BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN VŨ KIÊN

NGHIÊN CỨU THIẾT KẾ MA TRẬN VÀ CẢI TIẾN THUẬT TOÁN KHÔI PHỤC TÍN HIỆU ĐƯỢC LẤY MẪU NÉN

LUẬN ÁN TIẾN SĨ KỸ THUẬT

Hà Nội, 2023

BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

TRẦN VŨ KIÊN

NGHIÊN CỨU THIẾT KẾ MA TRẬN VÀ CẢI TIẾN THUẬT TOÁN KHÔI PHỤC TÍN HIỆU ĐƯỢC LẤY MẪU NÉN

Chuyên ngành: Kỹ thuật điện tử

Mã số: 9.52.02.03

LUẬN ÁN TIẾN SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC:

1. TS Nguyễn Ngọc Minh
2. TS Nguyễn Lê Cường

Hà Nội, 2023

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả luận án **"Nghiên cứu thiết kế ma trận và cải tiến thuật toán khôi phục tín hiệu được lấy mẫu nén"** là kết quả nghiên cứu của bản thân cùng sự hướng dẫn của thầy hướng dẫn và sự hợp tác của nhóm nghiên cứu. Kết quả luận án là kết quả mới không trùng lặp với các kết quả của các luận án và công trình đã có.

Hà Nội, ... \ ... \ 2023

Nghiên cứu sinh

Trần Vũ Kiên

LỜI CẢM ƠN

Lời đầu tiên tôi xin gửi lời cảm ơn chân thành, sâu sắc nhất tới thầy Nguyễn Ngọc Minh, thầy Nguyễn Lê Cường những người đã tận tình hướng dẫn, định hướng, dìu dắt, giúp đỡ tôi trên con đường nghiên cứu khoa học cũng như tác phong làm việc nghiêm túc và không biết mệt mỏi của các thầy trong thời gian hướng dẫn tôi làm nghiên cứu sinh và hoàn thành luận án tiến sĩ này.

Luận án cũng không thể được hoàn thành nếu thiếu sự giúp đỡ nhiệt tình và tận tâm của TS. Lê Chí Quỳnh trong việc trao đổi, chia sẻ kinh nghiệm, cùng những buổi sinh hoạt nhóm, thảo luận chuyên môn, có thể nói tôi đã học được rất nhiều điều từ đây, với những gì đã nhận được tôi xin gửi lời cảm ơn chân thành tới họ.

Môi trường và điều kiện học tập, nghiên cứu rất tốt tại cơ sở đào tạo cũng góp phần không nhỏ trong việc hình thành kỹ năng làm việc và kết quả nghiên cứu luận án của tôi. Qua đây tôi xin gửi lời cảm ơn đến Học viện Bưu chính - Viễn thông nơi tôi được đào tạo, nghiên cứu.

Nhân đây, tôi muốn gửi lời cảm ơn tới Ban Giám hiệu Trường Đại học Điện lực cùng các đồng nghiệp nơi tôi công tác đã giúp đỡ, động viên, hỗ trợ và tạo nhiều điều kiện tốt nhất về công tác cho tôi trong thời gian làm nghiên cứu sinh và hoàn thành luận án này.

Và trên hết, tôi xin bày tỏ lòng biết ơn tới gia đình, anh chị và bạn bè những người đã hết sức ủng hộ, động viên về mọi mặt để tôi vững tin hoàn thành luận án này.

Hà Nội, tháng ... năm 2023

MỤC LỤC

LỜI CAM ĐOAN i

LỜI CẢM ƠN ii

MỤC LỤC iii

THUẬT NGỮ VIẾT TẮT vi

DANH MỤC CÁC KÍ HIỆU ix

DANH MỤC CÁC BẢNG, BIỂU xi

DANH MỤC CÁC HÌNH VẼ xi

CHƯƠNG 1. TỔNG QUAN VỀ LẤY MẪU NÉN 6

1.1 Mô hình lấy mẫu nén 6

1.1.1 Tín hiệu thưa 9

1.1.2 Ma trận lấy mẫu nén 11

1.1.3 Thuật toán khôi phục 13

1.2 Hiệu năng của mô hình lấy mẫu nén 15

1.3 Các công trình nghiên cứu liên quan 18

1.3.1 Các nghiên cứu về thiết kế ma trận xác định 19

1.3.2 Các nghiên cứu về thuật toán tham lam 20

1.4 Nhận xét các công trình nghiên cứu liên quan và hướng nghiên cứu của luận án 21

1.4.1 Nhận xét về công trình nghiên cứu liên quan 21

1.4.2 Hướng nghiên cứu của luận án 22

1.5 Tổng kết chương 22

CHƯƠNG 2. THIẾT KẾ MA TRẬN LẤY MẪU NÉN XÁC ĐỊNH 24
2.1 Mở đầu . . . 24
2.2 Tiêu chí thiết kế ma trận lấy mẫu nén . . . 24
2.3 Thiết kế ma trận lấy mẫu nén . . . 26
2.4 Lý thuyết trường hữu hạn . . . 27
2.4.1 Cấu trúc $GF(p^n)$ . . . 27
2.4.2 Thanh ghi dịch phản hồi tuyến tính . . . 29
2.4.3 Biến đổi D . . . 31
2.4.4 Hàm Vết . . . 35
2.5 Chuỗi trải phổ PN phi tuyến lồng ghép . . . 37
2.5.1 Phân hoạch chuỗi lớn . . . 38
2.5.2 Đánh giá chuỗi PN giả ngẫu nhiên lồng ghép phi tuyến . . . 39
2.6 Xây dựng ma trận xác định BPNSM . . . 41
2.7 Tính chất không kết hợp của ma trận BPNSM . . . 43
2.8 So sánh đánh giá ma trận BPNSM . . . 46
2.9 Thực hiện ma trận lấy mẫu nén trên phần cứng . . . 49
2.10 Tổng kết chương . . . 51

CHƯƠNG 3. ĐỀ XUẤT THUẬT TOÁN KHÔI PHỤC TÍN HIỆU ĐƯỢC LẤY MẪU NÉN DRMP 53
3.1 Chỉ tiêu đánh giá thuật toán khôi phục . . . 53
3.2 Các thuật toán lặp lại tham lam . . . 55
3.2.1 Thuật toán đuổi khớp - MP . . . 56
3.2.2 Thuật toán đuổi khớp trực giao - OMP . . . 58
3.2.3 Thuật toán lấy mẫu nén đuổi khớp - CoSaMP . . . 60
3.3 Thuật toán cải tiến DRMP . . . 63
3.3.1 Xây dựng thuật toán DRMP . . . 65
3.3.2 Hiệu năng của thuật toán DRMP . . . 67
3.4 Tổng kết chương . . . 75

CHƯƠNG 4. ĐỀ XUẤT MÔ HÌNH LẤY MẪU NÉN 76
4.1 Mở đầu 76
4.2 Mô phỏng đánh giá mô hình với tín hiệu 1 chiều 77
4.2.1 Ma trận lấy mẫu tín hiệu 1 chiều 80
4.2.2 Khôi phục tín hiệu 1 chiều 82
4.3 Mô phỏng đánh giá mô hình với tín hiệu 2 chiều 89
4.3.1 Ma trận lấy mẫu ảnh số 90
4.3.2 Khôi phục lại ảnh gốc 91
4.4 Ứng dụng mô hình lấy mẫu nén đề xuất 98
4.4.1 Ứng dụng trong cảm nhận phổ băng rộng 98
4.4.2 Ứng dụng lấy mẫu nén ảnh số 99
4.5 Tổng kết chương 101

KẾT LUẬN VÀ KIẾN NGHỊ 103

DANH MỤC CÁC CÔNG TRÌNH CÓ LIÊN QUAN ĐẾN LUẬN ÁN 105

TÀI LIỆU THAM KHẢO 106

PHỤ LỤC 120

DANH MỤC CÁC CHỮ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
ACF	Autocorrelation Function	Hàm tự tương quan
ADC	Analog Digital Converter	Chuyển đổi tương tự sang số
AIC	Analog Analog to Information Converter	Chuyển đổi tương tự sang thông tin
BCH	Bose Chaudhuri Hocquenghem	Mã sửa lỗi vòng
BCS	Bayesian Compression Sensing	Lấy mẫu nén Bayesian
BP	Basic Pursuit	Thuật toán theo đuổi cơ sở
BPNSM	Bipolar Pseudorandom Numbers Sequence Matrix	Ma trận lưỡng cực giả ngẫu nhiên
CGP	Conjugate Gradient Pursuit	Tham lam theo Gradient liên hợp
CR	Compression ratio	Tỉ số nén
CS	Compressive Sensing	Lấy mẫu nén
CoSaMP	Compressed Sampling MP	Lấy mẫu nén đuổi khớp
DCT	Discrete Cosine Transform	Biến đổi cosin rời rạc
DFT	Discrete Fourier Transform	Biến đổi Fourier rời rạc
DMD	Digital Micro-Mirror Device	Mảng gương số
DP	Dynamic Programming	Quy hoạch động
DRMP	D-RIP Matching Pursuit	Thuật toán DRMP
DSS	Digital Signature Standard	Chữ ký số
DVC	Distributed Video Surveillance	Hệ thống video phân tán

DWT	Discrete Wavelet Transform	Biến đổi sóng con rời rạc
FPGA	Field Programmable Gate Array	Mảng logic khả trình
GP	Gradient Pursuit	Tham lam theo Gradient
IHT	Iterative Hard Thresholding	Ngưỡng lặp cứng
IRLS	Iteratively Reweighted Least Squares	Tái trọng số theo bình phương tối thiểu
IST	Iterative Soft Thresholding	Ngưỡng lặp mềm
LC	Linear Complexity	Độ phức tạp tuyến tính
LDPC	Low-density parity-check	Mã kiểm tra chẵn lẻ mật độ thấp
LFSR	Linear-Feedback Shift Register	Thanh ghi dịch phản hồi tuyến
LP	Linear Programming	Quy hoạch tuyến tính
MAE	Mean Absolute Error	Sai số trung bình tuyệt đối
MP	Matching Pursuit	Thuật toán đuổi khớp
MSE	Mean Squared Error	Sai số toàn phương trung bình
NSP	Null space conditions	Không gian vô hiệu
OMP	Orthogonal Matching Pursuit	Thuật toán đuổi khớp trực giao
OOC	Orthogonal Optical Codes	Mã quang học trực giao
ORLSMP	Order Recursive Least Square MP	Thuật toán đuổi khớp đối sánh đệ quy
PN	Pseudorandom Binary Numbers	Chuỗi giả ngẫu nhiên
PSNR	Peak Signal-to-Noise Ratio	Tỉ số tín hiệu cực đại trên nhiễu
RNG	Random Number Generated	Bộ tạo số ngẫu nhiên

RIP	Restricted Isometry Property	Tính chất giới hạn đẳng trị
SaMP	Sparsity Adaptive MP	Thuật toán đuổi khớp thích nghi
SNR	Signal-to-Noise Ratio	Tỉ số tín hiệu trên nhiễu
SSIM	Structural Similarity Index Measurement	Độ tương đồng về cấu trúc

DANH MỤC CÁC KÍ HIỆU

Ký hiệu	Ý nghĩa
i	Biến chỉ số
x	Tín hiệu thưa trong miền thời gian
$\hat{x}$	Tín hiệu thưa được khôi phục
$y[m]$	Tín hiệu lấy mẫu nén
Φ	Ma trận lấy mẫu nén
Ψ	Ma trận biểu diễn thưa
s	Vector có K phần tử khác 0
δ_K	Hằng số RIP bậc K
$\|s\|_p$	Chuẩn p của véc tơ s
$K - sparse$	Độ thưa K
$\mu(\Phi)$	Giá trị không kết hợp (*incoherent*) của ma trận Φ
I_p^T	Thứ tự lồng ghép dãy dịch pha
$\{b_n\}$	Chuỗi nhị phân phi tuyến
$Pr(.)$	Ký hiệu xác suất
σ	Tỉ lệ nhiễu cộng
∇	Giá trị Gradient
$GF(p)$	Trường hữu hãn
$Tr(\alpha)$	Hàm Vết
S_n	Trạng thái của thanh ghi dịch tại thời điểm n
$D[b_n]$	Biến đổi D của chuỗi b_n
R_{xy}	Tương quan giữa 2 tín hiệu x và y
$R_c(\tau)$	Hàm tự tương quan của chuỗi phi tuyến
$\mathbb{R}$	Tập số thực
$R_{a,b}(\tau)$	Tương quan chéo của chuỗi nhị phân a và b

