn các giá trị tương quan θ(j) trong tất cả các vị trí, đường chéo thứ k với k ≠ 0 chứa chính xác (T – P) vị trí trùng khớp cho m-dãy có độ dài $L = 2^m - 1$ [8], m chia hết cho n và:

$$P = \frac{L+1}{N+1} = \frac{2^m}{2^n}$$

3.2.2.3. Độ phức tạp

Độ phức tạp tuyến tính là thông số để đánh giá độ phức tạp của dãy. Ở đây sử dụng thước đo khoảng tuyến tính tương đương (ELS) được đưa ra trong [23] để đánh giá độ phức tạp của dãy phi tuyến được tạo bằng phương thức lồng ghép.

Khoảng tuyến tính tương đương là độ dài của dãy LFSR ngắn nhất để tạo ra dãy đó (hay chính là bậc của đa thức). Giá trị ELS càng lớn thì số lượng bit cần phải xác định đúng để có thể dự đoán được dãy là càng lớn.

Gọi S là số các dãy con được chọn cho việc lồng ghép, với mỗi dãy con có độ dài N. Việc tính toàn giá trị ELS có thể chia làm hai bước như sau [23]:

Bước 1: các dãy con $f_i(d)$ được trải ra S lần (chèn (S-1) giá trị 0 vào giữa hai giá trị liên tiếp của $f_i(d)$).

$$f_i(d^S) = \frac{s_i(d^S)}{h_1(d^S)} \qquad \text{i=1,2,...,S} \tag{3.8}$$

$f_i(d^S)$ là biểu diễn dãy con do đa thức $h_1(d^S)$ tạo thành.

Bước 2: Chèn ghép S pha khác nhau ($f_i(d^S)$) của dãy con có độ dài N để tạo dãy có độ dài L=S.N:

$$u(d) = \sum_{i=1}^{S} d^i \frac{s_i(d^S)}{h_1(d^S)} = \frac{G(d)}{h_1(d^S)} \quad (3.9)$$

ELS của dãy lồng ghép u(d) có thể được tính như sau. Gọi K(d) là ước chung lớn nhất của G(d) và $h_1(d^S)$, ta có:

$$K(d) = \gcd\{G(d), h_1(d^S)\} \quad (3.10)$$

Sau đó ta có:

$$\frac{G(d)}{h_1(d)} = \frac{K(d)G'(d)}{K(d)h_1'(d)} = \frac{G'(d)}{h_1'(d)} \quad (3.11)$$

Trong đó, G'(d) và $h_1'(d)$ có liên hệ nguyên tố với nhau. Việc tính ELS đơn giản là việc ước lượng bậc của đa thức $h_1'(d)$, ta có:

$$ELS = deg[h_1'(d)] = deg[h_1(d^S)] - deg[K(d)] \quad (3.12)$$

Trong đó, deg[*f*(d)] là kí hiệu bậc của đa thức *f*(d).

3.2.3. Các phương pháp sinh dãy lồng ghép và lồng ghép phi tuyến

Trong bài báo [1b] đã giới thiệu 3 phương pháp sinh dãy lồng ghép, trong đó phương pháp sinh dãy lồng ghép sử dụng biến đổi d và phương pháp sinh dãy lồng ghép sử dụng hàm vết là kế thừa các kết quả nghiên cứu trước đó (Bài báo IEEE 1985 – Le Minh Hieu]). *Phương pháp thứ ba tính toán trực tiếp giá trị* I_P^S *là một phương pháp mới, được tác giả luận án đề xuất trong công bố này*.

3.2.3.1. Phương pháp sinh dãy lồng ghép sử dụng biến đổi d

Cho $\{b_n\}$ là một m-dãy sinh bởi đa thức sinh g(d) có bậc n, chu kỳ L thỏa mãn các điều kiện:

$$L = 3^n - 1 = 3^{l*m} - 1 = S.(3^m - 1) = S.N,\ n = l.m,\ S = (3^n - 1)/(3^m - 1)$$

Gọi b(d) là biến đổi – d của $\{b_n\}$ theo công thức

$$b(d) = \frac{S(d)}{g(d)} \quad (3.13)$$

với S(d) là trạng thái khởi đầu của m-dãy

Ta luôn có thể biểu diễn b(d) theo dạng

$$b(d) = \sum_{i=0}^{S-1} d_i F_i(d^S) \tag{3.14}$$

với Fi(d) là dãy con được sinh từ đa thức sinh $g_1(d)$ có bậc m, chu kỳ N và được xác định trong biến đổi d theo công thức

$$F_i(d) = \frac{S_i(d)}{g_l(d)},\ i = 0, 1, \ldots, S\text{-}1 \tag{3.15}$$

Với $S_i(d)$ là trạng thái khởi đầu của dãy con và $g_1(d)$ là đa thức sinh tương ứng của dãy đó.

Điều này được suy ra trực tiếp từ các thuộc tính của biến đổi d vì $\{b_n\}$ có thể được xây dựng bằng cách lồng ghép các pha S của $\{F_n\}$. Các pha cụ thể của $\{F_n\}$ trong đó thứ tự lồng ghép có thể được xác định thông qua 3 bước.

Bước 1: mở rộng F_i sau (d) bằng S lần (chèn S-1 số 0 vào giữa hai bit liên tiếp của F_i (d)), trong biến đổi d, nó tương đương với việc thay thế d bằng d^S trong công thức

$$F_i(d^S) = \frac{S_i(d^S)}{g_l(d^S)} \tag{3.16}$$

Bước 2: Biểu diễn theo biến đổi d của $\{b_n\}$ theo cách để xen kẽ các $F_i(d)$, (hoặc chèn S pha khác nhau của Fi(d) để tạo thành b(d))

$$b(d) = \sum_{i=0}^{S-1} d_i F_i(d^S) = \sum_{i=0}^{S-1} d^i \frac{S_i(d^S)}{g_l(d^S)} \tag{3.17}$$

Sau đó đặt phần tử số của (3.17) thành

$$G(d) = \sum_{i=0}^{S-1} d^i S_i(d^S) \tag{3.18}$$

Từ (3.17), (3.18) vào (3.13) ta có

$$G(d) = \frac{S(d) \cdot g_1(d^S)}{g(d\ )} \tag{3.19}$$

Bước 3: - đặt $d^S = D$

- Tìm các bước dịch pha $\frac{S_i(D)}{g_l(D)} = F_i(D)$

Sau đó nhóm lại biến đổi d của b(d) thành:

$$b(d) = \sum_{i=0}^{S-1} d^i F_i(D) \quad (3.20)$$

So sánh phần $F_i(D)$ với bảng biến đổi d của từng phần của dãy tương ứng với cột sau của **Bảng 3.4**, ta có thể xây dựng lên toàn bộ các bậc lồng ghép I_P^S.

3.2.3.2. Phương pháp sinh dãy lồng ghép sử dụng hàm vết

Trong tài liệu [15 - R. LIDL & H. Niedemeiter, Introduction to finite field and their application, Cambridge University press 2000] mối quan hệ giữa lũy thừa của phần tử sinh α của dãy và biến đổi d đã được giải thích rất rõ ràng.

Vì cả hai biểu diễn (thông qua giữa lũy thừa của phần tử sinh α và biến đổi d) khá tương đương, nên bậc lồng ghép (vị trí mà các dãy con thành phần sẽ được sắp xếp) có thể được xác định bằng phương pháp biểu diễn theo lũy thừa của α (còn gọi là hàm vết – Trace function). Khi hợp này, bậc lồng ghép được xác định như sau:

Gọi m, n là hai số nguyên dương và m là ước số của n, $S = \frac{2^n-1}{2m-1}$ và α là phần tử nguyên tố thuộc trường hữu hạn $GF(2^n)$.

Hàm Vết của α là ánh xạ của $GF(2^n)$ xuống $GF(2^m)$:

$$Tr_m^n(\alpha) = \sum_{k=0}^{\frac{n}{m}-1} \alpha^{2^{mk}} \quad (3.21)$$

Thứ tự lồng ghép I_p^T được định nghĩa theo hàm Vết được tính như sau:

$$I_p^j = \begin{cases} i, & Tr_m^n(\alpha^j) = \alpha^{T.i} \\ \infty, & Tr_m^n(\alpha^j) = 0 \end{cases} \quad (3.22)$$

Trong đó, i = 0, 1, 2,…, $2^m - 2$ và j = 0,1,2,…,S – 1.

3.2.3.3. Phương pháp tính toán trực tiếp giá trị thứ tự lồng ghép

Thực tế khi lập chương trình trên máy vi tính để cài đặt hai phương pháp phân rã m-dãy nêu trên, việc thực hiện cả hai phương pháp trên dường như không hiệu quả, đặc biệt là khi độ dài dãy tăng lên đáng kể. Ta chỉ có thể lập được bảng 1 nếu tổng kích thước của bảng có thể lưu hiệu quả trong bô nhớ máy tính hoặc thiết bị tính toán. Đồng thời ta cần phải thực hiện đầy đủ S bước để xây dựng bậc lồng ghép gồm S phần tử. Vì thế tác giả luận án sẽ giới thiệu một phương pháp hiệu quả hơn để tìm ra những phần tử đầu tiên của bậc lồng ghép.

Trước hết ta sinh ra phần đầu của chuỗi $\{b_n\}$ từ trạng thái ban đầu được cho trước, nhưng thay vì tạo chuỗi toàn chu kỳ (p^n-1 phần tử), ta chỉ cần tạo ra m.S giá trị đầu tiên.

Tiếp đó ta sẽ sắp xếp lại các giá trị này bằng cấu trúc lồng ghép theo định nghĩa của dãy lồng ghép, từ đó ta có thể nhận được trực tiếp các trạng thái ban đầu của dãy con $F_i(d)$.

Từ vị trí của trạng thái này (liên quan đến thứ tự xen kẽ), Ta chỉ cần sao chép N giá trị liên tiếp của $F_i(d)$ thành chuỗi đầu ra của dãy lồng ghép. Nếu dãy con $F_i(d)$ có kích thước quá lớn, ta có thể xây dựng lên dãy con này từ trạng thái ban đầu vừa được chỉ ra mà không cần quan tâm tới giá trị vị trí trong tập thứ tự lồng ghép.

Lợi thế của phương pháp tính trực tiếp thứ tự lồng ghép

Phương pháp tính trực tiếp thứ tự lồng ghép không yêu cầu các tính toán đa thức trên trường hữu hạn như hai phương pháp trước đó, chỉ cần sử dụng phương pháp sinh

m-dãy song cũng tạo ra kết quả tương ứng. Trong trường hợp dãy con $F_i(d)$ có kích thước rất lớn, phương pháp tính trực tiếp thứ tự lồng ghép cũng cho phép sinh ra dãy lồng ghép mà không cần xây dựng toàn bộ dãy con $F_i(d)$.

Hiệu quả của phương pháp tính trực tiếp thứ tự lồng ghép: nếu sử dụng phương pháp d-Transform (phương pháp 1) ta cần tính toán toàn bộ chu kỳ của dãy ban đầu với số lượng 2^m-1 phần tử để lập bảng 3.1; nếu sử dụng phương pháp tính trực tiếp ta

chỉ cần tính toán cho m.S giá trị đầu tiên của dãy ban đầu. Với các tham số cụ thể, hiệu quả đạt được như trong **Bảng 3.3**

Lượng bộ nhớ cần thiết cho phương pháp tính trực tiếp thứ tự lồng ghép là m.S ô nhớ. Lượng bộ nhớ này thường nhỏ hơn so với việc xây dựng toàn bộ dãy con $F_i(d)$.

STT	n	m	N	S	m.S	Tỷ lệ rút gọn
1	18	6	262143	4161	24966	9.52%
2	18	9	262143	513	4617	1.76%
3	24	8	16777215	65793	526344	3.14%
4	26	13	67108863	8193	106509	0.16%

Bảng 3.3. Bảng tính hiệu quả cải tiến số phép tính

Thực nghiêm đánh giá các dãy lồng ghép cụ thể

Tác giả đã sử dụng công cụ Matlab để mô phỏng, tính toán và thực hiện thuật toán tạo m-dãy và dãy lồng ghép:

Luận án sẽ thực hiện tạo dãy phi tuyến có độ dài $(2^{18} – 1) = 262143$ bit. Để tạo dãy phi tuyến có độ dài 262143 bit, ta thực hiện lồng ghép các m-dãy con với các đa thức nguyên thủy thuộc trường hữu hạn $GF(2^9)$ có độ dài dãy tương ứng là 511 bit theo các pha lồng ghép I_p^T được tạo dựa trên hàm Vết với các cặp phép ánh xạ $\{GF(2^{18}) \rightarrow GF(2^9)\}$. Trong bảng 3.3, giá trị các đa thức nguyên thủy được trình bày theo các giá trị hệ octal, ví dụ: $(736)_8 = (111011110)_2 \Leftrightarrow g(d) = 1 + d + d^2 + d^4 + d^5 + d^6 + d^7$.

Việc tính toán mô phỏng để thực hiện lồng ghép dãy chỉ dừng lại ở thực hiện lồng ghép 48 m-dãy được tạo bởi đa thức thuộc $GF(2^9)$ để tạo các dãy có độ dài $2^{18} – 1$ bit. ta có thể chọn một cặp đa thức để tạo dãy được cho trong **Bảng 3.3**.

Dựa vào thuật toán tính ELS đã trình bày và chạy thuật toán bằng Matlab ta có thể tính được các thông số ELS khi thực hiện lồng ghép, các kết quả tính toán này như sau:

Bảng 3.4. Các cặp đa thức lồng ghép tạo dãy mới.

STT	Đa thức thuộc GF(2^{18}) $g_0 \rightarrow g_{18}$	Đa thức thuộc GF(2^9) $g_0 \rightarrow g_9$
1	1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1	1 1 1 1 0 0 1 0 1 1
2	1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1	1 1 0 1 0 0 1 1 1 1
3	1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1	1 1 1 0 1 1 1 0 0 1
4	1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1	1 0 0 1 1 1 0 1 1 1
5	1 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1	1 0 1 0 0 1 0 1 0 1
6	1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1	1 0 1 0 1 0 0 1 0 1
7	1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1	1 1 0 1 1 0 0 0 0 1
8	1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1	1 0 0 0 0 1 1 0 1 1
9	1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1	1 1 1 0 0 0 0 1 0 1
10	1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1	1 0 1 0 0 0 0 1 1 1
11	1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 0 1 1
12	1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 1 1	1 1 0 1 1 1 1 1 1 1
13	1 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1	1 1 1 1 1 1 1 0 1 1
14	1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1	1 1 0 1 1 1 1 1 1 1
15	1 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1	1 1 0 0 0 1 1 1 1 1
16	1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1	1 1 1 1 1 0 0 0 1 1
17	1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1	1 1 1 0 1 1 1 0 0 1
18	1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 1	1 0 0 1 1 1 0 1 1 1
19	1 1 1 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 1	1 1 0 1 1 1 0 0 1 1
20	1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 1 1	1 1 0 0 1 1 1 0 1 1
21	1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1	1 1 1 1 0 1 1 0 0 1
22	1 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1	1 0 0 1 1 0 1 1 1 1

STT	Đa thức thuộc GF(2^{18}) $g_0 \rightarrow g_{18}$	Đa thức thuộc GF(2^9) $g_0 \rightarrow g_9$
23	1 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1	1 0 1 0 1 0 1 1 1 1
24	1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1	1 1 1 1 0 1 0 1 0 1
25	1 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1	1 1 0 1 1 0 0 0 0 1
26	1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 0 1	1 0 0 0 0 1 1 0 1 1
27	1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 1	1 0 1 0 1 0 0 1 0 1
28	1 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 1	1 0 1 0 0 1 0 1 0 1
29	1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1	1 1 1 1 0 0 1 0 1 1
30	1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 1 1	1 1 0 1 0 0 1 1 1 1
31	1 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1	1 1 1 1 0 0 1 0 1 1
32	1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 1	1 1 0 1 0 0 1 1 1 1
33	1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 1	1 1 0 0 0 1 1 1 1 1
34	1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 1	1 1 1 1 1 0 0 0 1 1
35	1 0 1 0 1 1 1 0 0 1 0 0 0 0 0 0 0 0 1	1 1 1 1 1 0 1 0 0 1
36	1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 1 0 1	1 0 0 1 0 1 1 1 1 1
37	1 1 0 1 0 1 0 1 0 1 0 0 0 0 0 0 0 0 1	1 1 1 0 1 1 0 1 0 1
38	1 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 1 1	1 0 1 0 1 1 0 1 1 1
39	1 0 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 0 1	1 0 0 1 1 1 1 1 0 1
40	1 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 1	1 0 1 1 1 1 1 0 0 1
41	1 1 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 1	1 0 0 1 0 1 1 1 1 1
42	1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1	1 1 1 1 1 0 1 0 0 1
43	1 1 0 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1	1 0 0 0 0 1 1 0 1 1
44	1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 0 1 1	1 1 0 1 1 0 0 0 0 1
45	1 0 1 1 0 1 0 0 1 1 0 0 0 0 0 0 0 0 1	1 0 0 1 1 1 0 1 1 1

STT	Đa thức thuộc GF(2^{18}) $g_0 \rightarrow g_{18}$	Đa thức thuộc GF(2^9) $g_0 \rightarrow g_9$
46	1 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 0 1	1 1 1 0 1 1 1 0 0 1
47	1 1 1 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1	1 1 0 0 0 1 1 1 1 1
48	1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1 1 1	1 1 1 1 1 0 0 0 1 1
49	1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 1	1 0 1 0 1 0 0 1 0 1
50	1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1	1 0 1 0 0 1 0 1 0 1
51	1 0 1 0 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1	1 1 1 0 1 1 1 0 0 1
52	1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 1	1 0 0 1 1 1 0 1 1 1
53	1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 1	1 0 0 1 0 1 1 0 0 1
54	1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1	1 0 0 1 1 0 1 0 0 1
55	1 0 0 1 0 1 0 1 1 1 0 0 0 0 0 0 0 0 1	1 1 0 1 1 1 1 1 1 1
56	1 0 0 0 0 0 0 0 0 1 1 1 0 1 0 1 0 0 1	1 1 1 1 1 1 1 0 1 1
57	1 0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1	1 0 1 0 1 0 0 1 0 1
58	1 0 0 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 1	1 0 1 0 0 1 0 1 0 1
59	1 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1	1 1 0 1 0 1 1 0 1 1
60	1 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 0 1	1 1 0 1 1 0 1 0 1 1
61	1 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 1	1 1 1 0 1 1 1 0 0 1
62	1 0 0 0 0 0 0 0 0 1 1 1 1 0 1 0 1 1 1	1 0 0 1 1 1 0 1 1 1
63	1 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1	1 0 1 1 1 1 0 1 0 1
64	1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1	1 0 1 0 1 1 1 1 0 1
65	1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 1	1 1 1 1 0 0 0 1 1 1
66	1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 1 0 1	1 1 1 0 0 0 1 1 1 1
67	1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1	1 0 0 1 1 0 1 0 0 1
68	1 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1	1 0 0 1 0 1 1 0 0 1

STT	Đa thức thuộc GF(2^{18}) $g_0 \rightarrow g_{18}$	Đa thức thuộc GF(2^9) $g_0 \rightarrow g_9$
69	1 0 1 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 1	1 0 1 0 0 1 0 1 0 1
70	1 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 0 1	1 0 1 0 1 0 0 1 0 1
71	1 1 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1	1 1 1 0 0 0 0 1 0 1
72	1 0 0 0 0 0 0 0 1 0 0 0 1 0 1 1 0 1 1	1 0 1 0 0 0 0 1 1 1
73	1 0 0 1 1 1 1 0 0 0 1 0 0 0 0 0 0 0 1	1 0 1 1 1 1 0 1 0 1
74	1 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1	1 0 1 0 1 1 1 1 0 1
75	1 0 0 1 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1	1 1 1 0 0 0 1 1 1 1
76	1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 1	1 1 1 1 0 0 0 1 1 1
77	1 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 0 1	1 1 1 0 1 1 1 0 0 1
78	1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 1	1 0 0 1 1 1 0 1 1 1
79	1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 1	1 0 0 1 1 0 1 1 1 1
80	1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 1	1 1 1 1 0 1 1 0 0 1
81	1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1	1 0 1 0 1 0 0 0 1 1
82	1 0 0 0 0 0 0 0 1 0 1 1 0 0 1 0 0 1 1	1 1 0 0 0 1 0 1 0 1
83	1 1 1 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1	1 0 0 0 0 1 0 0 0 1
84	1 0 0 0 0 0 0 0 1 0 1 1 0 0 1 1 1 1 1	1 0 0 0 1 0 0 0 0 1
85	1 0 1 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 1	1 1 0 0 0 1 0 1 0 1
86	1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 1 0 1	1 0 1 0 1 0 0 0 1 1
87	1 1 0 1 1 0 1 1 1 0 1 0 0 0 0 0 0 0 1	1 0 1 0 0 1 0 1 0 1
88	1 0 0 0 0 0 0 0 1 0 1 1 1 0 1 1 0 1 1	1 0 1 0 1 0 0 1 0 1
89	1 0 0 0 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1	1 1 0 0 1 0 0 0 1 1
90	1 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1	1 1 0 0 0 1 0 0 1 1
91	1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 0 0 1	1 0 1 1 0 1 0 0 0 1