$\mathcal{O}()$	Độ phức tạp tính toán của thuật toán
γ	Tỉ số suy giảm lỗi của thuật toán
C_{D_Γ}	Ma trận con xây dựng theo hướng tối đa $gradient$
w	Nhiễu cộng
E_x	Lỗi khôi phục
CR	Tỉ số nén trong lấy mẫu nén
cov	Giá trị hiệp phương sai
$\ell(x)$	Hàm mục tiêu
$P_{D_T}x$	Phép chiếu trực giao của vector x lên ma trận D_T

DANH MỤC CÁC BẢNG, BIỂU

Bảng 2.1 Các phần tử của $GF(2^3)$ 29
Bảng 2.2 Biến đổi D của chuỗi m 35

Bảng 3.1 Bảng so sánh thuật toán cải tiến và thuật toán gốc MP . . 68

Bảng 4.1 Bảng α^{Ti} 81
Bảng 4.2 Bảng $Tr(\alpha)$ 82
Bảng 4.3 Bảng PSNR(dB) trong trường hợp không cộng nhiễu . . . 92
Bảng 4.4 Bảng MSE trong trường hợp không cộng nhiễu 92
Bảng 4.5 Bảng thời gian xử lý trong trường hợp không cộng nhiễu . 93
Bảng 4.6 Bảng PSNR(dB) trong trường hợp cộng nhiễu 95
Bảng 4.7 Bảng MSE trong trường hợp cộng nhiễu 95
Bảng 4.8 Bảng thời gian xử lý trong trường hợp cộng nhiễu (giây) . 96

DANH MỤC CÁC HÌNH VẼ

Hình 1.1 Lấy mẫu truyền thống và lấy mẫu nén 6
Hình 1.2 Mô hình lấy mẫu nén [57] 7
Hình 1.3 Ma trận biểu diễn thưa [57] 8
Hình 1.4 Tính chất giới hạn đẳng trị RIP [19] 8
Hình 1.5 Biểu diễn tín hiệu trong miền (a) thời gian (b) tần số . . . 10
Hình 1.6 (a) Ảnh gốc (b) Ảnh biến đổi wavelet 10
Hình 1.7 Phân loại ma trận lấy mẫu nén 11
Hình 1.8 Phân loại thuật toán khôi phục 13
Hình 1.9 Tối thiểu hóa ℓ_1 [2] 14
Hình 1.10 Sơ đồ khối máy ảnh 1 pixel [99] 18

Hình 2.1 Các bước xây dựng ma trận BPNSM 27
Hình 2.2 LFSR phản hồi Fibonacci [42] 31
Hình 2.3 Hàm tự tương quan của chuỗi phi tuyến 40
Hình 2.4 Hàm tương quan chéo của chuỗi phi tuyến 40
Hình 2.5 Mô hình lấy mẫu nén băng rộng sử dụng ADC tốc độ thấp 49
Hình 2.6 Mô hình chuyển đổi từ byte trong bộ nhớ thành luồng bit 50
Hình 2.7 Lồng ghép các chuỗi dịch 51
Hình 2.8 Giản đồ xung đầu ra sau chuyển mạch 51

Hình 3.1 Lưu đồ thuật toán đuổi khớp 57
Hình 3.2 Lưu đồ thuật toán đuổi khớp trực giao 59
Hình 3.3 Lưu đồ thuật toán CoSaMP 62
Hình 3.4 Lưu đồ thuật toán DRMP 66

Hình 4.1 Mô hình lấy mẫu nén đề xuất 77
Hình 4.2 Phổ tần số sử dụng của Flycam Mavic pro 79

Hình 4.3 Phổ của chuỗi PN lồng ghép phi tuyến 82
Hình 4.4 Hàm tự tương quan của chuỗi PN lồng ghép phi tuyến . . 83
Hình 4.5 Hàm tương quan chéo của 2 chuỗi PN lồng ghép phi tuyến 83
Hình 4.6 Dạng tín hiệu của Flycam: 4.6a là tín hiệu gốc trong miền thời gian, 4.6b là biến đổi FFT của nó 84
Hình 4.7 Thời gian thực hiện trong trường hợp không cộng nhiễu . 85
Hình 4.8 Thời gian thực hiện trong trường hợp cộng nhiễu 85
Hình 4.9 Hệ số tương quan trong trường hợp không cộng nhiễu . . 86
Hình 4.10 Hệ số tương quan trong trường hợp cộng nhiễu 87
Hình 4.11 Hệ số MAE trong trường hợp không cộng nhiễu 88
Hình 4.12 Hệ số MAE trong trường hợp cộng nhiễu 88
Hình 4.13 Ảnh thử nghiệm 89
Hình 4.14 Ma trận zigzag 90
Hình 4.15 Đồ thị đánh giá 3 ma trận lấy mẫu 92
Hình 4.16 Ảnh khôi phục bằng thuật toán DRMP. 4.16a sử dụng ma trận Gauss, 4.16b sử dụng ma trận Bernoulli, 4.16c sử dụng ma trận BPNSM 93
Hình 4.17 Đồ thị đánh giá 3 ma trận lấy mẫu 94
Hình 4.18 Đồ thị đánh giá 3 thuật toán lấy mẫu 95
Hình 4.19 Ảnh khôi phục bằng thuật toán DRMP trong trường hợp cộng thêm nhiễu. 4.19a sử dụng ma trận Gauss, 4.19b sử dụng ma trận Bernoulli, 4.19c sử dụng ma trận BPNSM 97
Hình 4.20 Đồ thị đánh giá 3 ma trận lấy mẫu trong trường hợp cộng nhiễu 97
Hình 4.21 Đồ thị đánh giá 3 thuật toán lấy mẫu trong trường hợp cộng nhiễu 98
Hình 4.22 Sơ đồ khối hệ thống thu tín hiệu vô tuyến từ Flycam . . . 99
Hình 4.23 Kiến trúc của hệ thống giám sát video phân tán [53] . . . 100
Hình 4.24 Hình ảnh giám sát 101
Hình 4.25 Phần sai khác giữa 2 ảnh 101

MỞ ĐẦU

Định lý lấy mẫu của Nyquist-Shannon phát biểu rằng để không mất thông tin và có thể khôi phục lại hoàn toàn tín hiệu thì phải lấy mẫu tín hiệu với tần số lấy mẫu cao hơn ít nhất hai lần băng thông của tín hiệu. Trên thực tế, nguyên tắc này làm nền tảng cho gần như tất cả các phương thức chuyển đổi tín hiệu được sử dụng trong các thiết bị điện tử âm thanh và hình ảnh, thiết bị hình ảnh y tế, máy thu radio. Trong nhiều ứng dụng như trong ảnh số và âm thanh số, tốc độ lấy mẫu Nyquist là cao và thu được quá nhiều mẫu, do đó cần phải có quá trình nén tín hiệu để có thể phù hợp với việc lưu trữ, xử lý hoặc truyền đi xa. Hay trong các ứng dụng khác, như trong hệ thống siêu cao tần, các ứng dụng này đòi hỏi phải lấy mẫu tín hiệu ở tần số rất cao nếu tuân theo định lý Nyquist. Điều này dẫn đến yêu cầu phải có các bộ chuyển đổi ADC tốc độ rất cao, hệ thống lưu trữ, xử lý dữ liệu phức tạp gây ra nhiều khó khăn trong chế tạo, và giá thành thiết bị trở nên rất đắt.

Trong những năm gần đây, lĩnh vực viễn thông và công nghệ thông tin phát triển một cách nhanh chóng, lượng thông tin được trao đổi ngày càng nhiều dẫn đến hạ tầng viễn thông và công nghệ thông tin luôn phải đổi mới và nâng cấp để có thể đáp ứng những nhu cầu trao đổi thông tin của người dùng. Ngoài ra, trong lĩnh vực y tế việc lấy mẫu nén cũng được quan tâm nghiên cứu và cho nhiều kết quả khả quan [44], [46], [62], [108]. Các nghiên cứu gần đây cho thấy lấy mẫu nén (CS) đang được coi như một ứng cử viên hứa hẹn để giải quyết các vấn đề trên [52], [89], [102] trong đó, số lượng mẫu cần lấy để có thể khôi phục được tín hiệu có thể ít hơn nhiều so với khi lấy mẫu tuân theo định lý Nyquist-Shannon.

Các nghiên cứu hiện nay tập trung vào việc thiết kế ma trận lấy mẫu nén và phát triển thuật toán khôi phục lại tín hiệu từ các mẫu nén một cách hiệu

quả. Các ma trận trong CS phải thỏa mãn tính chất giới hạn đẳng trị RIP, yêu cầu số lượng phép đo nhỏ (số hàng của ma trận lấy mẫu là nhỏ), sai số khôi phục và thời gian thực hiện nhỏ. Để thiết kế các ma trận như vậy là một nhiệm vụ khó khăn vì nó phải đáp ứng các mục tiêu trái ngược nhau. Chúng ta cần phải thu thập nhiều phép đo hơn (số hàng của ma trận lấy mẫu phải lớn) để giảm thiểu sai số trong quá trình khôi phục lại tín hiệu được lấy mẫu nén nhưng việc tăng số phép đo sẽ làm số mẫu phải lấy tăng lên trái với mục tiêu giảm số mẫu phải lấy của CS, và do đó làm tăng thời gian xử lý. Một khó khăn khác gặp phải trong quá trình thiết kế ma trận lấy mẫu nén là đảm bảo tính phổ quát của nó. Các ma trận ngẫu nhiên là phổ quát, nhưng vì không có các thuật toán tính toán nhanh cho các ma trận này dẫn đến quá trình khôi phục tiêu tốn nhiều thời gian tính toán xử lý, do đó ma trận Gauss hoặc các ma trận phi cấu trúc khác không thực tế cho các bài toán có quy mô lớn. Để cải thiện tốc độ tính toán, người ta có thể sử dụng các ma trận ngẫu nhiên có cấu trúc để tăng tốc độ cho quá trình khôi phục nhưng phải đánh đổi là mất tính phổ quát. Ngoài ra, việc xác định một ma trận thỏa mãn tính chất RIP là rất khó khăn [7], [70], [82]. Để giải quyết vấn đề này các nghiên cứu hiện nay tập trung vào thiết kế các ma trận xác định [43], [76], [94]. Các ưu điểm của ma trận xác định là: có cấu trúc xác định, quá trình lấy mẫu đơn giản, có yêu cầu về lưu trữ nhỏ.

Bên cạnh việc thiết kế ma trận lấy mẫu nén thì một vấn đề quan trọng nữa là xây dựng các thuật toán khôi phục tín hiệu được lấy mẫu nén. Hầu hết các nghiên cứu hiện nay tập trung vào xây dựng các thuật toán có cấu trúc ổn định, độ phức tạp tính toán thấp và nâng cao độ chính xác của tín hiệu được khôi phục. Trong số đó, các thuật toán khôi phục tham lam dựa trên thuật toán gốc là thuật toán đuổi khớp (MP) được sử dụng rộng rãi vì tính đơn giản và hiệu quả vượt trội [57], [81].

Với mục đích kết hợp các ưu điểm của ma trận xác định và thuật toán tham lam trong việc thực hiện nhanh, lưu trữ ít phù hợp với các ứng dụng yêu cầu thời gian thực, độ phức tạp phần cứng thấp, nghiên cứu sinh đã lựa

chọn đề tài: ***"Nghiên cứu thiết kế ma trận và cải tiến thuật toán khôi phục tín hiệu được lấy mẫu nén"*** cho luận án nghiên cứu của mình. Theo đó, đối tượng nghiên cứu của luận án là các ma trận lấy mẫu nén và thuật toán khôi phục tín hiệu được lấy mẫu nén. Việc thiết kế ma trận lấy mẫu, cải tiến thuật toán khôi phục nhằm nâng cao tốc độ, giảm yêu cầu lưu trữ, tính toán trong khi vẫn đảm bảo độ chính xác của tín hiệu được khôi phục là hết sức cấp thiết.

Ý nghĩa khoa học, thực tiễn

Ý nghĩa khoa học của luận án là đề xuất mô hình lấy mẫu nén với ma trận xác định được thiết kế cùng với thuật toán khôi phục được cải tiến, xây dựng chương trình tính toán và mô phỏng để đánh giá hiệu năng của mô hình lấy mẫu nén đề xuất. Ý nghĩa thực tiễn của luận án kỳ vọng thể hiện ở việc ma trận lấy mẫu và thuật toán khôi phục cải tiến được đề xuất trong luận án có thể áp dụng vào thực tế để nâng cao tốc độ, giảm yêu cầu lưu trữ trong khi vẫn đảm bảo độ chính xác của tín hiệu lấy mẫu nén được khôi phục.

Mục tiêu

Mục tiêu đầu tiên của luận án là đề xuất được một ma trận lấy mẫu nén thỏa mãn tiêu chí giới hạn đẳng trị RIP, có tính bảo mật cao đối với tín hiệu cần lấy mẫu, khả thi khi triển khai trên các hệ thống điện tử số. Mục tiêu thứ 2 là cải tiến một thuật toán khôi phục đảm bảo độ chính xác và các yêu cầu về thời gian tính toán. Mục tiêu thứ 3 là xây dựng phần mềm, công cụ để tiến hành phân tích đánh giá ma trận và thuật toán được đề xuất thông qua mô phỏng.

Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu của luận án là các ma trận lấy mẫu nén, các kỹ thuật thiết kế ma trận lấy mẫu, các thuật toán khôi phục tín hiệu được lấy mẫu nén, các chỉ tiêu đánh giá hiệu năng của mô hình lấy mẫu nén.

Phạm vi nghiên cứu của đề tài luận án là các tín hiệu có độ dài hữu hạn, các ma trận lấy mẫu nén xác định và các thuật toán tham lam.

Phương pháp nghiên cứu

Phương pháp nghiên cứu của luận án là nghiên cứu lý thuyết kết hợp với tính toán và mô phỏng trên máy tính để chứng tỏ hiệu quả của ma trận lấy mẫu nén thiết kế và khả năng áp dụng thuật toán cải tiến trong các trường hợp khác nhau.

Phương pháp nghiên cứu lý thuyết: sử dụng lý thuyết thông tin, đại số tuyến tính và các công cụ toán học để thiết kế một ma trận lấy mẫu nén xác định, phân tích mô hình toán học của thuật toán gốc MP để đưa ra thuật toán cải tiến DRMP.

Phương pháp nghiên cứu mô phỏng: xây dựng chương trình mô phỏng sử dụng các công cụ và thư viện ngôn ngữ lập trình Python theo kịch bản của ma trận và thuật toán đã được thiết kế. Trên cơ sở đó, đánh giá được ưu và nhược điểm của ma trận lấy mẫu nén và thuật toán cải tiến được đề xuất trong luận án.

Cấu trúc của luận án

Với lý do, mục tiêu nghiên cứu, phương pháp và kết quả đạt được, luận án được bố cục gồm 4 chương với các nội dung sau:

Chương 1 *"Tổng quan về lấy mẫu nén"* trình bày tổng quan về lấy mẫu nén và cơ sở lý thuyết trong lĩnh vực lấy mẫu nén. Nội dung chương cũng khảo

sát đánh giá các nghiên cứu về lấy mẫu nén bao gồm việc thiết kế ma trận lấy mẫu và thuật toán khôi phục. Từ các điểm còn tồn tại của các nghiên cứu đó để đề xuất các hướng nghiên cứu của luận án.

Chương 2 *"Thiết kế ma trận lấy mẫu nén xác định"* trình bày về các phương pháp thiết kế, mô hình toán học của ma trận lấy mẫu nén xác định. Đóng góp của luận án trong chương này là đề xuất một ma trận lấy mẫu nén xác định được thiết kế từ các chuỗi nhị phân phi tuyến giả ngẫu nhiên tốc độ cao. Kết quả nghiên cứu của chương 2 được trình bày trong [C1].

Chương 3 *"Đề xuất thuật toán khôi phục tín hiệu được lấy mẫu nén DRMP"*. Nội dung trình bày một thuật toán khôi phục tín hiệu được lấy mẫu nén dựa trên thuật toán gốc MP. Đồng thời cũng chứng minh thuật toán cải tiến giảm lỗi xảy ra ở mỗi bước trong quá trình khôi phục. Kết quả nghiên cứu được trình bày trong [J1].

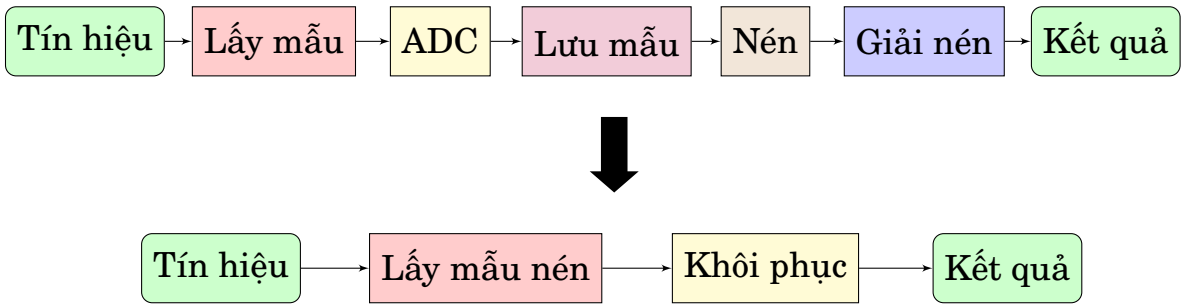
Chương 4 *"Đề xuất mô hình lấy mẫu nén"* trình bày quá trình mô phỏng, đánh giá hiệu năng của ma trận và thuật toán khôi phục được đề xuất trên 02 ứng dụng cụ thể với đầu vào là các tín hiệu phổ biến bao gồm tín hiệu vô tuyến và tín hiệu ảnh đa cấp xám. Kết quả nghiên cứu của chương đã được công bố trong [J2], [J3], [J4].

CHƯƠNG 1. TỔNG QUAN VỀ LẤY MẪU NÉN

Nội dung chương giới thiệu mô hình lấy mẫu nén và cơ sở toán học cơ bản của nó. Ngoài ra, các tiêu chí và các tham số đánh giá hiệu năng của mô hình lấy mẫu nén cũng được trình bày trong chương. Tình hình nghiên cứu trong và ngoài nước liên quan đến lĩnh vực lấy mẫu nén cũng được khảo sát theo các hướng khác nhau. Trên cơ sở các hướng nghiên cứu đó, các hướng nghiên cứu của luận án cũng được đề xuất.

1.1. Mô hình lấy mẫu nén

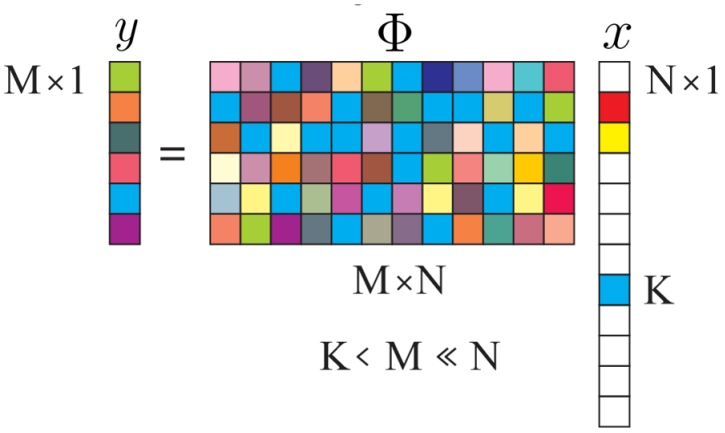
Lấy mẫu nén là một phương pháp thu nhận và xử lý tín hiệu tiên tiến được đề xuất bởi Candès và Donoho [15], [31]. Đối với phương pháp lấy mẫu truyền thống, tín hiệu được lấy mẫu bằng với tốc độ Nyquist, trong khi đó với phương pháp lấy mẫu nén, tín hiệu được lấy mẫu dưới tốc độ Nyquist.



Hình 1.1: *Lấy mẫu truyền thống và lấy mẫu nén*

Điều này có thể thực hiện được bởi tín hiệu được chuyển đổi sang một miền mà trong đó có biểu diễn thưa và được thực hiện lấy mẫu nén thông qua ma trận lấy mẫu. Sau đó, tín hiệu được tái tạo lại từ các mẫu nén bằng cách sử dụng một kỹ thuật tối ưu hóa. Hình 1.1 là sơ đồ khối minh họa sự khác biệt giữa phương pháp thu tín hiệu truyền thống và phương pháp CS. Đối

với phương pháp lấy mẫu truyền thống, tín hiệu được lấy mẫu với tốc độ lấy mẫu Nyquist, sau đó, các mẫu được lưu lại và quá trình nén tiếp theo sau sẽ loại bỏ đi phần lớn các tín hiệu dư thừa. Trong khi đó, đối với phương pháp CS tín hiệu được lấy mẫu ở dạng đã được nén. Do đó, có thể giảm bớt nhiều khâu cả phần cứng và phần mềm trong quá trình lấy mẫu.



Hình 1.2: *Mô hình lấy mẫu nén [57]*

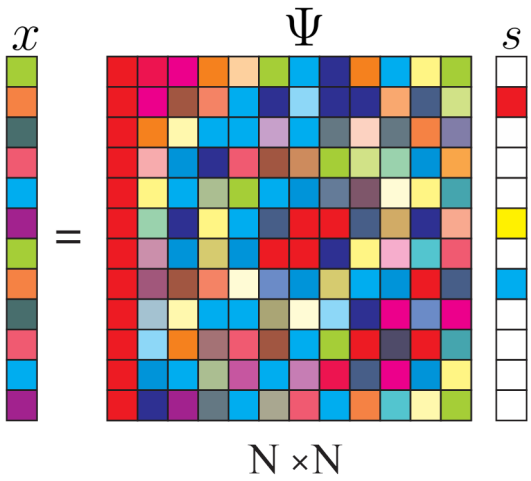
Mô hình tổng quan về CS được thể hiện trong hình 1.2. Một tín hiệu rời rạc giả định $x_{N\times 1} \in R^N$ được biến đổi thành $y_{M\times 1} \in R^M$ bởi ma trận $\Phi_{M\times N}$. Quá trình lấy mẫu nén có thể được biểu diễn như sau:

$$y = \Phi x, \tag{1.1}$$

trong đó $M < N$, và Φ được gọi là ma trận lấy mẫu. Từ biểu thức (1.1) tín hiệu $x_{N\times 1}$ được nén thành tín hiệu $y_{M\times 1}$ và không thể tìm lại được x từ biểu thức (1.1) bởi số ẩn nhiều hơn số phương trình. Điều kiện tiên quyết để có thể tìm được x là x phải thưa hoặc x thưa trên một số cơ sở trực giao, nghĩa là,

$$x = \Psi s, \tag{1.2}$$

trong đó Ψ là một ma trận trực giao có kích thước $N \times N$ như được biểu diễn trong hình 1.3 mà thỏa mãn điều kiện $\Psi\Psi^T = I$ và $\Psi^T\Psi = I$. Ở đây, Ψ là ma trận thưa và s là vector thưa. Khi đó K là số giá trị mà ở đó $s \neq 0$, $N - K$ giá trị còn lại $s = 0$ và $K \ll N$, gọi vector s là vector có độ thưa K ($K - sparse$). Các



Hình 1.3: *Ma trận biểu diễn thưa [57]*

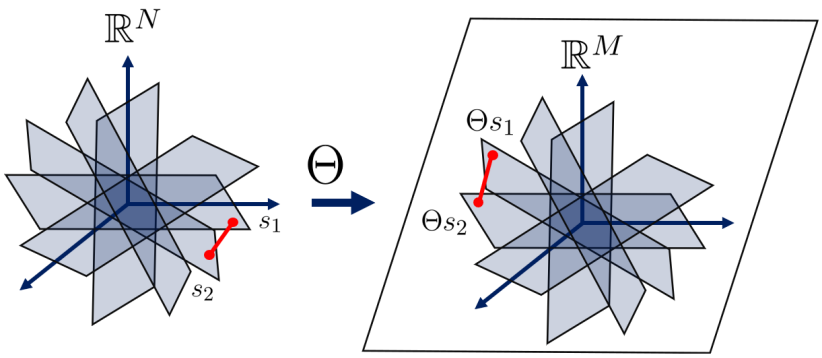
ma trận thưa phổ biến được biết đến như ma trận biến đổi Fourier rời rạc (DFT) [12], biến đổi Wavelet rời rạc (DWT) [38], và ma trận biến đổi Cosine rời rạc (DCT) [54]. Từ biểu thức (1.1) và (1.2) có

$$y = \Phi x = \Phi\Psi s = \Theta s, \tag{1.3}$$

ở đây, $\Phi\Psi$ là ma trận lấy mẫu nén. Để khôi phục x từ y, ma trận lấy mẫu $\Phi\Psi$ phải thỏa mãn tính chất giới hạn đẳng trị RIP [11], [87] như được mô tả trong hình 1.4. Hằng số RIP δ_K bậc K đối với ma trận Θ là

$$(1-\delta_K)\|s\|_2 \leq \|\Theta s\|_2 \leq (1+\delta_K)\|s\|_2, \tag{1.4}$$

trong đó $\delta_K \in (0,1)$. Quá trình khôi phục lại tín hiệu thưa được lấy mẫu nén



Hình 1.4: *Tính chất giới hạn đẳng trị RIP [19]*

có thể biểu diễn như sau

$$\min_{\hat{s}} \|\hat{s}\|_{\ell_1} \quad \text{trong đó} \quad y = \Theta\hat{s}, \tag{1.5}$$

việc khôi phục lại tín hiệu được lấy mẫu nén là một bài toán tối ưu hóa lồi.