STT	Đa thức thuộc GF(2^{18}) $g_0 \rightarrow g_{18}$	Đa thức thuộc GF(2^9) $g_0 \rightarrow g_9$
92	1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1	1 0 0 0 1 0 1 1 0 1
93	1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 1	1 1 0 1 1 1 0 0 1 1
94	1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 1	1 1 0 0 1 1 1 0 1 1
95	1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 1	1 1 1 1 0 0 1 0 1 1
96	1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 0 1	1 1 0 1 0 0 1 1 1 1
97	1 1 0 1 0 1 0 0 0 1 1 0 0 0 0 0 0 0 1	1 0 1 1 1 1 0 1 0 1
98	1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 1 1	1 0 1 0 1 1 1 1 0 1
99	1 0 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1	1 1 0 1 1 1 1 1 1 1
100	1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 1 0 1	1 1 1 1 1 1 1 0 1 1

Các m-dãy được tạo bởi các cặp đa thức trong **Bảng 3.4** sau khi thực hiện lồng ghép sẽ tạo ra các dãy phi tuyến mới có các tính chất (phân bố, ACF, ELS) tương tự nhau. Tuy nhiên, ta cũng có thể nhận thấy được rằng đa thức ngược của các đa thức con trên nếu được sử dụng lồng ghép để tạo dãy phi tuyến thì sẽ tạo ra dãy mới có ELS = 18 (tuyến tính).

Để đơn giản trong việc mô tả phương pháp tạo dãy phi tuyến bằng việc sử dụng phương pháp lồng ghép phi tuyến, ta sẽ chọn một cặp đa thức để mô tả và tính toán như sau:

1) Đa thức trên trường GF(2^{18}):

$g(d) = 1 + d^8 + d^9 + d^{12} + d^{13} + d^{14} + d^{15} + d^{16} + d^{18}$.

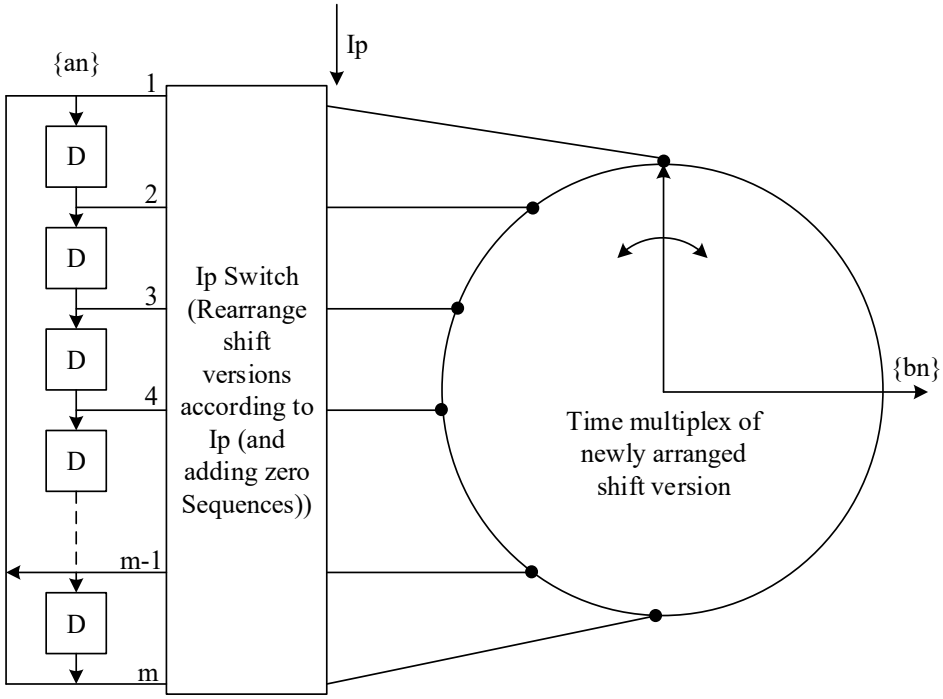
2) Đa thức trên trường GF(2^9):

$g(d) = 1 + d + d^2 + d^3 + d^4 + d^5 + d^6 + d^8 + d^9$. (3.23)

3.3. Thực thi dãy lồng ghép bằng phần cứng Vi xử lý

Bằng việc thực hiện lồng ghép các m-dãy con có độ dài $N = 2^m - 1$ theo các giá trị dãy pha lồng ghép có độ dài T ta tạo được dãy mới là phi tuyến có độ dài $L = N.T = 2^n - 1$ (trong đó, n chia hết cho m).

Mô hình cấu trúc lồng ghép phi tuyến được đưa ra trong [1b] :



Hình 3.3. Lồng ghép các thanh ghi dịch

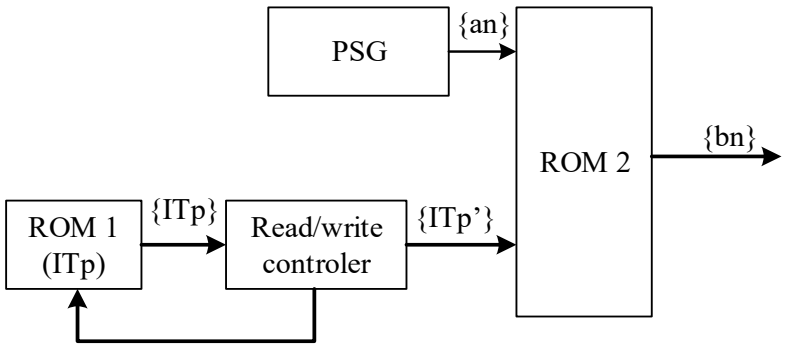
Trong phần này sẽ thực hiện phần cứng việc tạo dãy phi tuyến theo phương pháp lồng ghép với các giá trị đã biết:

1) Độ dài dãy con bậc m: $N = 2^m - 1$.

2) Độ dài dãy lớn: $L = 2^n - 1$.

3) Các giá trị dãy pha lồng ghép $\{I_p^T\}$có độ dài $T = \frac{L}{N}$.

Mô hình lồng ghép được biểu diễn trong Hình 3.3. Với sơ đồ lồng ghép này, việc thực hiện tạo dãy phi tuyến có thể được thực hiện trên các phần cứng các chip xử lý,…

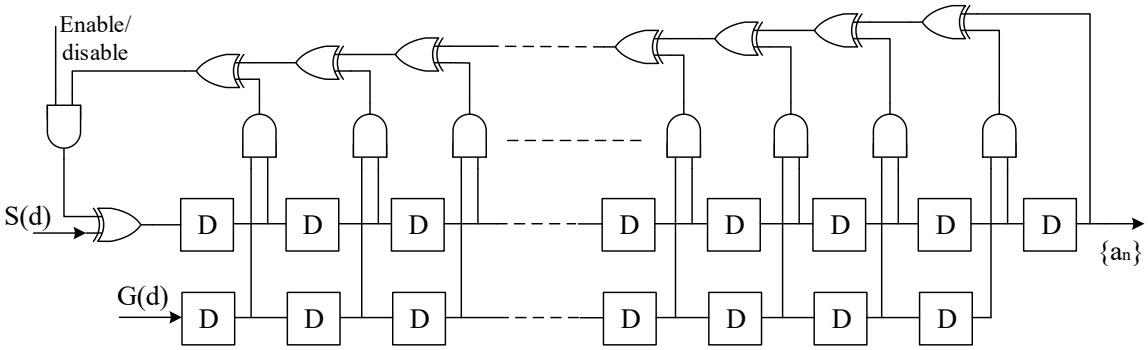


Hình 3.4. Sơ đồ khối phần cứng tạo dãy lồng ghép phi tuyến

Trong đó:

1) PSG

Là khối tạo m-dãy tuyến tính được tạo bởi đa thức nguyên tố. M-dãy được tạo bằng phần cứng sử dụng mô hình LFSR tái cấu hình được mô tả như trong Hình 3.45.



Hình 3.5. LFSR tái cấu hình

Ta thấy, trong mô hình LFSR thì các tham số: đa thức khởi tạo bộ LFSR G(d) và trạng thái kích hoạt phản hồi trên các thanh ghi dịch S(d) là hai tham số chính quyết định đến dãy đầu ra được cho bởi công thức 3.13 theo biến đổi D. Hình 3.5 là mô hình LFSR tái cấu hình sử dụng các phần tử nhớ là các thanh ghi dịch, các cổng logic AND và XOR. Với sơ đồ này, ta có thể thay đổi các tham số S(d) và G(d) để có thể tạo m-dãy đầu ra tương ứng.

Giả sử, muốn thay đổi trạng thái phản hồi (thay đổi đa thức tạo dãy) để tạo m-dãy dùng đa thức bậc 9 $g(d) = 1 + d + d^2 + d^3 + d^4 + d^5 + d^6 + d^8 + d^9$, thì bộ LFSR theo hình 3.5 sẽ có 17 bộ ghi dịch (9 bộ cho S(d), 8 bộ cho G(d)) với dãy đầu vào G(d) là 11111101 có độ dài 8, và để thay đổi trạng thái kích hoạt của LFSR là $S(d) = 1 + d^2$

+ d^8 thì ta phải kích hoạt chân Enable/Disable ở trạng thái '0' và dãy đầu vào là 101000001 có độ dài 9.

Với mô hình này, độ linh hoạt khối tạo dãy PN được tăng cao khi mà ta có thể thay đổi được đa thức tạo m-dãy tức là có thể thay đổi được dãy đầu ra $\{a_n\}$ tùy ý.