1.1.1. Tín hiệu thưa

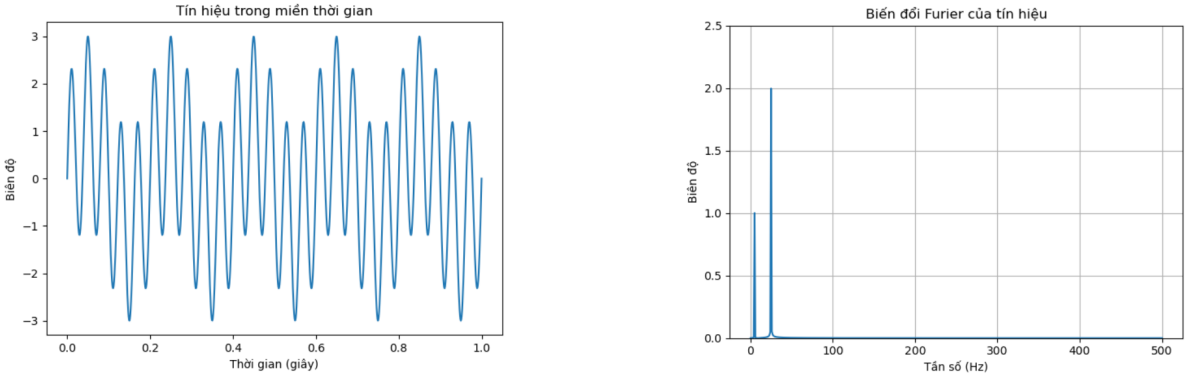
Việc chuyển đổi tín hiệu sang một hệ cơ sở mới phù hợp có thể giúp thu được một biểu diễn ngắn gọn hơn so với tín hiệu gốc. Trong một số ứng dụng việc lưu trữ và truyền dữ liệu là rất tốn kém. Do đó, việc lưu trữ và truyền đi các hệ số trong hệ cơ sở mà có biểu diễn ngắn gọn của tín hiệu sẽ thuận lợi hơn nhiều so với việc xử lý với tín hiệu gốc. Trong trường hợp số lượng các hệ số khác 0 của tín hiệu là nhỏ, khi đó nó được gọi là một tín hiệu có biểu diễn thưa. Đối với tín hiệu thưa, trong đa số các trường hợp chỉ cần lưu trữ hoặc truyền đi các hệ số khác 0, phần còn lại của tín hiệu có thể được giả định bằng 0 [87], [88].

Về mặt toán học, có thể gọi tín hiệu x là $K - sparse$ (x có độ thưa K) khi nó có nhiều nhất K phần tử khác 0, tức là $\|x\|_0 \leq K$. Có

$$\sum\nolimits_K = \{x : \|x\|_0 \leq K\}, \tag{1.6}$$

là biểu thị tập hợp tất cả các tín hiệu có $K - sparse$. Tín hiệu trong thực tế thông thường không có biểu diễn thưa trong hệ cơ sở của nó nhưng có thể biểu diễn thông qua các vector thưa của một hệ cơ sở Ψ. Trong trường hợp này, x vẫn được xem là $K - sparse$, và có thể biểu diễn x dưới dạng $x = \Psi s$ trong đó $\|s\|_0 \leq K$ [31].

Hình 1.5a biểu diễn một tín hiệu được tổng hợp từ 2 tín hiệu hình sin có tần số khác nhau trong miền thời gian. Tín hiệu này trong miền thời gian có nhiều điểm khác 0, hình 1.5b biểu diễn biến đổi Furier của nó và chỉ yêu cầu 2 vector cơ sở khác 0 để biểu diễn nó. Do đó, có thể coi tín hiệu ban đầu là thưa trong miền tần số.



(a) ***(b)***

Hình 1.5: *Biểu diễn tín hiệu trong miền (a) thời gian (b) tần số*

Tính chất thưa của tín hiệu cũng được biểu hiện đối với các tín hiệu 2-D. Ví dụ, một bức ảnh nhị phân chụp bầu trời vào ban đêm là thưa trong miền điểm ảnh vì phần lớn các điểm ảnh có màu đen và có giá trị bằng không. Tương tự như vậy, các bức ảnh thông thường cũng có các vùng ảnh với số điểm ảnh tương đồng nhau đặc biệt là đối với các bức ảnh y tế [50]. Tín hiệu ảnh thể hiện tính chất thưa khi được biểu diễn trong miền wavelet [66]. Hình 1.6 minh họa về một bức ảnh được biến đổi wavelet, hầu hết các hệ số đều rất nhỏ. Do đó, có thể có đặt các hệ số có giá trị nhỏ bằng 0 và thu được một tín hiệu thưa có $K - sparse$.

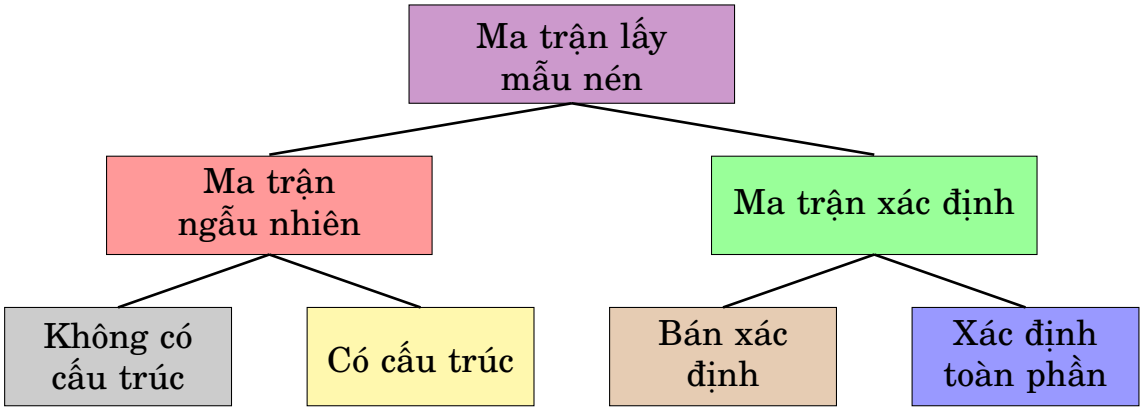
(a) ***(b)***

Hình 1.6: *(a) Ảnh gốc (b) Ảnh biến đổi wavelet*

Các tín hiệu thưa hoặc thưa trên một hệ cơ sở trực giao là phổ biến trong tự nhiên. Do đó, phương pháp CS rất có tiềm năng khi được ứng dụng với các tín hiệu thưa trong thực tế [34], [50], [88].

1.1.2. Ma trận lấy mẫu nén

Lấy mẫu nén bao gồm ba quá trình chính, biểu diễn tín hiệu thưa, lấy mẫu tín hiệu dựa trên ma trận lấy mẫu, khôi phục tín hiệu được lấy mẫu nén [31]. Ma trận lấy mẫu đóng vai trò quan trọng đến độ chính xác và thời gian xử lý của quá trình khôi phục lại tín hiệu được lấy mẫu nén. Trong thập kỷ qua, các nghiên cứu về ma trận lấy mẫu nén đã được công bố và có thể phân thành 2 nhóm chính là ma trận ngẫu nhiên và ma trận xác định [7] như được liệt kê trong hình 1.7.



Hình 1.7: *Phân loại ma trận lấy mẫu nén*

a. Ma trận ngẫu nhiên

Ma trận ngẫu nhiên có 2 loại là ma trận ngẫu nhiên không có cấu trúc và ma trận ngẫu nhiên có cấu trúc. Ma trận ngẫu nhiên không có cấu trúc với các phần tử được tạo ra ngẫu nhiên theo một phân bố xác suất như Gauss và Bernoulli [9]. Một ma trận ngẫu nhiên có kích thước $N \times N$ được tạo thành, sau đó từ M hàng của ma trận ban đầu sẽ được chọn ngẫu nhiên để tạo thành ma trận lấy mẫu nén. Các ma trận dạng này có ưu điểm dễ xây dựng và đáp

ứng tiêu chí RIP với xác suất cao. Tuy nhiên, chúng có một số hạn chế trong thực tế bởi các phần tử trong ma trận là các số thực dấu phảy động nên không khả thi với các bài toán quy mô lớn do khối lượng tính toán và cần bộ nhớ để lưu trữ lớn.

Loại ma trận ngẫu nhiên có cấu trúc với các phần tử được tạo thành từ một hàm hoặc một cấu trúc nhất định. Sau đó các hàng được lựa chọn ngẫu nhiên từ các cấu trúc ban đầu để tạo ra ma trận lấy mẫu nén. Các ví dụ điển hình của loại ma trận này là các ma trận con được tạo thành từ ma trận Fourier [103] và ma trận Hadamard [86]. Các ma trận loại này có ưu điểm làm tăng tốc trong quá trình khôi phục lại tín hiệu. Tuy nhiên, chúng có nhược điểm là không ổn định, lỗi khôi phục cao và yêu cầu số hàng của ma trận lấy mẫu lớn.

b. Ma trận xác định

Ma trận lấy mẫu xác định là ma trận được thiết kế theo các cấu trúc xác định và đáp ứng tiêu chí RIP hoặc tính chất không kết hợp (*incoherent*). Một số ma trận lấy mẫu xác định đã được đề xuất để giải quyết các vấn đề của ma trận ngẫu nhiên [5], [59], [60], [107]. Ma trận xác định được phân thành 2 loại là ma trận bán xác định và ma trận xác định toàn phần.

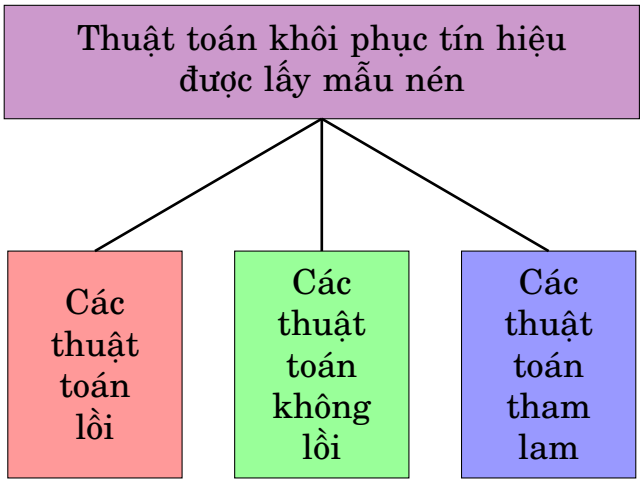
Các ma trận bán xác định thường được tạo thành qua 2 bước. Bước thứ nhất là tạo cột đầu tiên với các phần tử ngẫu nhiên và bước thứ hai là tạo ma trận đầy đủ bằng cách áp dụng một phép biến đổi đơn giản trên cột đầu tiên, chẳng hạn như phép quay để tạo ra từng hàng của ma trận. Ví dụ điển hình về ma trận loại này là ma trận Circulant và Toeplitz [83]. Các ma trận dạng này có ưu điểm dễ xây dựng, giảm tính ngẫu nhiên và sử dụng ít bộ nhớ hơn so với ma trận ngẫu nhiên không có cấu trúc. Các ma trận này không thực sự phổ biến và chỉ được áp dụng trong một số ứng dụng cụ thể.