2) ROM 1

Đây là nơi chứa các giá trị I_p^T được sử dụng để xác định pha lồng ghép dãy $\{a_n\}$ được tạo bởi khối PSG. Đầu ra của khối vày sẽ là các giá trị I_p^T tương ứng được điều khiển bởi khối "Điều khiển đọc/ghi".

3) Điều khiển đọc/ghi

Khối này sẽ điều khiển việc đọc/ghi cho ROM 2. Khi ở chế độ ghi, khối này sẽ điều khiển ROM 2 ghi dữ liệu đầu vào $\{a_n\}$ vào bộ nhớ với chu kì bằng chu kì của dãy đầu vào N giá trị. Khi ở chế độ đọc, khối này sẽ nhận giá trị đầu vào từ đầu ra ROM 1 là các giá trị $\{I_p^T\}$và cứ sau i chu kì T thì giá trị $\{I_p^{T\prime}\}$ sẽ được tính như sau:

$$\{I_p^{T\prime}[i]\} = (\{I_p^T\} + i)\ mod\ N \tag{3.12}$$

Với i = 0, 1, 2,…, N – 1.

Như vậy, dãy $\{b_n\}$ đầu ra sẽ được tính như sau:

$$b_n = \left\{\{a_n[I_p^{T\prime}[0]]\}, \{a_n[I_p^{T\prime}[1]]\}, \{a_n[I_p^{T\prime}[2]]\}, \ldots, \{a_n[I_p^{T\prime}[\text{N-1}]]\}\right\} \tag{3.13}$$

Trong đó, $a_n\left[I_p^T[i]\right]$ là giá trị a_n tại pha $I_p^T[i]$.

4) ROM 2

Đây là nơi thực hiện việc lưu giá trị của dãy tuyến tính $\{a_n\}$ được tạo từ khối PSG.Bộ nhớ của khối này có độ lớn là N địa chỉ lưu trữ.

3.4. Ứng dụng dãy lồng ghép phi tuyến trong kỹ thuật mật mã

Các nghiên cứu về kiến trúc dãy lồng ghép phi tuyến ở trên đã cung cấp một bộ tạo dãy có tính chất giả ngẫu nhiên tốt, có khả năng ứng dụng trong kỹ thuật mật mã. Để có thể áp dụng dãy lồng ghép phi tuyến vào việc mã hóa luồng dữ liệu thoại số, tác giả luận án đề xuất quy trình thực hiện như sau

Bước 1: Lựa chọn tham số

Trường Galois sẽ sử dụng là trường GF(2^n) để có thể dễ dàng khai thác các ưu thế của các hệ vi xử lý cũng như FPGA khi tính toán nhị phân.

Chọn bậc dãy ban đầu là n = 26 với m=13. Giá trị các tham số khác như sau:

S = (2^{26}-1) / (2^{13}-1) = 8193

Bộ tham số được lựa chọn như trên có một lý do khác là do với giá trị tham số này, toàn bộ tập tập thứ tự lồng ghép IPS (8193 x 16 bit) có thể lưu trữ gọn trong một Block RAM 18Kbit của FPGA.

Bước 2: Lựa chọn đa thức sinh và tính toán đa thức con

Để tạo dãy lồng ghép phi tuyến ta cần hai dãy ban đầu với hai đa thức sinh nguyên thủy được lựa chọn là:

$f(d) = d^{26} + d^{22} + d^{21} + d^{18} + d^{13} + d^{12} + d^{10} + d^8 + d^6 + d^5 + d^2 + d + 1$

$g(d) = d^{26} + d^{20} + d^{19} + d^{18} + d^{16} + d^{15} + d^{14} + d^{13} + d^9 + d^8 + d^4 + d + 1$

Với các tham số trên, ta sử dụng công thức sinh m-dãy để sinh ra hai bộ m*S bit tương ứng với hai dãy, sau đó sử dụng thuật toán Belekamp – Massey để tính được các đa thức con của hai dãy con là:

$f_1(d) = d^{13} + d^{12} + d^{11} + d^9 + d^7 + d^5 + d^4 + d^3 + 1$

$g_1(d) = d^{13} + d^{11} + d^{10} + d^8 + d^5 + d^3 + d^2 + d + 1$

Trong thực tế ta không cần sử dụng tới $f_1(d)$, chỉ cần tìm $g_1(d)$ là đủ

Bước 3: Tìm tập thứ tự lồng ghép IPS cho dãy lồng ghép thứ nhất (với đa thức sinh f(d) và đa thức con $f_1(d)$)

Tạo bảng lồng ghép từ bộ m*S phần tử của m-dãy thứ nhất theo đa thức sinh f(d)

Ta sinh ra toàn bộ chu kỳ 2^m-1 phần tử của dãy con 1 theo đa thức sinh $f_1(d)$

So sánh các cột của bảng lồng ghép với từng đoạn con m-bit lệch pha trong dãy con để xác định từng phần tử của tập thứ tự lồng ghép I_k. Nếu cột thứ k của bảng lồng ghép trung với m phần tử của dãy con bắt đầu từ vị trí j thì I_k = j. Nếu cột thứ k của bảng lồng ghép chứa m bit toàn 0 thì ta gán I_k = -1 (trường hợp này theo mô tả lý thuyết ở trên thì cần đặt $I_k = \infty$, song để biểu diễn trong mảng số nguyên ta sử dụng giá trị -1).

Bước 4: Thực thi sinh dãy lồng ghép phi tuyến trong thực tế

Toàn bộ 3 bước trên là các bước tiền xử lý, thực hiện trong quá trình chuẩn bị. Dữ liệu được lưu trữ để thực thi sinh dãy lồng ghép phi tuyến bao gồm tham số n, m, S, đa thức $g_1(d)$ và tập thứ tự lồng ghép IPS.

Để sinh một phần dãy lồng ghép phi tuyến sử dụng cho việc mã hóa một buffer dữ liệu, ta thực hiện các bước sau

Sử dụng công thức sinh m-dãy để sinh đầy đủ chu kỳ của dãy con với đa thức sinh $g_1(d)$ bậc m, lưu kết quả trong mảng dữ liệu kích thước 2^m-1 phần tử. Để tránh thao tác quay vòng dữ liệu, ta copy nhân đôi mảng dữ liệu thành $2(2^m-1)$ phần tử.

Để sinh chuỗi khóa lồng ghép phi tuyến từ giá trị khởi đầu n phần tử, ta sẽ tách riêng (n-m) bit đầu tiên của giá trị khởi đầu, chuyển thành một số nguyên k để xác định thứ tự cột trong ma trận lồng ghép. Ta cũng chuyển m bit còn lại thành số nguyên t để xác định vị trí bắt đầu lấy khóa trong cột.

Như vậy chuỗi khóa lấy ra sẽ được bắt đầu từ vị trị I_k + t, lấy liên tục tới vị trí I_k+2^m-2. Nếu chưa đủ lượng bit khóa đầu ra cần thiết, ta sẽ tiếp tục chuyển sang các cột tiếp theo với chuỗi khóa lấy từ I_{k+1} tới $I_{k+1}+2^m-2$. Quá trình cứ tiếp tục như vậy tới khi lấy được đủ lượng bit khóa đầu ra theo yêu cầu. Nếu giá trị cột khóa vượt quá S-1 ta lại quay lại cột đầu tiên. Nếu I_k = -1 thì chuỗi khóa đầu ra là lấy từ một chuỗi toàn 0 kích thước 2^m-1 phần tử.

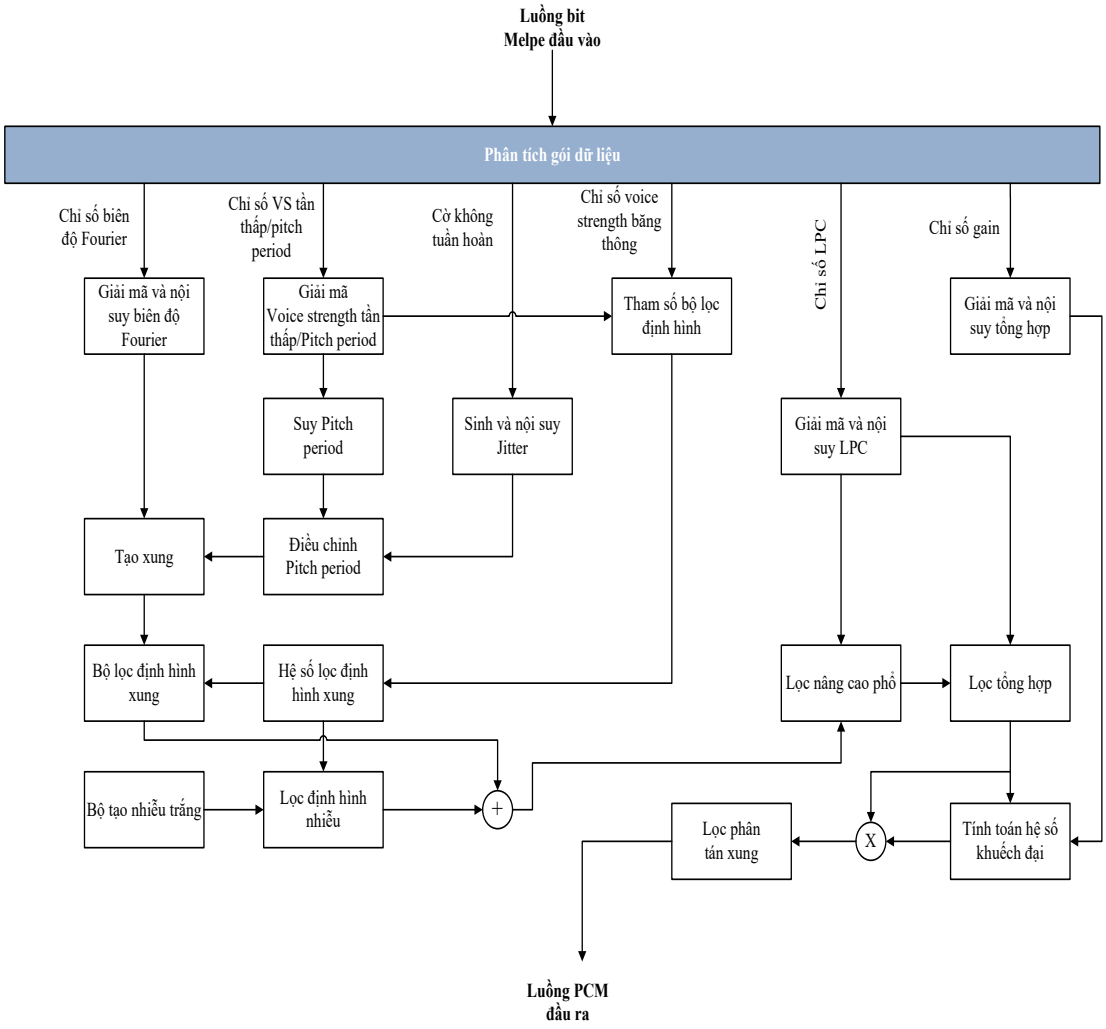
Để có thể sử dụng chuỗi khóa đầu ra trong môi trường vi xử lý, ta cần chuyển từ dãy bit thành dãy các byte nhị phân bằng cách ghép 8 bit liên tục thành một byte dữ liệu.

Quá trình mã hóa và giải mã dữ liệu thực hiện theo phương pháp mã dòng (Stream Cipher) thông thường: khi mã hóa thì bản mã là kết quả cộng module 2 từng bit (XOR) giữa bản rõ và khóa, ngược lại khi giải mã thì bản mã là kết quả cộng module 2 từng bit (XOR) giữa bản mã và khóa.

3.5. Thực thi thuật toán nén Melpe bằng Vi xử lý STM32F

3.5.1. Lưu đồ thuật toán nén thoại Melpe trên ARM [24]

Lý thuyết phầns nén/giải nén thuật toán Melp đã được trình bày trong chương 1, phần dưới đây chỉ trình bày các giá trị thực nghiệm nén Melpe thực hiện trên Vi xử lý ARM STM32F:



Hình 3.6. Lưu đồ giải nén thoại thuật toán Melpe trên ARM

- Mã hóa voice strength băng thông: Voice strength của bốn dải tần cao được lượng tử hóa theo thủ tục mã giả. Thủ tục này có đầu vào là voice strength của 5 dải tần. Đối với trường hợp không phải tiếng nói, dựa vào biên độ voice strength đầu tiên ≤ 0.6, giá trị voice strength lượng tử hóa của bốn dải tần cao sẽ được đặt là 0. Ngược

lại, chúng sẽ được lượng tử hóa là 0 hoặc 1 tùy theo biên độ là nhỏ hơn hay lớn hơn 0.6.

- Lượng tử hóa pitch period và voice strength tần thấp: Pitch period T và voice strength thấp tần được lượng tử hóa cùng nhau sử dụng 7 bit. Nếu giá trị voice strength đầu tiên ≤ 0.6, khung là không tiếng và một mã toàn giá trị 0 sẽ được gửi đi. Ngược lại, logT sẽ được lượng tử hóa với bộ lượng tử đồng nhất 99 cấp có phạm vi từ log20 tới log160. Giá trị voice strength đầu tiên đã lượng tử hóa, nó sẽ nhận giá trị 0 đối với trạng thái không tiếng và 1 đối với trạng thái có tiếng.

- Tính toán độ khuếch đại (Gain): Gain được đo hai lần trên một khung sử dụng kích thước cửa sổ thích nghi âm sắc. Kích thước này được xác định như sau:

+ Nếu voice strength đầu tiên > 0.6, kích thước cửa sổ là bội nhỏ nhất của T(1), thường lớn hơn 120 mẫu. Nếu kích thước này vượt quá 320 mẫu, nó sẽ được chia cho 2. Trường hợp này tương ứng với các khung tiếng nói, khi sự đồng bộ âm sắc được tìm trong suốt quá trình tính toán gain. Bằng cách sử dụng một bội nguyên của pitch period, sự biến thiên của gain tương ứng với vị trí của cửa sổ sẽ được giảm thiểu.

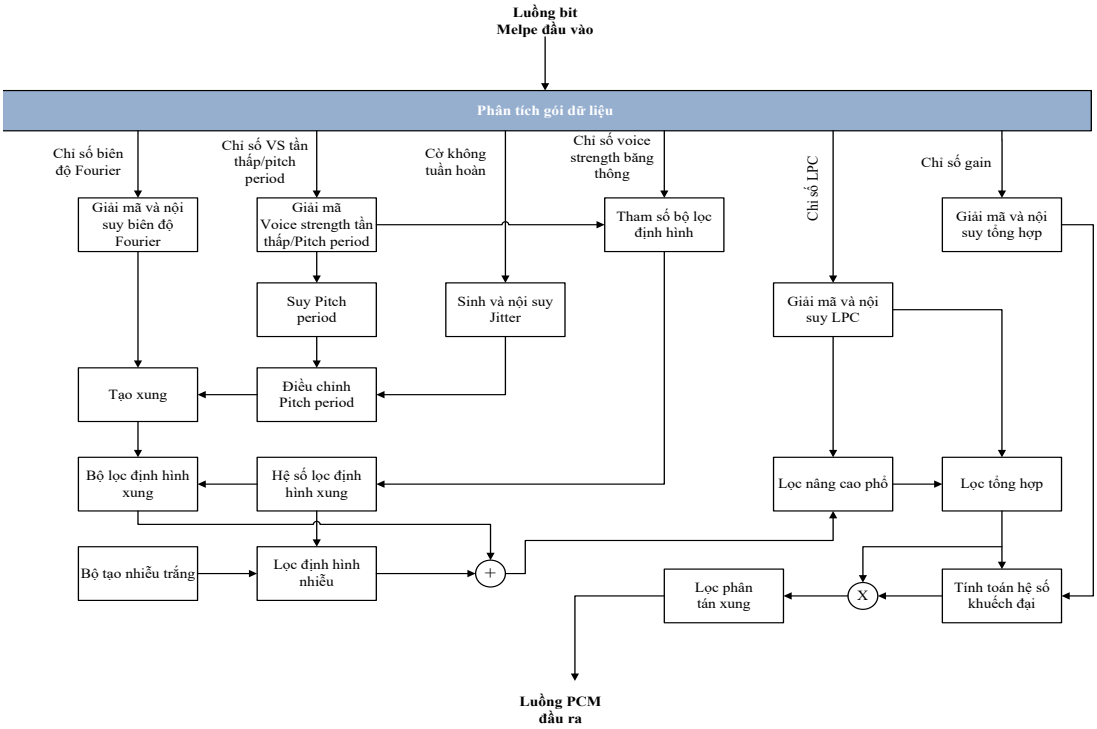
+ Nếu voice strength đầu tiên ≤ 0.6, kích thước cửa sổ là 120 mẫu. Đây là trường hợp khung không tiếng hoặc khung tiếng chập chờn.

- Mã hóa gain: Giá trị gain thứ 2 được lượng tử hóa bởi một bộ lượng tử đồng nhất 5bit có phạm vi từ 10 tới 77 dB. Một số điều kiện được xác định để xem khung là trạng thái ổn định hay không (năng lượng ít thay đổi). Nếu điều kiện được thỏa mãn, thì thủ tục kết thúc với chỉ số bằng 0. Ngược lại, khung này là tạm thời và ta sử dụng bộ lượng tử hóa đồng nhất cấp 7. Giá trị giới hạn của bộ lượng tử (gmin, gmax) được tính toán; mã hóa thông qua bộ lượng tử đồng nhất. Như vậy, tổng cộng có 8 mức, có thể khai thác bằng 3 bits.

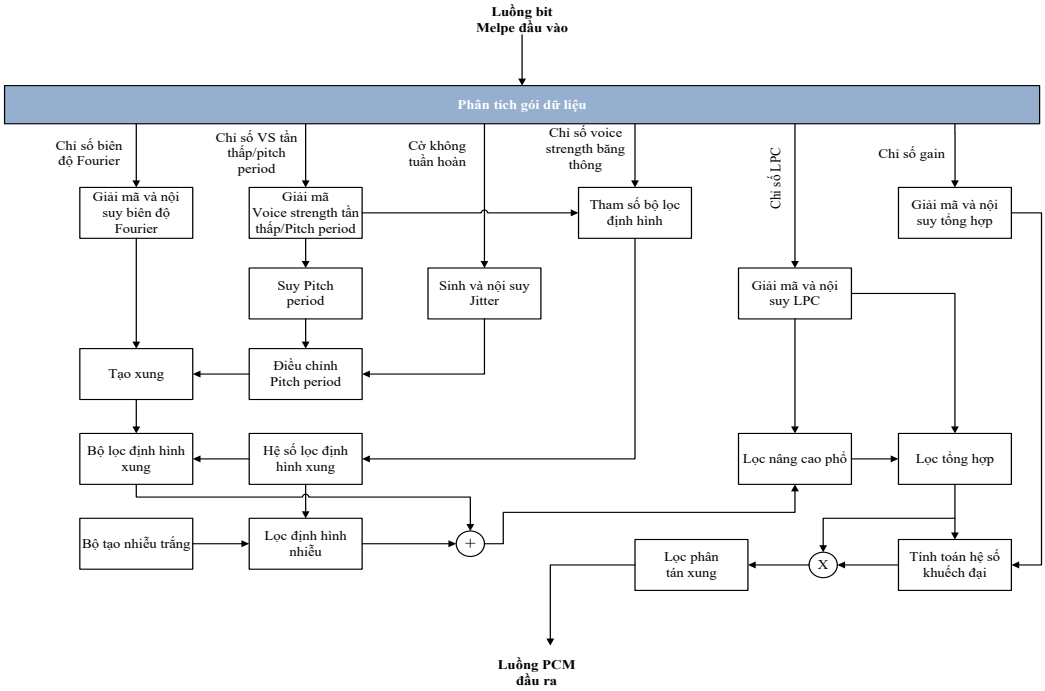
- Cấp phát bit: Ta đã biết, các mô hình LPC đều được lượng tử như là LSF sử dụng MSVQ. Việc đồng bộ hóa này là một mẫu thay đổi 1/0. Việc bảo vệ lỗi chỉ dành

cho các khung không tiếng. Mỗi khung sẽ được truyền tổng cộng là 27 bit, với độ dài là 22.5 ms. Kết quả ta có bit-rate là 1200 bps.