Ma trận loại xác định toàn phần là ma trận có cấu trúc xác định hoàn toàn nhằm thỏa mãn tiêu chí RIP hoặc tính chất không kết hợp. Các ví dụ điển

hình về loại ma trận có cấu trúc xác định toàn phần là ma trận cấu tạo từ mã nhị phân tuần hoàn BCH [7] và ma trận cấu tạo từ mã không tuần hoàn *chirp* [84]. Các ma trận xác định toàn phần có ưu điểm là thời gian thực hiện nhanh, đơn giản trong quá trình lấy mẫu, giảm độ phức tạp tính toán. Tuy nhiên, chúng có nhược điểm là kích thước ma trận không thể lựa chọn tùy ý, khó khăn để xác định tiêu chí RIP.

1.1.3. Thuật toán khôi phục

Vấn đề khôi phục tín hiệu được lấy mẫu nén là tìm các phương pháp để giải quyết bài toán hệ phương trình tuyến tính với số ẩn nhiều hơn số phương trình trong biểu thức (1.3). Điều kiện cần và đủ để giải quyết bài toán này là tín hiệu lấy mẫu phải đủ thưa và ma trận lấy mẫu thỏa mãn tính chất RIP [39]. Trong những năm gần đây một số các thuật toán đã được đề xuất [6], [64], [65] trong đó có thể phân loại thành 3 nhóm chính như được thể hiện trong hình 1.8 [6].



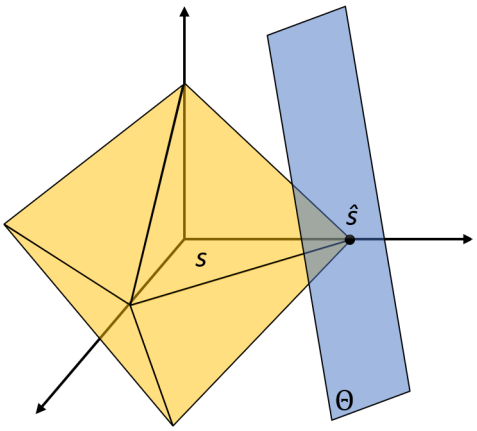
Hình 1.8: *Phân loại thuật toán khôi phục*

a. Các thuật toán lồi

Trong thực tế không thể khôi phục lại chính xác s mà chỉ tìm được một ước lượng $\hat{s}$ gần đúng so với s. Mục tiêu của các thuật toán khôi phục tín hiệu

được lấy mẫu nén dạng hàm lồi là giải bài toán (1.3) bằng cách tìm kiếm theo các giá trị tối thiểu hóa ℓ_1 trên không gian nghiệm. Thuật toán Theo đuổi cơ sở (BP) [104] là một đại diện điển hình của thuật toán hàm lồi và nó được sử dụng để giải quyết vấn đề tối ưu hóa cho biểu thức dưới đây dựa trên phương pháp quy hoạch tuyến tính (LP)

$$\hat{s} = \arg\min \|s\|_1 \text{ với } y = \Theta s. \tag{1.7}$$



Hình 1.9: *Tối thiểu hóa ℓ_1 [2]*

Vì định chuẩn ℓ_1 $\|\cdot\|_1$ là một hàm lồi nên phương trình (1.7) là một bài toán tối ưu hóa lồi. Cần phải lựa chọn số phép đo M tối thiểu để có thể khôi phục lại chính xác tín hiệu có độ thưa K với một ma trận lấy mẫu cố định cho trước. Độ phức tạp tính toán khi giải phương trình (1.7) là $(\mathcal{O}(N^3))$ về mặt tính toán rất tốn kém và không thích hợp cho các ứng dụng thời gian thực. Một số các nghiên cứu đã cải tiến thuật toán tối ưu hóa ℓ_1 để giảm độ phức tạp tính toán bằng việc đưa vào các điều kiện ràng buộc đối với thuật toán [20].

b. Các thuật toán không lồi

Các thuật toán khôi phục không lồi như thuật toán tối thiểu hóa ℓ_p với $(0 < p < 1)$ đã được đề xuất để tăng độ thưa K của tín hiệu được lấy mẫu nén. Tối thiểu hóa ℓ_p cho phép khôi phục lại tín hiệu thưa với số phép đo ít hơn so với tối thiểu hóa lồi ℓ_1 nhưng có độ phức tạp tính toán cao hơn [90], [106]. Thuật toán khôi phục IRLS là điển hình cho các dạng thuật toán không lồi [24].

c. Các thuật toán tham lam

Một kiểu tiếp cận khác trong khôi phục tín hiệu được lấy mẫu nén là các thuật toán tham lam dựa trên quy hoạch động (DP), nhằm xác định các thành phần biểu diễn của tín hiệu theo từng lần lặp và kết thúc quá trình dựa trên việc thiết lập các điều kiện dừng. Các thuật toán dựa trên DP được áp dụng phổ biến hơn các thuật toán dựa trên LP vì các thuật toán dựa trên DP có thể dễ dàng thực hiện và có độ phức tạp tính toán thấp. Một số thuật toán tham lam điển hình như thuật toán đuổi khớp (MP) [63] và thuật toán đuổi khớp trực giao (OMP) [96], thuật toán đuổi khớp đệ qui bình phương tối thiểu (ORLSMP) [1], thuật toán lấy mẫu nén đuổi khớp (CoSaMP) [72] và đuổi khớp thích nghi (SaMP) [29]. Luận án này tập trung nghiên cứu các thuật toán tham lam do tính phổ biến và tính khả thi cao khi ứng dụng vào trong thực tế.

1.2. Hiệu năng của mô hình lấy mẫu nén

Hiệu năng là vấn đề rất quan trọng để đánh giá hiệu quả của một mô hình. Có rất nhiều các tham số để đánh giá hiệu năng của mô hình lấy mẫu nén. Trong đó các tham số thường sử dụng như thời gian thực hiện là khoảng thời gian được tính từ khi bắt đầu lấy mẫu đến khi khôi phục thành công, độ phức tạp tính toán, tỉ số nén, lỗi khôi phục. Tuy nhiên, việc lựa chọn tham số nào

để đánh giá tùy thuộc vào đặc điểm và ứng dụng của từng hệ thống. Trong luận án, với mô hình lấy mẫu nén dựa trên ma trận được thiết kế và thuật toán được cải tiến, các tham số được sử dụng là thời gian thực hiện, hệ số tương quan và sai số trung bình tuyệt đối đối với tín hiệu mô phỏng 1 chiều. Đối với tín hiệu mô phỏng là ảnh 2 chiều luận án sử dụng tham số thời gian thực hiện, tham số PSNR và tham số MSE để đánh giá hiệu năng của mô hình.

a. Hệ số tương quan

Hệ số tương quan Pearson được sử dụng để đánh giá độ tương quan của tín hiệu sau khi khôi phục và tín hiệu gốc trước khi thực hiện lấy mẫu nén theo biểu thức sau:

$$R_{xy} = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N} (x_i - \overline{x})^2 \sum_{i=1}^{N} (y_i - \overline{y})^2}}. \tag{1.8}$$

Trong đó, N là số mẫu, x_i, y_i là các mẫu tương ứng với $i \in \{0, 1, ...N\}$, $\bar{x}, \bar{y}$ là trung bình mẫu. Hệ số tương quan Pearson (R_{xy}) có giá trị dao động trong khoảng liên tục từ -1 đến $+1$:

- $R_{xy} = 0$: Hai biến không có tương quan tuyến tính.
- $R_{xy} = 1; R_{xy} = -1$: Hai biến có mối tương quan tuyến tính tuyệt đối.
- $R_{xy} < 0$: Hệ số tương quan âm. Nghĩa là giá trị biến x tăng thì giá trị biến y giảm và ngược lại, giá trị biến y tăng thì giá trị biến x giảm.
- $R_{xy} > 0$: Hệ số tương quan dương. Nghĩa là giá trị biến x tăng thì giá trị biến y tăng và ngược lại, giá trị biến y tăng thì giá trị biến x cũng tăng.

b. Sai số trung bình tuyệt đối MAE

Là một phương pháp đo lường sự khác biệt giữa hai biến liên tục. Giả sử rằng X và $\hat{X}$ là hai tín hiệu 1 chiều có kích thước $1 \times N$, trong đó X là tín hiệu gốc và $\hat{X}$ thể hiện tín hiệu khôi phục được từ thuật toán. Hệ số MAE được tính theo công thức sau:

$$MAE = \frac{\sum_{i=1}^{N} |\hat{x}_i - x_i|}{N} = \frac{\sum_{i=1}^{N} |e_i|}{N}. \tag{1.9}$$

c. Sai số toàn phương trung bình MSE

Để đánh giá sai số của tín hiệu ảnh 2 chiều có kích thước $M \times N$ được khôi phục so với tín hiệu ảnh gốc. Giá trị MSE được đưa ra bởi công thức:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [X(i,j) - \hat{X}(i,j)]^2, \tag{1.10}$$

trong đó X là ảnh gốc và $\hat{X}$ là ảnh được khôi phục.

d. Tham số PSNR

PSNR được sử dụng để đánh giá chất lượng tín hiệu khôi phục của các thuật toán nén có mất mát dữ liệu (ví dụ: dùng trong nén ảnh). Tín hiệu trong trường hợp này là dữ liệu gốc, và nhiễu là các lỗi xuất hiện khi nén. Giá trị PSNR được đưa ra bởi công thức:

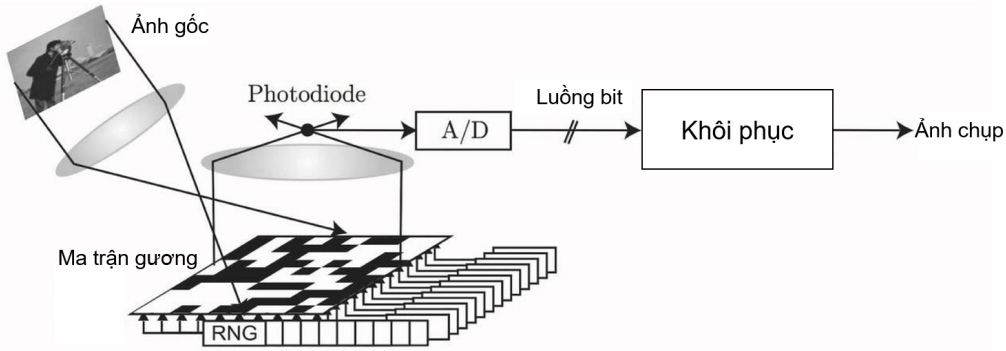
$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right), \tag{1.11}$$

trong đó MAX_I là giá trị tối đa của một điểm ảnh, ví dụ trong trường hợp một điểm ảnh được mã hóa bởi 8 bit, giá trị $MAX_I = 255$.

1.3. Các công trình nghiên cứu liên quan

Theo tìm hiểu của nghiên cứu sinh, ở Việt Nam hiện tại các nghiên cứu về lấy mẫu nén chưa được quan tâm nhiều. Các công trình nghiên cứu liên quan đến lấy mẫu nén mới chỉ nghiên cứu một số ứng dụng của lấy mẫu nén như trong một số nghiên cứu [41], [58], [77], [78].