3.5.2. Lưu đồ thuật toán giải nén Melpe trên ARM [24]



Hình 3.7. Lưu đồ giải nén thoại thuật toán Melpe trên ARM



Hình 3.8. Lưu đồ giải nén thoại thuật toán Melpe trên ARM

Luồng bit Melpe đầu vào được phân tích với các chỉ số được đưa đến các bộ giải mã tương ứng. So với lưu đồ nén, ta thấy mô hình tạo tiếng nói đã được nhúng bên trong cấu trúc của phần giải nén. Có hai bộ lọc được bổ sung vào trong quá trình xử lý đó là bộ lọc nâng cao phổ với đầu vào là kích thích hỗn hợp và bộ lọc phân tán xung ở cuối của quá trình xử lý. Hai bộ lọc này được sử dụng để nâng cao chất lượng của tiếng nói tổng hợp (luồng PCM đầu ra).

Trong giải mã Melpe, các tham số từ dòng bit sẽ được phân tích và giải mã theo các lược đồ tương ứng. Những tham số này bao gồm: LPC (LSF), pitch period/*voice strength* tần thấp, *voice strength* băng thông, gain (g1 và g2), cờ không tuần hoàn và các biên độ Fourier. Các tham số này đại diện cho thông tin của khung, hầu hết được nội suy một cách tuyến tính trong quá trình tổng hợp tiếng nói.

Đối với các khung không tiếng (được phát hiện thông qua mã voice strength thấp tần/ pitch period), chúng ta sẽ sử dụng các giá trị mặc định cho một vài tham số, đó là pitch period = 50, jitter = 0.25, tất cả các biên độ Fourier đều là 1, và tất cả các giá trị voice strength đều là 0. Các giá trị mặc định này là cần thiết đối với khung không tiếng bởi vì việc nội suy tuyến tính được thực hiện trên cơ sở "pitch period – by – pitch period" trong suốt quá trình tổng hợp. Ở đây có xuất hiện một tham số mới: jitter, nó chỉ được sử dụng trong việc giải mã để điều khiển số lượng ngẫu nhiên xảy ra trong quá trình tạo ra các âm kích thích không tuần hoàn.

Đối với khung có tiếng, giá trị của jitter được sử dụng như sau: jitter = 0.25 nếu cờ không tuần hoàn là 1, ngược lại thì jitter = 0. Trong trường hợp này, pitch period được giải mã từ dòng bit.

Bộ lọc tổng hợp: Đây là một bộ lọc tổng hợp đỉnh cộng theo hình thức trực tiếp, với các hệ số tương ứng với LSF đã suy ra được.

Bộ lọc phân tán xung: Bộ lọc này là một bộ lọc FIR 65 lớp trích xuất từ một xung tam đỉnh phổ phẳng. Như ta thấy thì nó gần như là một bộ lọc thông suốt, khi mà các

thay đổi trong đáp ứng biên độ là tương đối nhỏ. Bộ lọc phân tán xung được dùng để cải thiện cho bộ lọc tổng hợp băng thông với tiếng nói tự nhiên dạng sóng trong các vùng không có cộng hưởng đỉnh. Tiếng nói tự nhiên đã qua lọc băng thông thì có một tỉ lệ đỉnh-trũng nhỏ hơn so với tiếng tổng hợp.

3.6. Tối ưu hóa melpe

Do hiệu suất tuyệt vời và tốc độ bit thấp của MELPe, nó thường được sử dụng nhiều trong các lĩnh vực, đặc biệt là trong an ninh quốc phòng. Tuy nhiên, thuật toán của MELPe rất phức tạp và tốn nhiều thời gian.

3.6.1. Phân tích hiệu suất

Trong luận văn này, MELPe được áp dụng trên nền tảng của ARM STM32F437 Cortex M4. Sử dụng Gprofile (GNU profiler), là một công cụ lập hồ sơ thống kê trên toàn hệ thống, được sử dụng để phân tích hiệu suất của mã nguồn. Công cụ này có khả năng lập hồ sơ toàn bộ chương trình, tìm ra nơi chương trình đã dành thời gian và số lần hàm được gọi, đây là một tham số quan trọng trong việc tối ưu hóa.

Bảng thống kê dưới cung cấp cấu hình các hàm thực thi của bộ mã hóa MELPe trước khi tối ưu hóa. Với dữ liệu hồ sơ, tối ưu hóa của chương trình, thường được chia thành hai loại: tối ưu hóa thuật toán và tối ưu hóa mã nguồn, trong đó mỗi phương pháp tối ưu hóa có thể chi tiết hóa.

Đối với các hàm số lần gọi ít, nhưng mỗi lần gọi rất lâu. Chẳng hạn như iir_2nd_s. Số mili giây trung bình dành cho hàm này trên mỗi lần gọi là 0,03 mili giây. Loại hàm này có thể được tối ưu hóa ở cấp độ thuật toán.

Đối với các hàm có số lượng code nhỏ và được gọi thường xuyên, chẳng hạn như L40_mac, L_mac và L_ shl, tỷ lệ phần trăm tổng thời gian chạy chương trình của ba hàm này là 50,81%. Các chức năng này có thể được tối ưu hóa ở cấp mã.

Bảng 3.5. Thống kê các hàm thực thi chính của Melpe

Tên hàm thực thi	Số lần gọi hàm	Thời gian (%)
L40_mac	18109111	21.28
L_mac	69517406	19.36
L_hsl	14839245	10.17
L_mult	16685416	4.06
L_v_inner	187816	3.61
Iir_2nd_s	20168	2.64
L_40_shl	423407	2.43
Shr	7950898	2.27

3.6.2. Tối ưu hóa thuật toán (Optimization of algorithm)

Mục tiêu chính của việc tối ưu hóa thuật toán là đơn giản hóa mà không làm giảm chất lượng giọng nói. Để tối ưu hóa logic của thuật toán mã, các phương pháp phổ biến nhất bao gồm thuật toán tái cấu trúc, sửa đổi thứ tự của mã và loại bỏ tính toán thừa.

***1) Cấu trúc lại bộ lọc IIR*:** Trong thuật toán của MELPe, quy trình của bộ mã hóa bao gồm mô-đun tính toán đỉnh dư (residual peak calculation module) và sửa đổi mô-đun cường độ giọng nói băng thông (modification of bandpass speech strength module). Hai mô-đun này được gọi là bộ lọc IIR bậc hai (hàm "iir_2nd_s") nhiều lần. Bên trong hàm, có một lệnh lặp "for" gọi hàm "L_mult" và hàm "L_mac". Các hàm này bao gồm nhân, chuyển và cộng. Câu lệnh shift có thể được đưa ra khỏi vòng lặp

và được thực thi ở cuối vòng lặp. Điều này không thay đổi kết quả, nhưng đơn giản hóa tính toán.

2) Đơn giản hóa câu lệnh lựa chọn: Để đáp ứng nhu cầu của nhiều loại lệnh gọi khác nhau, một số hàm đang sử dụng rất nhiều cấu trúc if-else. Hạn chế là các mã này có thể tiêu tốn một lượng lớn thời gian thực hiện trong các lệnh phán quyết và nhảy. Do đó các hàm này phải được tối ưu hóa từ cấu trúc bằng cách được viết lại dựa trên tần suất của lệnh rẽ nhánh.

3.6.3. Tối ưu hóa mã (Optimization of code)

Mã nguồn MELPe tiêu chuẩn được viết bằng ANSI-C, sử dụng thiết kế mô-đun để đảm bảo khả năng đọc tốt. Nhưng điều đó cũng làm tăng số lượng lệnh gọi hàm, làm giảm hiệu quả. Theo quy tắc 2/8 tức là 80% thời gian chạy được sử dụng trong 20% mã, để tối ưu hóa hiệu quả hơn, trọng tâm của việc tối ưu hóa là nhằm vào 20% mã chạy chính. Phong cách mã hóa C nên được thay đổi để phù hợp với các đặc điểm của kiến trúc và trình biên dịch ARM.

1) Tối ưu hóa các lệnh cơ bản: Tệp "mathhalf.c" chứa rất nhiều hàm thực hiện các phép toán số học nguyên thủy. Nhiều trong số chúng là các phép toán cơ bản, chẳng hạn như nhân và tích lũy 32 bit (hàm "L _ mac") hoặc phép cộng 32 bit bão hòa (hàm "L_add"), có thể được thực hiện bằng lệnh Extended ARM của như SMLAL và QDADD. Sử dụng các hướng dẫn đặc biệt này có thể lưu các instructions một cách hiệu quả.

2) Hàm nội tuyến (Inline function): Việc thêm tiền tố nội tuyến vào hàm có thể loại bỏ thời gian gọi hàm, vốn tiêu tốn nhiều thời gian. Hàm nội tuyến đang thay thế trình gọi bằng mã nguồn của hàm, điều này sẽ làm tăng kích thước mã, cụ thể là trao đổi không gian lấy thời gian. Do đó, chỉ có hàm với dung lượng mã nhỏ là thích hợp để sử dụng hàm nội tuyến.

Theo bảng trên, các hàm được sử dụng thường xuyên nhất là “L40_mac”, “L_mac”, “L_shl”, “L_mult”, tác động của kích thước không gian lưu trữ là rất nhỏ, nhưng hiệu quả rất rõ ràng khi giảm 20% tổng thời gian chạy.

***3) Tối ưu hóa vòng lặp**:* Hầu hết các chương trình quan trọng sẽ chứa một vòng lặp. Trên nền tảng ARM, các vòng lặp có instructions nhỏ khi chúng đếm ngược về 0. Đôi khi, các vòng lặp unrolling (loop unrolling) có thể đạt được hiệu suất tối đa. Đây là các kỹ thuật đều được hiệu chỉnh trong code.

3.7. Phân tích kết quả thực nghiệm

Nền tảng thử nghiệm được xây dựng dựa trên hệ thống ARM Cortex M4. Quá trình mã hóa và giải mã được thực hiện bởi vi điều khiển STM32F437 của hãng ST dựa trên lõi ARM Cortex M4:

+ Core: Arm® 32-bit Cortex®-M4 CPU với bộ tính toán số thực FPU, hoạt động với tần số 180 MHz, tỷ suất DMIPS/MHZ cao 1.25 giúp cho hệ thống có thể đạt được hiệu năng 225 DMIPS.

+ Bộ nhớ: dung lượng bộ nhớ Flash 2 MByte, dung lượng SRAM 256Kbyte.