Trên thế giới, các nghiên cứu và ứng dụng của lấy mẫu nén đã được triển khai. Vào khoảng năm 2004, Emmanuel Candès, Justin Romberg, Terence Tao và David Donoho đã chứng minh rằng nếu biết một tín hiệu là thưa thì tín hiệu đó có thể được khôi phục lại với số mẫu ít hơn so với lấy mẫu với tần số Nyquist [15], [31]. Một trong những ứng dụng nổi tiếng trong lĩnh vực lấy mẫu nén và được trích dẫn bởi rất nhiều bài báo là hệ thống máy ảnh 1 pixel như được thể hiện trong hình 1.10 được phát triển tại Đại học Rice [99] và được chọn là một trong 10 công nghệ mới nổi bật của tạp chí MIT Technology Review [67] vào năm 2007. Ý tưởng của máy ảnh 1 pixel là đổi độ phân giải không gian thành độ phân giải theo thời gian. Đối với các máy ảnh sử dụng ở các tần số đặc biệt như chụp ảnh hồng ngoại giá thành cho mỗi pixel cực kỳ đắt đỏ, đây là một sự đánh đổi đem lại lợi ích kinh tế lớn.



Hình 1.10: *Sơ đồ khối máy ảnh 1 pixel [99]*

Các kỹ thuật lấy mẫu nén cũng rất được chú trọng trong xử lý ảnh y tế, kết quả nghiên cứu của Lustig [61] đã được áp dụng trong viện nghiên cứu y học của trường Stanford. Trong lĩnh vực viễn thông phương pháp lấy mẫu

nén cũng được sử dụng trong một số ứng dụng như ra-đa [45], cảm thụ phổ vô tuyến băng rộng [105]. Các hướng nghiên cứu chính hiện nay về lĩnh vực lấy mẫu nén bao gồm:

- Nghiên cứu thiết kế ma trận lấy mẫu nén nhằm nâng cao hiệu suất nén, hỗ trợ quá trình tính toán, khôi phục lại tín hiệu thưa.
- Nghiên cứu thuật toán khôi phục tín hiệu được lấy mẫu nén để giảm sai số khôi phục, giảm độ phức tạp tính toán và tăng tốc độ của thuật toán.
- Nghiên cứu áp dụng kỹ thuật lấy mẫu nén cho những ứng dụng cụ thể.

1.3.1. Các nghiên cứu về thiết kế ma trận xác định

Các ma trận ngẫu nhiên thỏa mãn tiêu chí RIP với xác suất cao nhưng khó thực hiện trong thực tế đối với các bài toán quy mô lớn do khối lượng tính toán và bộ nhớ lưu trữ lớn. Hiện nay phần lớn các nghiên cứu về ma trận lấy mẫu nén tập trung vào thiết kế các ma trận xác định. Howard và các cộng sự trình bày việc xây dựng ma trận lấy mẫu nén xác định dựa trên mã ReedMuller bậc hai và các hàm liên quan. Ma trận này có thể đạt tiêu chí RIP với xác suất cao [47]. Các kết quả trong [5] xây dựng ma trận lấy mẫu nén nhị phân với các phần tử $[0, 1]$ dựa trên các mã OOC, ma trận lấy mẫu nén lưỡng cực với các phần tử $[-1, 1]$ dựa trên mã tuyến tính BCH, và ma trận lấy mẫu nén tam phân với các phần tử $[-1, 0, 1]$ bằng việc kết hợp 2 loại ma trận nhị phân và lưỡng cực phía trên. Tiêu chí RIP được đảm bảo bằng tính chất không kết hợp của các ma trận này.

Shuxing Li [55] giới thiệu một ma trận lấy mẫu có cấu trúc xác định thông qua các đường cong đại số trên trường hữu hạn. Sự đa dạng của các đường cong đại số cung cấp nhiều sự lựa chọn cho ma trận lấy mẫu. Có thể tạo ra các ma trận lấy mẫu nén với hiệu quả cao bằng việc lựa chọn các đường cong thích hợp. Jun Zhang [109] thiết kế ma trận lấy mẫu nén từ mã LDPC với mục tiêu tạo ra một ma trận lấy mẫu có độ phức tạp thấp, dễ dàng triển khai

trên phần cứng. Hiệu suất của ma trận đề xuất được phân tích từ tính chất không kết hợp của các ma trận.

Nghiên cứu trong [95] đề xuất xây dựng ma trận lấy mẫu nén nhị phân có cấu trúc xác định từ hình học đơn nhất. Nghiên cứu cũng trình bày các phương pháp cắt và ghép từ các ma trận con để có thể mở rộng ma trận lấy mẫu nén với nhiều kích thước khác nhau giúp linh động khi ứng dụng trong thực tế.

Các ma trận lấy mẫu nén này cũng được thực hiện trên phần cứng số FPGA với độ phức tạp phần cứng thấp, tốc độ lấy mẫu và khôi phục cao, khả thi khi được triển khai với các ứng dụng trong thực tế [37], [40], [91].

1.3.2. Các nghiên cứu về thuật toán tham lam

Các thuật toán khôi phục tín hiệu được lấy mẫu nén nhanh chóng và chính xác là trọng tâm của nghiên cứu về CS, chúng sẽ là chìa khóa để có thể ứng dụng kỹ thuật CS trong thực tế. Mallat và Zhang đã trình bày thuật toán đuổi khớp (MP) trong [63], thuật toán này tìm kiếm các ánh xạ phù hợp nhất của dữ liệu nhiều chiều thông qua một bộ từ điển chính là ma trận lấy mẫu nén Θ. Ý tưởng cơ bản của thuật toán này là biểu diễn một tín hiệu f từ không gian Hilbert H dưới dạng tổng trọng số của các hàm lấy từ ma trận cảm biến Θ. Có nhiều thuật toán cải tiến dựa trên thuật toán gốc MP, thuật toán OMP là phiên bản phổ biến của các thuật toán cải tiến này và được đề xuất bởi Tropp và Gilbert [96]. Sự khác biệt chính của thuật toán OMP so với thuật toán MP là các hệ số trong các bước khôi phục là hình chiếu trực giao của tín hiệu lên ma trận lấy mẫu Θ. Needell và Tropp [72] đề xuất một thuật toán khôi phục tham lam được gọi là CoSaMP, thuật toán này đảm bảo khả năng khôi phục giống với cách tiếp cận theo phương pháp tối ưu hóa nhưng nó cũng cung cấp các giới hạn nghiêm ngặt về chi phí tính toán và lưu trữ dẫn đến nó khả thi khi áp dụng trong thực tế. Không giống như các thuật toán tham lam đơn giản, thuật toán CoSaMP xác định nhiều thành phần hơn

trong mỗi lần lặp lại, điều này cho phép thuật toán có thể thực hiện nhanh hơn đối với nhiều loại tín hiệu. CoSaMP sử dụng tính chất giới hạn đẳng trị RIP của ma trận lấy mẫu nén để đảm bảo cho quá trình khôi phục lại tín hiệu thành công.

Một dạng thuật toán tham lam khác rất phổ biến là các thuật toán tham lam gradient như GP và CGP [13]. Thuật toán GP ước lượng và cập nhật tại từng bước lặp bằng cách chọn hướng và kích thước bước theo gradient như đối với quá trình giải bài toán bình phương tối thiểu trong thuật toán đuổi khớp MP. Thuật toán tính toán gradient trong mỗi lần lặp do đó không làm tăng chi phí tính toán so với thuật toán MP. Chi phí tính toán tăng thêm so với thuật toán MP là phải đánh giá kích thước bước trong mỗi một lần lặp. Thuật toán CGP [13] cũng tượng tự như thuật toán GP nhưng thay vì chọn hướng theo gradient thì trong thuật toán CGP chọn hướng và kích thước bước theo gradient liên hợp.

1.4. Nhận xét các công trình nghiên cứu liên quan và hướng nghiên cứu của luận án

1.4.1. Nhận xét về công trình nghiên cứu liên quan

Qua tìm hiểu khảo sát và phân tích ở trên, nghiên cứu sinh nhận thấy vẫn còn một số vấn đề chưa được đề cập đến trong các nghiên cứu trước đây cụ thể như sau:

Các nghiên cứu về thiết kế các ma trận phù hợp với phần cứng và phương pháp tạo ra ma trận lấy mẫu với tốc độ cao để tích hợp cho các ứng dụng thực tế còn hạn chế.

Các nghiên cứu trước đây chưa chú trọng nhiều đến đánh giá khả năng bảo mật của ma trận lấy mẫu nén.

Các nghiên cứu về ma trận lấy mẫu và thuật toán khôi phục chủ yếu là các nghiên cứu mang tính phổ quát do đó đối với các ứng dụng cụ thể không

đạt được hiệu năng tối đa.

1.4.2. Hướng nghiên cứu của luận án

Trên cơ sở ý nghĩa khoa học, tính cấp thiết của đề tài và dựa trên các kết quả phân tích về hạn chế của các nghiên cứu liên quan, các hướng nghiên cứu trong luận án gồm:

- Đề xuất ma trận lấy mẫu nén xác định BPNSM được thiết kế dựa trên các chuỗi phi tuyến giả ngẫu nhiên. Trong đó ma trận lấy mẫu này có thể khả thi khi thực hiện trên phần cứng với tốc độ cao, nâng cao tính bảo mật của tín hiệu cần lấy mẫu.
- Đề xuất thuật toán khôi phục DRMP dựa trên việc cải tiến thuật toán tham lam gốc MP. Thuật toán cải tiến có độ phức tạp tính toán thấp hơn so với thuật toán gốc và lỗi khôi phục giảm sau mỗi bước lặp trong quá trình khôi phục.
- Đánh giá, so sánh hiệu năng của mô hình lấy mẫu nén dựa trên ma trận lấy mẫu nén BPNSM và thuật toán khôi phục DRMP thông qua các ví dụ mô phỏng.

1.5. Tổng kết chương

Nội dung chương 1 trình bày tổng quan về mô hình lấy mẫu nén, các phần tử trong mô hình, các tham số để đánh giá hiệu quả của mô hình. Khảo sát, đánh giá, phân tích các ưu nhược điểm đến từ các công trình nghiên cứu liên quan đến thiết kế ma trận lấy mẫu nén xác định [5], [59], [60], [83], [84], [107] và xây dựng thuật toán khôi phục tham lam trong lĩnh vực lấy mẫu nén [1], [29], [63], [72], [96]. Qua quá trình phân tích, đánh giá các công trình nghiên cứu đó, luận án đưa ra một số hạn chế còn tồn tại của các nghiên cứu trước đây. Trên cơ sở các hạn chế này, hướng nghiên cứu của luận án là

đề xuất một ma trận lấy mẫu nén xác định BPNSM với các phần tử được tạo thành từ các chuỗi giả ngẫu nhiên phi tuyến nhằm tăng tốc độ và tăng mức độ bảo mật trong quá trình lấy mẫu. Đồng thời luận án cũng đề xuất một thuật toán DRMP dựa trên việc cải tiến thuật toán tham lam gốc MP nhằm giảm độ phức tạp tính toán và giảm số lỗi ở mỗi bước trong quá trình khôi phục lại tín hiệu được lấy mẫu nén.

CHƯƠNG 2. THIẾT KẾ MA TRẬN LẤY MẪU NÉN XÁC ĐỊNH

Nội dung chương trình bày về cơ sở toán học và phương pháp thiết kế một ma trận lấy mẫu nén xác định dựa trên các chuỗi phi tuyến giả ngẫu nhiên và cách thức thực hiện ma trận lấy mẫu nén với tốc độ cao trên một phần cứng điện tử số. Kết quả nghiên cứu của chương 2 được trình bày trong [C1].

2.1. Mở đầu

Lấy mẫu nén bao gồm ba quá trình chính: biểu diễn tín hiệu thưa, mã hóa tuyến tính, và giải mã phi tuyến hoặc khôi phục tín hiệu nén. Trong quá trình lấy mẫu nén, một ma trận lấy mẫu được sử dụng để lấy mẫu các thành phần của tín hiệu. Sự lựa chọn ma trận lấy mẫu có tác động quan trọng đến độ chính xác và thời gian xử lý đối với quá trình khôi phục. Do đó, việc thiết kế một ma trận lấy mẫu phù hợp đóng vai trò quan trọng và thiết yếu trong lấy mẫu nén.