Bảng 3.6 dưới đưa ra độ trễ của hai chuỗi lời kiểm tra trước và sau khi tối ưu hóa. Thời gian của hai bài kiểm tra là 16,75 s và 3 s. Sau khi tối ưu hóa, độ trễ mã hóa mỗi khung hình giảm 63,6% và độ trễ giải mã mỗi khung hình giảm 41,6%. Tổng độ trễ của thuật toán MELPe trên mỗi khung hình là khoảng 55,4 ms, đáp ứng nhu cầu giao tiếp thời gian thực. Chất lượng giọng nói được kiểm tra bởi PESQ (Đánh giá cảm nhận về chất lượng giọng nói). Kết quả PESQ của giọng nói được mã hóa sau khi tối ưu hóa là 3.201, rất gần với kết quả PESQ trước khi tối ưu hóa, 3.158. PESQ cho thấy rằng việc tối ưu hóa không làm giảm chất lượng giọng nói.

Bảng 3.6. So sánh độ trễ tính toán

Thời gian thoại (giây)	Frame	Enc/Dec	Delay khi chưa tối ưu (ms)	Delay sau khi tối ưu (ms)

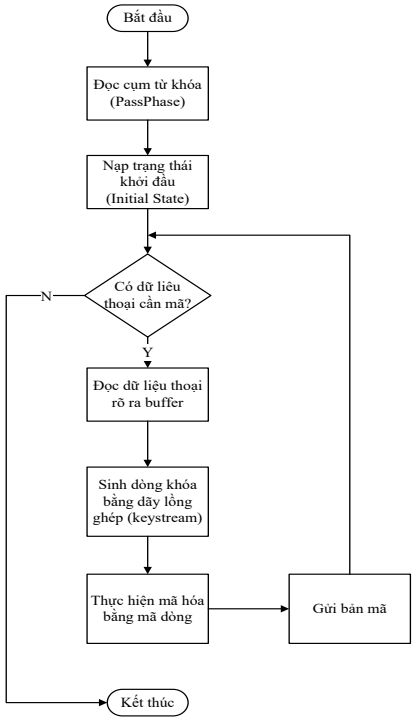
16.75	248	Encode	127.1	46.2
16.75	249	Decode	16.6	9.6
3	44	Encode	111.2	45.5
3	45	Decode	14.8	8.7

*** Với ba tính năng bổ sung, MELPe có hiệu suất tốt hơn ở tốc độ bit thấp hơn. Để đáp ứng nhu cầu ứng dụng kỹ thuật dựa trên ARM Cortex M4, việc tối ưu hóa được thực hiện theo hai cách, bao gồm tối ưu hóa thuật toán và tối ưu hóa mã. Sau khi tối ưu hóa, độ trễ của mỗi frame được giảm từ 135.1 mili giây xuống 55.4 mili giây mà chất lượng không giảm. Các thí nghiệm chỉ ra rằng hiệu quả của việc tối ưu hóa, đáp ứng nhu cầu thực hiện theo thời gian thực.

3.8. Lưu đồ giải thuật khối mã hóa/giải mã

Giải pháp phân phối cụm từ khóa (passphase) được lựa chọn trong Luận án là giải pháp phân phối trước. Hai bên sẽ biết trước được cụm từ khóa giống nhau.

3.8.1. Lưu đồ giải thuật khối mã hóa



Hình 3.9. Lưu đồ giải thuật khối mã hóa

Phía gửi sẽ đọc cụm từ khóa (PassPhase) mà 2 bên đã thống nhất từ trước và nạp trạng thái khởi đầu. Hai bước này sẽ được thực hiện duy nhất lần đầu trong 1 phiên liên lạc.

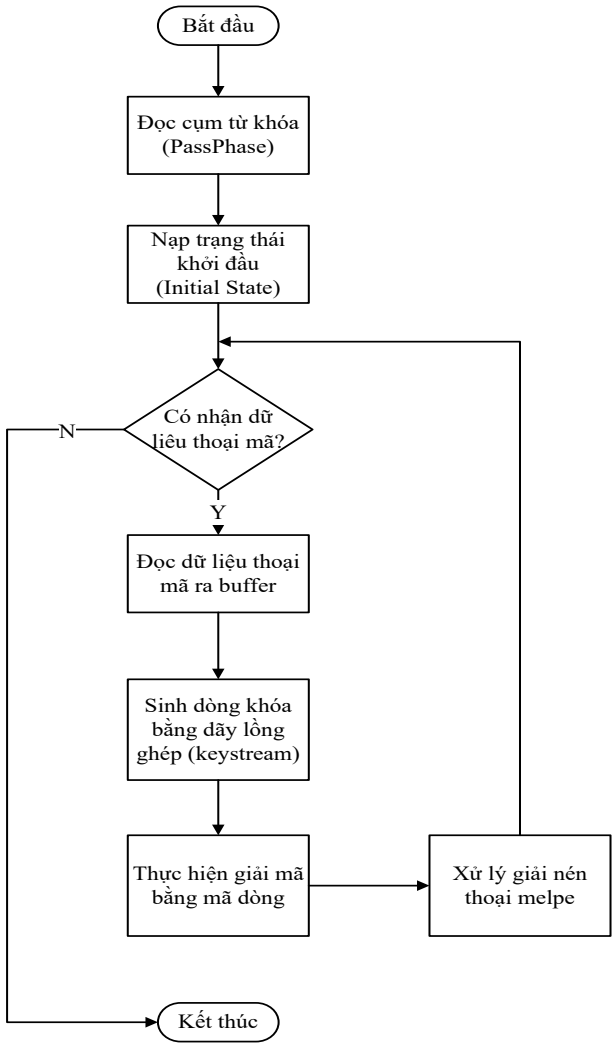
Khi có dữ liệu thoại cần mã (luồng dữ liệu Melpe) từ khối nén thoại Melpe sẽ được lưu vào bộ nhớ đệm buffer. Nếu không có sẽ kết thúc phiên.

Tiếp theo sẽ sinh dòng khóa (keystream) bằng dãy lồng ghép.

Thực hiện mã hóa luồng dữ liệu Melpe bằng mã dòng và tạo ra bản mã thoại.

Thực hiện gửi bản mã thoại sang phía bên nhận.

3.8.2. Lưu đồ giải thuật khối giải mã



Hình 3.10. Lưu đồ giải thuật khối giải mã

Phía nhận sẽ đọc cụm từ khóa (PassPhase) mà 2 bên đã thống nhất từ trước và nạp trạng thái khởi đầu.

Khi nhận được dữ liệu thoại mã từ phía gửi, sẽ lưu vào bộ nhớ đệm buffer. Nếu không nhận được sẽ kết thúc phiên.

Tiếp theo sẽ sinh dòng khóa (keystream) bằng dãy lồng ghép.

Thực hiện giải mã luồng dữ liệu thoại mã nhận được bằng mã dòng và gửi luồng dữ liệu đó tới khối xử lý giải nén thoại Melpe.

3.9. Kết luận chương 3

Chương 3 đã giới thiệu tổng quan về m-dãy, các đa thức các thuộc tính của m-dãy, tính chất các dãy lồng ghép; giới thiệu về cấu trúc dãy lồng ghép (bao gồm cả dãy phi tuyến lồng ghép), trong đó kiến trúc dãy lồng ghép có kế thừa nội dung các bài báo của chính nghiên cứu sinh là tác giả và đồng tác giá (bài báo số 1b); về các phương pháp sinh dãy lồng ghép và lồng ghép phi tuyến, từ Luận án nghiên cứu này nghiên cứu sinh đã đóng góp một phương pháp mới (ngoài phương pháp biến đổi -d và hàm Vết đã kế thừa từ các bài báo trước của các đồng tác giả) đó là *phương pháp thứ ba tính toán trực tiếp giá trị* I_P^S *là một phương pháp mới, được tác giả luận án đề xuất trong công bố [1b] này.*

Chương này cũng đưa ra phương pháp là lợi thế của phương pháp tính toán trực tiếp giá trị thứ tự lồng ghép; *xây dựng được bảng so sánh hiệu quả rút gọn tính toán khi ứng dụng phương pháp này*; *Ứng dụng dãy lồng ghép phi tuyến trong kỹ thuật mật mã;* Thực nghiêm đánh giá các dãy lồng ghép cụ thể**,** phương pháp thực thi dãy lồng ghép bằng phần cứng; Tối ưu và thực thi thuật toán nén/giải nén Melpe, phân tích đánh giá hiệu năng sau tối ưu và các thủ tục mã mật/giải mã bằng Vi xử lý ARM STM32F.

KẾT LUẬN

Trong phạm vi luận án, tác giả đã nghiên cứu cơ sở lý thuyết mã thoại, các bộ tạo dãy giả ngẫu nhiên m-dãy; đề xuất thiết kế, phân tích và xây dựng cấu trúc tổng quát của bộ tạo dãy giả ngẫu nhiên phi tuyến dựa trên m-dãy lồng ghép, nghiên cứu một số phương pháp và đưa ra hướng giải quyết trong bảo mật cuộc gọi thoại trong mạng viễn thông và thử nghiệm trên nền tảng phần cứng. Trong quá trình thực hiện luận án, tác giả đã có một số đóng góp khoa học mới, cụ thể như sau:

(i) Đề xuất giải pháp bảo mật dữ liệu thoại sử dụng thuật toán sinh số giả ngẫu nhiên dựa trên dãy phi tuyến lồng ghép;

(ii) Đề xuất thuật toán cải tiến, nâng cao chất lượng mã thoại MELPe và giải pháp truyền dữ liệu thoại bảo mật qua kênh thoại GSM;

(iii) Đề xuất thực hiện kỹ thuật điều chế và giải điều chế để truyền dữ liệu thoại đã được mã hóa bảo mật qua các thiết bị đầu cuối và mạng (liên mạng) truyền dẫn.

Với những đóng góp khoa học nêu trên, luận án là cơ sở để nghiên cứu, phát triển cho các hệ thống truyền dẫn bảo mật tín hiệu thoại qua kênh thoại GSM và qua các nền tảng khác nhau dựa trên kênh thoại. Các thuật toán, giải pháp và thiết bị được chứng minh và mô phỏng, đánh giá rõ ràng, thực hiện cài đặt thuật toán trên chip FPGA hoặc ARM tạo ra Module được kiểm tra an toàn, thẩm định tính thực thi đúng đắn với lý thuyết để có thể ứng dụng đáp ứng nhu cầu cấp thiết trong thực tế.

Các vấn đề cần nghiên cứu tiếp

Việc phát triển thuật toán nâng cao chất lượng tiếng nói cho phép thiết kế, chế tạo phần cứng thiết bị điện thoại di động, cài đặt các thư viện, các chương trình điều khiển, các thuật toán và hoàn thiện thành một thiết bị điện thoại di động có bảo mật dùng kênh 2G của mạng viễn thông di động GSM đảm bảo tính an toàn trong cài đặt thuật toán vào thiết bị.

Hướng tiếp theo là nghiên cứu lý thuyết lấy mẫu theo Nyquist đa băng con để tăng tốc độ điều chế / giải điều chế Modem OFDM, thực thi tích hợp toàn bộ Modem này vào Chip ARM để có thể lắp vào điện thoại di động. Lập trình trên chip với không gian chật hẹp, tài nguyên hạn chế nên yêu cầu phải tối ưu hóa về tốc độ, về kích thước

mã chương trình, về không gian vùng nhớ dữ liệu và vùng nhớ phục vụ thao tác tính toán. Hướng khác là tích hợp chức năng modem vào phần mềm của điện thoại di động thông minh.

DANH MỤC CÁC CÔNG TRÌNH ĐÃ CÔNG BỐ CỦA LUẬN ÁN

1b. Hieu Le Minh, Truong Dang Van, **Binh Nguyen Thanh** and Quynh Le Chi, "Construction of Nonlinear q-ary m-sequences with Interleaved Structure by d-Transform", IEEE ICCE 2018, pp.389-392, 2018.

2b. **Nguyễn Thanh Bình**, Nguyễn Thành Vinh, Nguyễn Xuân Liêm. "Phân tích, thiết kế tích hợp hệ mã thoại Vocoder dựa trên chuẩn MELP cải tiến phục vụ bảo mật thoại và dữ liệu qua kênh vô tuyến HF chuyên dụng", Tạp chí Khoa học và Công nghệ (Journal of Science and Technology), Số 115, bài số 10, 11/2016,

3b. **Nguyễn Thanh Bình**, Đặng Vân Trường, Trần Văn Liên, "Một phương án truyền dữ liệu qua kênh thoại GSM", Tạp chí Khoa học Công nghệ Thông tin và Truyền thông (Journal of Science and Technology on Information and Communications), Số 03&04, trang 80 – 86, năm 2019,

4b. Đặng Vũ Sơn, **Nguyễn Thanh Bình**, Nguyễn Hữu Trung, "Về vấn đề đảm bảo an ninh mạng thông tin vô tuyến theo tiếp cận xử lý tín hiệu nhiều chiều", Tạp chí An Toàn Thông Tin, Số 1, bài số 6, năm 2015,

TÀI LIỆU THAM KHẢO

1. TS. Nguyễn Phạm Anh Dũng*, Thông tin di động, 2013*

2. Ammar Yasir Korkusuz*. Security in the GSM Network. Bogazici Univercity, Electrical-Electronics Engineering Department, 2012.*

3. Wilayat Khan, Habib Ullah*. Authentication and Secure Communication in GSM, GPRS, and UMTS Using Asymmetric Cryptography. IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 9, May 2010.*

4. La Hữu Phúc, Lê Mỹ Tú. *Truyền dữ liệu qua kênh thoại GSM với CD-FSK*. Chuyên san Nghiên cứu Khoa học và Công nghệ trong lĩnh vực An toàn thông tin. *Số 1.CS (01) 2015.*

5. Sigurdur Sverrisson, Xiaoyun Liang*. Digital Communication over Speech Compressed Channel. CHALMERS UNIVERSITY OF TECHNOLOGY, Goteborg, Sweden. EXE028/2008.*

6. Solutions to the GSM Security Weaknesses, Mohsen Toorani & Ali A. Beheshti, 2008

7. Cryptography : An Introduction (3rd Edition), Nigel Smart

8. Wai C. Chu (2003)*, Speech coding algorithms – Foundation and evolution of standardized coders,* A JOHN WILEY & SONS, INC PUBLICATION

9. Lecture 03: SOUND PROPAGATION (Deller, et. al., Discrete-Time Processing of Speech Signals, MacMillan Publishing Co., ISBN: 0-7803-5386-2, 2000).

9b. Lecture 3_winter_2012_6tp

10. Speech Coding: A Tutorial Review; ANDREAS S. SPANIAS; Proceedings of the IEEE, Vol 82, No 10, October 1994.

11. Alan McCree*, Low_Bit_Rate Speech Coding. Springer Handbook on Speech Processing and Speech Communication.*

11b. Qiuyun Hao,Ye Li, Peng Zhang, Yanhong Fan, Xiaofeng Ma, Jingsai Jiang ; Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Jinan, China. *A 600BPS*

MELP VOCODER WITH VOICE ACTIVITY DETECTION; 978-1-5090-0654-0/16/$31.00 ©2016 IEEE (ICALIP 2016)

12. Carl Kritzinger, Low Bit Rate Speed Coding. Apr 2006

13. MIL-STD-3005 MELP

14. NATO_STANG_4591 MELPe

15. Goldberg, R. G*, Practical Handbook Of Speech Coders. Boca Raton: CRC Press LLC, 2000*

16. www.gsm-security.net

17. Xiaoqun Zhao*, "Digital Speech Coding", China Machine Press, pp.171-189, May 2007*

18. Jie Meng*, "System implementation of MELP speech codec based on 1.2k", May 2012*

19. Fateme Khalili *; K.N.Toosi; Hossein Sameti, "Design and implementation of Vector Quantizer for a 600 bps cocoder Based on MELP", 11th International Conference on Advanced Communication Technology, 2009. ICACT 2009*

20. Fan.P.Z and Darnell.M (1996), *"Sequence Design for Communications Applications", New York: Wiley, 1996.*

21. Bùi Lai An, *"Về một cấu trúc tổng quát của mã tựa ngẫu nhiên phi tuyến đa cấp – đa chiều theo kiểu lồng ghép", luận án Tiến sĩ kỹ thuật, Học viện công nghệ Bưu chính Viễn thông, 2012.*

22. Lê Minh Hiếu, Lê Chí Quỳnh, *"Design and analysis of sequences with interleaved structure by d-transform", IETE Journal of Research, vol. 51, no. l, tr.61-67, Jan-Feb. 2005*

23. S. Prasad, Lê Chí Quỳnh*, "Equivalent linear span analysis of binary sequences having an interleaved structure," IEE PROCEEDINGS, Vol. 133, Pt. F, No. 3, June 1986, tr.288-292*

24. Chu, W.C.*: Speech Coding Algorithms: Foundation And Evolution of Standardized Coders. Wiley, Berlin (2003)*

25. Zdenko MEZGEC, Amor CHOWDHURY, Bojan KOTNIK: *Implementation of PCCD-OFDM-ASK Robust Data Transmission over GSM Speech Channel*, INFORMATICA, 2009 Institute of Mathematics and Informatics, Vilnius, Vol. 20, No. 1, 51–78

26. CHMAYSSANI, HENDRYCKX: *Data transmission over voice dedicated channels using digital modulations*. du Gros Chêne, 95610 Eragny, France

27. Christoph K. Ladue, Vitaliy V. Sapozhnykov, and Kurt S. Fienberg: *A Data Modem For GSM Voice Channel*, IEEE Transactions on Vehicular Technology, Vol. 57, No. 4, July 2008.

28. Parwinder pal singh, Bhupinder singh, Satinder pal Ahuja: *Need of Secure Voice Encryption and its Methods*, ISSN: 2277 128X, Volume 2, Issue 1, January 2012.

29. Wesley Tanner, Nick Lane-Smith, Keith Lareau: *End to End Voice Encryption, DEFCON-13*, CellutarCrypto.com

*3*0. Xiaoyun Liang, Sigurdur Sverrisson: "*Digital Communication over Speech Compressed Channel*", CHALMERS UNIVERSITY OF TECHNOLOGY, Sweden, Exe028/2008.

31. Kazemi, R., Mosayebi, R., Etemadi, S., Boloursaz, M., & Behnia, F. (2012). *A lower capacity bound of secure end to end data transmission via GSM network*. In 6th International Symposium on Telecommunications (IST). (IEEE). pp. 1015–1020

32. Challans, P., Gover, R., & Thorlby. End to end data bearer performance characterization for communications over wide area mobile networks. In *IEEE Seminar Secure GSM and Beyond: End to End Security for Mobile Communications. IET, 2013.*

33. Mobeen Ur Rehman, Muhammad Adnan, Liaqat Ali Khan, Ammar Masood, Mouazma Batool. *Effective Model for Real Time End to End Secure Communication Over GSM Voice Channel.* Springer Science+Business Media, LLC part of Springer Nature 2021

34. Prawit Chumchu: Department of Telecommunication Mahanakorn University of Technology. *A Simple and Cheap End-to-End Voice Encryption Framework over GSM-based Networks*. ©2012 IEEE.

35. Min-Jae Hwang[1], Hong-Goo Kang[2]. *Parameter enhancement for MELP speech codec in noisy communication environment*; arXiv:1906.08407v1 [eess.AS] 20 Jun 2019.

36. Sebastian CIORNEI, Ion BOGDAN. *A low cost and open source solution for end to end secure calls over VoLTE*. ISSN 2286-3540 U.P.B. Sci. Bull., Series C, Vol. 77, Iss. 4, 2015.

37. Tikui Zhang, Sensen Li, Bin Yu; Zhengzhou Information Science and Technology Institute china. *A universal data transfer technique over voice channels of cellular mobile communication networks*. IET Commun. 2020;1–11.

38. Théo Royer; KTH Royal Institute of Technology School of Electrical Engineering and Computer Science. *Pitch Shifting Algorithm Design and Applications in Music*. STOCKHOLM, SWEDEN 2019.

39. A. von dem Knesebeck, P. Ziraksaz, and U. Zolzer. "*High quality time-domain pitch shifting using PSOLA and transient preservation*". In: Audio Engineering Society (Nov. 2010).

40. Hieu Le Minh, Truong Dang Van, Binh Nguyen Thanh and Quynh Le Chi, "*Design and Analysis of Ternary m-sequences with Interleaved Structure by d-Transform*", Journal of Information Engineering and Applications, vol.5, no.8, pp.93-101, 2015.