2.2. Tiêu chí thiết kế ma trận lấy mẫu nén

Luận án này giới hạn vấn đề nghiên cứu trong mô hình lấy mẫu nén với tín hiệu đầu vào có kích thước hữu hạn. Cụ thể, với một tín hiệu $x \in \mathbb{R}^N$, coi các hệ thống lấy mẫu thu được M phép chiếu tuyến tính và có thể biểu diễn quá trình này bằng toán học như sau

$$y = \Phi x, \tag{2.1}$$

trong đó Φ là một ma trận có kích thước $M \times N$ và $y \in \mathbb{R}^M$. Ma trận Φ thực hiện giảm chiều tín hiệu nó sẽ ánh xạ tín hiệu trong không gian kích thước lớn

$\mathbb{R}^N$ xuống không gian kích thước nhỏ hơn $\mathbb{R}^M$ ($M \ll N$). Đối với mô hình lấy mẫu nén tiêu chuẩn, giả định rằng các phép đo là không thích nghi, nghĩa là các hàng của ma trận lấy mẫu Φ là cố định trước và không phụ thuộc vào các mẫu lấy được trước đó.

Vấn đề đặt ra đối với ma trận lấy mẫu nén là ma trận phải được thiết kế như thế nào để đảm bảo rằng có thể thu thập đầy đủ thông tin trong tín hiệu x. Tiếp theo, ma trận lấy mẫu nén phải đảm bảo cho quá trình khôi phục lại chính xác tín hiệu x từ các mẫu nén y. Trong hầu hết các nghiên cứu [10], [16], [17], [19], [49] chỉ tiêu RIP đối với ma trận lấy mẫu nén được sử dụng như một điều kiện để đảm bảo cho khả năng khôi phục lại chính xác tín hiệu thưa ban đầu.

Ma trận lấy mẫu nén tổng quát Φ thỏa mãn tính chất giới hạn đẳng trị RIP có độ phức tạp tính toán lớn, vì trong mỗi trường hợp, về cơ bản phải xem xét tất cả các ma trận con $\binom{N}{K}$. Do đó, việc thiết kế ma trận lấy mẫu nén xác định thỏa mãn tiêu chí RIP là một bài toán khó. Để các ma trận lấy mẫu nén có thể khả thi khi ứng dụng trong thực tế tính chất không kết hợp (*incoherent*) được đề xuất [30], [96]. Trong nhiều trường hợp, tính chất không kết hợp của ma trận lấy mẫu Φ có thể được tính toán một cách dễ dàng để đảm bảo quá trình khôi phục chính xác tín hiệu được lấy mẫu nén. Trong [49] chỉ ra rằng nếu giá trị không kết hợp thấp thì xác suất thỏa mãn tiêu chí RIP là cao.

Tính chất không kết hợp của ma trận Φ, ký hiệu là $\mu(\Phi)$, được định nghĩa là giá trị tuyệt đối lớn nhất của tích vô hướng giữa hai cột bất kỳ $\phi_i, \phi j$ của ma trận Φ và được trình bày như sau:

$$\mu(\Phi) = \underset{1 \le i < j \le N}{max} \frac{| < \phi_i, \phi j > |}{\|\phi_i\|_2 \|\phi_j\|_2}. \tag{2.2}$$

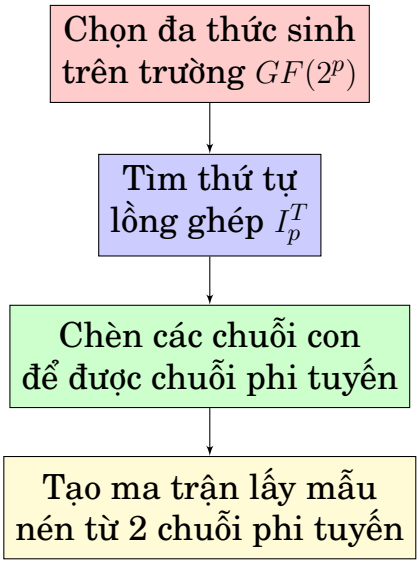
Các kết quả trong [30], [49], [96] chỉ ra rằng giá trị không kết hợp $\mu(\Phi)$ của ma trận lấy mẫu nén là nhỏ có nghĩa là cần ít số phép đo hơn để tái tạo lại tín hiệu ban đầu và các thuật toán khôi phục cũng sẽ có hiệu năng tốt hơn. Có

thể chỉ ra rằng tính chất không kết hợp của ma trận luôn nằm trong khoảng $\mu(\Phi) \in \left[\sqrt{\frac{N-M}{M(N-1)}}, 1\right]$ và giới hạn dưới được gọi là giới hạn Welch [101]. Đối với trường hợp $N \gg M$, giới hạn dưới $\mu(\Phi) \geq 1/\sqrt{M}$. Hơn nữa, khi các phần tử trong ma trận được tạo theo một phân bố ngẫu nhiên và có phương sai trung bình bằng 0 với sai lệch nhỏ, khi đó nếu giá trị M và N tăng thì giá trị không kết hợp sẽ hội tụ về $\mu(\Phi) = \sqrt{(2\log N/M)}$ [32]. Cấu trúc như vậy sẽ cho phép $K - sparse$ tăng tiệm cận tới $M = \mathcal{O}(K^2 \log N)$. Trong luận án ma trận lấy mẫu nén xác định sẽ được thiết kế với việc sử dụng tính chất không kết hợp để đảm bảo cho quá trình lấy mẫu và khôi phục lại tín hiệu.

2.3. Thiết kế ma trận lấy mẫu nén

Hầu hết các trình bày về thiết kế ma trận lấy mẫu nén trong [34], [68], [100] đều tập trung vào việc làm sao để khôi phục tín hiệu hiệu quả hơn. Tuy nhiên, trong lấy mẫu nén một vấn đề quan trọng khác là quá trình thu thập các tín hiệu thưa dựa trên các ma trận lấy mẫu nén tốn nhiều thời gian cho cả quá trình lấy mẫu và truyền tín hiệu. Trong luận án này, dựa trên các chuỗi giả ngẫu nhiên (PNS) được trình bày trong [C1], ma trận lấy mẫu nén xác định gọi là BPNSM được xây dựng theo các bước như trong hình 2.1. Các cột của ma trận lấy mẫu được tạo thành từ các chuỗi phi tuyến giả ngẫu nhiên. Chuỗi phi tuyến giả ngẫu nhiên đầu tiên được tạo thành bằng việc sử dụng các công cụ toán học trên trường hữu hạn [69] như thanh ghi dịch LFSR để tạo ra một chuỗi giả ngẫu nhiên tuyến tính. Sau đó, thông qua việc sử dụng biến đổi D và hàm Vết [22] để xác định thứ tự pha lồng ghép I_p^T, tiếp theo sẽ thực hiện việc thay thế các chuỗi con vào chuỗi tuyến tính. Các cột còn lại của ma trận được tạo thành bằng cách dịch vòng chuỗi PN phi tuyến ban đầu, sau đó sắp xếp và tổ hợp tất cả các chuỗi PN với nhau dưới dạng các vector cột, cuối cùng thu được ma trận BPNSM.

Ma trận trong luận án được thiết kế dựa trên một chuỗi PN phi tuyến và có sai số toàn phương trung bình tái thiết (MSE) gần với giá trị tối ưu, thỏa



Hình 2.1: *Các bước xây dựng ma trận BPNSM*

mãn tính chất không kết hợp. Do đó, ma trận lấy mẫu nén đảm bảo cho quá trình khôi phục tín hiệu một cách chính xác, bên cạnh đó việc tạo ra ma trận một cách nhanh chóng cũng làm giảm thời gian của quá trình lấy mẫu, đặc biệt khi thực hiện trên các hệ thống phần cứng số như FPGA.

2.4. Lý thuyết trường hữu hạn

Trường Galois [69] hay còn được gọi là trường hữu hạn được sử dụng hầu hết trong cấu trúc chuỗi giả ngẫu nhiên và phân tích chu kì, hàm tương quan và khoảng tuyến tính của thanh ghi dịch phản hồi tuyến tính (LFSR) và khởi tạo chuỗi phi tuyến. Nó còn được dùng trong các hệ thống bảo mật (ví dụ: khóa trao đổi Diffie-Hellman, chữ kí số (DSS), hệ thống bảo mật công cộng ElGamal...).

2.4.1. Cấu trúc $GF(p^n)$

Ta có trường hữu hạn $GF(p)$, trong đó p là nguyên tố, các phần tử trong $GF(p)$ là $\{0, 1, \ldots p-1\}$, với các phép cộng và nhân được tính theo mô-đun p. Đặt n là một số nguyên, cấu trúc trường hữu hạn $GF(p^n)$ bậc p^n, chọn $f(x)$ là

một đa thức tối giản trên $GF(p)$ mũ n. Gọi α là thành phần thỏa mãn $f(\alpha) = 0$. Ta có:

$$GF(p^n) = \{a_0 + a_1\alpha + ... + a_1\alpha^{n-1} | a_i \in GF(p)\}. \tag{2.3}$$

Cho $g(\alpha), h(\alpha) \in GF(p^n)$ và

$$g(\alpha) = \sum_{i=0}^{n-1} a_i\alpha^i,\ h(\alpha) = \sum_{i=0}^{n-1} b_i\alpha^i. \tag{2.4}$$

Các phép toán cộng và nhân trên $GF(p^n)$ được biểu diễn như sau:

$$g(\alpha) + h(\alpha) = \sum_{i=0}^{n-1}(a_i + b_i)\alpha^i \in GF(p^n), \tag{2.5}$$

$$g(\alpha).h(\alpha) = r(\alpha). \tag{2.6}$$

Trong đó, $r(\alpha)$ được tính là

$$\begin{gathered} g(\alpha).h(\alpha) = \sum_{i=0}^{n-1} a_i\alpha^i \sum_{j=0}^{m-1} b_j\alpha^j = \sum_{k=0}^{m+n} c_k\alpha^k = c(\alpha), \\ c_k = \sum_{i+j=k;0\le i\le n,0\le j\le m} a_i b_j. \end{gathered} \tag{2.7}$$

Chia $c(\alpha)$ cho $f(\alpha)$, ta có thể tạo hai đa thức $q(\alpha)$ và $r(\alpha)$ sao cho: $c(\alpha) = q(\alpha)f(\alpha) + r(\alpha)$ với $\deg(r(\alpha)) < n$. Với α thỏa mãn điều kiện $f(\alpha) = 0$, ta có $c(\alpha) = r(\alpha) \in GF(p^n)$.

$GF(p^n)$ cùng với hai phép toán (2.5), (2.6) được định nghĩa ở trên có dạng trường hữu hạn và bậc của trường này là p^n. Ví dụ: Đặt $p = 2$ và $f(x) = x^3+x+1$. Thì $f(x)$ là tối giản trên $GF(2)$. Gọi α là nghiệm của $f(x)$, với $f(\alpha) = 0$. Trường hữu hạn $GF(2^3)$ được định nghĩa bởi:

$$GF(2^3) = \{a_0 + a_1\alpha + a_2\alpha^2 | a_i \in GF(2)\}. \tag{2.8}$$

Ta có $GF(2^3)$ được định nghĩa bởi $f(x) = x^3 + x + 1$ và $f(\alpha) = 0$, được biểu diễn theo bảng sau:

Bảng 2.1: *Các phần tử của* $GF(2^3)$

$a_0a_1a_2$	$f(\alpha)$	α^i
000	0	0
001	1	1
010	α	α
100	α^2	α^2
110	$1+\alpha$	α^3
011	$\alpha+\alpha^2$	α^4
111	$1+\alpha+\alpha^2$	α^5
101	$1+\alpha^2$	α^6
$\alpha^7=1$		

2.4.2. Thanh ghi dịch phản hồi tuyến tính

Thanh ghi dịch phản hồi tuyến tính (LFSR) đã được sử dụng rộng rãi trong các ứng dụng như: tạo chuỗi giả ngẫu nhiên, chuỗi giả nhiễu, bộ đếm kỹ thuật số. LFSR có thể được thực hiện một cách đơn giản đối với cả phần cứng và phần mềm. Trong phần này, các cấu trúc của LFSR được sử dụng để biểu diễn một chuỗi giả ngẫu nhiên tuyến tính với các kiểu cấu hình của LFSR được tham khảo trong [22], [42].

Trạng thái tại thời điểm n của thanh ghi dịch được biểu diễn theo vector S_n:

$$S_n = [S_{0,n}S_{1,n}...S_{m-2,n}S_{m-1,n}]^T. \tag{2.9}$$

Thanh ghi dịch tại thời điểm $(n+1)$ có thể được ước lượng như sau:

$$\begin{aligned}
&S_{0,n+1} = S_{1,n} + g_1 S_{0,n},\\
&S_{1,n+1} = S_{2,n} + g_2 S_{0,n},\\
&\vdots\\
&S_{m-2,n+1} = S_{m-1,n} + g_{m-1} S_{0,n},\\
&S_{m-1,n+1} = g_r S_{0,n}.
\end{aligned} \tag{2.10}$$

Trong đó, phép cộng và phép nhân được tính mô-đun 2 và $g_i \in \{0,1\}, i = \{1,2,\ldots,m\}$. Công thức (2.10) có thể được viết dưới dạng ma trận như sau:

$$S_{n+1} = \begin{bmatrix} S_{0,n+1} \\ S_{1,n+1} \\ S_{2,n+1} \\ \vdots \\ S_{m-1,n+1} \end{bmatrix} = \begin{bmatrix} g_1 & 1 & 0 & \cdots & 0 \\ g_2 & 0 & 1 & \cdots & 0 \\ g_3 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ g_m & 0 & 0 & \cdots & 1 \end{bmatrix} \times \begin{bmatrix} S_{0,n} \\ S_{1,n} \\ S_{2,n} \\ \vdots \\ S_{m-1,n} \end{bmatrix}. \tag{2.11}$$

Hay dạng ngắn hơn:

$$S_i = G^i S_0,\ i = \{1,2,\cdots,n\}. \tag{2.12}$$

Trong đó, G là ma trận vuông chuyển tiếp bậc m, S_0 là vector trạng thái kích hoạt ban đầu. Đầu ra b_n của thanh ghi dịch tại thời điểm n được biểu diễn dưới dạng ngắn gọn hơn:

$$b_n = [1\ 0\ 0\ \cdots\ 0] \times G^n \times S_0. \tag{2.13}$$

Công thức (2.13) biểu diễn mối liên hệ giữa chuỗi đầu ra, ma trận chuyển tiếp và ma trận trạng thái kích hoạt ban đầu S_0 của LFSR.

Ví dụ cho thanh ghi dịch được biểu diễn như trong hình 2.2, kết quả có thể được ước lượng theo cách tương tự như sau:

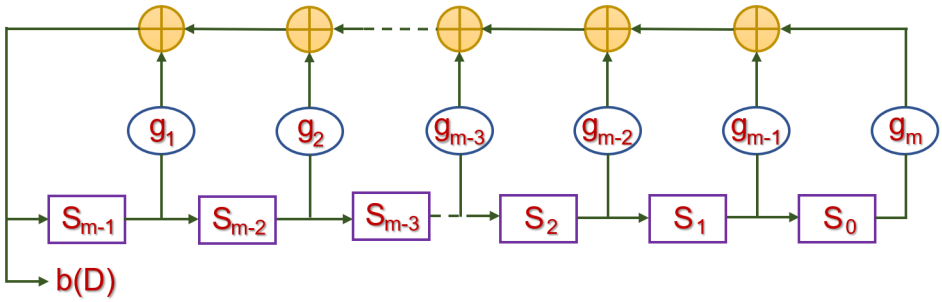
$$S_{n+1} = F \times S_n. \tag{2.14}$$

Trong đó,

$$S_{n+1} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ g_m & g_{m-1} & g_{m-2} & g_{m-3} & \cdots & g_1 \end{bmatrix}. \tag{2.15}$$

Đầu ra b_n tại thời điểm n là

$$b_n = [g_m \quad g_{m-1} \quad g_{m-2} \quad g_{m-3} \quad \cdots \quad g_1] \times F^n \times S_0. \tag{2.16}$$



Hình 2.2: *LFSR phản hồi Fibonacci [42]*

Việc biểu diễn chuỗi giả ngẫu nhiên bằng việc sử dụng LFSR đơn giản dễ thực hiện và có tốc độ cao. Đặc điểm này của thanh ghi dịch được sử dụng để có thể tạo ra ma trận lấy mẫu nén với tốc độ cao khi được thực hiện với các phần cứng điện tử số khả trình như FPGA.

2.4.3. Biến đổi D

Biến đổi D của chuỗi $\{b_n\}$ trên $GF(p)$ kí hiệu $D[b_n]$ được biểu diễn như sau:

$$D[b_n] = \sum_{i=0}^{m} b_i d^i \quad b_i \in \{GF(p)\}. \tag{2.17}$$

Ví dụ: Đặt $b_n = 010111$, biến đổi D của b_n có dạng $D(b_n) = d + d^3 + d^4 + d^5$. Biến đổi ngược của D có dạng $D^{-1} = b_n$. Biến đổi D của chuỗi có dạng một đa thức d trên $GF(p)$ và được ứng dụng trong phân tích tín hiệu, hệ thống truyền dữ liệu. Phép biến đổi D trong việc tạo chuỗi b_n của thanh ghi dịch phản hồi tuyến tính (LFSR) cho bởi:

$$b(d) = \frac{S(d)}{g(d)}, \tag{2.18}$$

trong đó, $g(d)$ là đa thức tạo chuỗi sử dụng bộ LFSR có bậc n và $S(d)$ có bậc

$\leq (n-1)$ là trạng thái kích hoạt trên các thanh ghi dịch phản hồi. Khi $g(d)$ là nguyên tố, chuỗi LFSR là một chuỗi m và có (p^n-1) đa thức $S(d)$ (p^n-1 trạng thái kích hoạt các thanh ghi LFSR). Ví dụ cho đa thức $g(d) = 1+d+d^3, S(d) = 1$ ta có

$$b(d) = \frac{S(d)}{g(d)} = \frac{1}{1+d+d^3} = 1 + d + d^2 + d^4 + d^7 + d^8 + \cdots$$

cho chuỗi nhị phân đầu ra là: $b_n = 111010011\cdots$

Dựa trên tính chất của phép biến đổi D, ta có thể biểu diễn cấu trúc lồng ghép (chuỗi dịch) của các chuỗi con (các chuỗi thành phần) để tạo chuỗi với hàm tương quan lí tưởng đóng vai trò đảm bảo cho tính chất không kết hợp khi tạo thành ma trận có giá trị nhỏ. Hai thủ tục được đưa ra ở đây là:

- **Thủ tục 1: Mở rộng chuỗi con**

  Đặt b_n là một chuỗi m được khởi tạo bởi $g(d)$ có bậc $n = l.m, T = \dfrac{2^n-1}{2^m-1}$.

  Đặt $b(d)$ là biến đổi D của b_n, theo công thức (2.18) ta có: $b(d) = \dfrac{S(d)}{g(d)}$

  Đặt $\{W_n\}$ là chuỗi con có độ dài $N = 2^m - 1$ và được biểu diễn bởi phép biến đổi D của nó: $W(D) = \dfrac{S_i(d)}{g_s(d)}$, trong đó: $S_i(d)$ là trạng thái kích hoạt thanh ghi LFSR của chuỗi con và $g_s(d)$ là đa thức khởi tạo của chuỗi con đó. Nhờ có cấu trúc lồng ghép của $\{b_n\}$, ta có thể biểu diễn nhanh $b(d)$ như sau:

  $$b(d) = \sum_{i=0}^{T-1} d^i W_i(d^T), \quad i = 0, 1, \cdots, T-1. \tag{2.19}$$

  Theo tính chất của biến đổi D, chuỗi phi tuyến giả ngẫu nhiên $\{b_n\}$ có thể được cấu trúc bằng cách lồng ghép T pha của $\{W_n\}$. Các pha của $\{W_n\}$ có thể được biểu diễn thông qua 3 bước sau:

  - Bước 1: mở rộng chuỗi con $W_i(d)$ bằng cách chèn $(T-1)$ giá trị 0 giữa hai bit liên tiếp của W_n, trong biến đổi D, nó tương đương với việc

thay đổi d với d^T.

$$W_i(d^T) = \frac{S_i(d^T)}{g_s(d^T)}. \tag{2.20}$$

– Bước 2: Biểu diễn biến đổi D của b_n theo dạng lồng ghép của $W_i(d)$.

$$b(d) = \sum_{i=0}^{T-1} d^i W_i(d^T) = \sum_{i=0}^{T-1} \frac{d^i S_i(d^T)}{g_s(d^T)}. \tag{2.21}$$

Sau đó, đặt $G(d) = \sum_{i=0}^{T-1} d^i S_i(d^T)$ ta có:

$$G(d) = \frac{S(d).g_s(d^T)}{g(d)}. \tag{2.22}$$

– Bước 3: Đặt $d^T = D$, pha dịch đầu ra $\frac{S_i(D)}{g_s(D)}$. Nhóm lại $b(d)$ như sau:

$$b(d) = \sum_{i=0}^{T-1} d^i W_i(D). \tag{2.23}$$

Ví dụ: Đặt $g(d) = 1+d^3+d^4,\ n = 2.2,\ m = 2,\ T = 15/3 = 5,\ g_s(d) = 1+d+d^2$. Với $G(d) = \frac{S(d).g_s(d^5)}{g(d)}$,$S(d) = 1$ ta có:

$$G(d) = \frac{1 + d^5 + d^{10}}{1 + d^3 + d^4} = 1 + d^3 + d^4 + d^5 + d^6.$$

Và

$$b(d) = \frac{G(d)}{g_s(d^5)} = \frac{1 + d^3 + d^4 + d^5 + d^6}{1 + d^5 + d^{10}}.$$

Đặt $d^5 = D$ ta có: $b(d) = \frac{1(1 + D) + D.d + 1.d^3 + 1.d^4}{1 + D + D^2}$, so sánh với bảng 2.2 thu được thứ tự lồng ghép của b_n : $I_p^5 = \{2, 0, \infty, 1, 1\}$. Trong đó, ∞ biểu diễn cho chuỗi toàn 0.

- **Thủ tục 2: Tách chuỗi m bằng phép giảm mẫu**

Nếu giảm mẫu $\{b_n\}$ T lần, được:

$$\{a_n\} = \{b_n\} = Tr_1^n(\alpha^{Tn}) = Tr_0^n(\beta^n), \tag{2.24}$$

với $\beta = \alpha_T$, α_T là thành phần nguyên tố trong $GF(2^m)$, $\{a_n\}$ còn là chuỗi m có độ dài $N = \dfrac{2^n - 1}{T}$.

Khi lấy mẫu bắt đầu từ bit đầu tiên, được chuỗi con: $\left\{a_0, a_T, \cdots, a_{(2^m-2)T}\right\}$.

Tương tự, có thể ước lượng chuỗi con $\left\{a_{t-1}, a_{T+t-1}, \cdots, a_{(2^m-2)T+t-1}\right\}$ khi lấy mẫu bắt đầu từ bit thứ t. Thật vậy, trên miền thời gian, những chuỗi con này có thể được biết như chuỗi dồn thời gian $\{a_{nT}\}\{a_{nT+1}\}\cdots\{a_{(n+1)T-1}\}$ trên T khe thời gian:

$$M = \begin{bmatrix} a_0 & a_1 & \cdots & a_{T-1} \\ a_T & a_{T+1} & \cdots & a_{2T-1} \\ \vdots & \vdots & \cdots & \vdots \\ a_{(2^m-2)T} & a_{(2^m-2)T+1} & \cdots & a_{(2^m-1)T-1} \end{bmatrix} \tag{2.25}$$
$$= \{a_{nT}\}\{a_{nT+1}\}\cdots\{a_{(n+1)T-1}\}.$$

Ví dụ: Với $m = 3$, $n = 6$, α là một thành phần nguyên tố trên $GF(2^6)$ với đa thức nguyên thủy $b(d) = 1 + d^5 + d^6$ trên trường $GF(2)$. $\{b_n\}$ là chuỗi m được khởi tạo bởi $b(d)$.

$\{b_n\} = \{0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 1.\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1.\ 0\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 1.\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0.$
$0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0.\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1.\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ \}$.

Lấy mẫu b_n với $T = 9$ được $\{a_n\} = \{b_n\}$ và phân hoạch lại $\{b_n\}$ theo ma trận 7×9 như sau:

$$M = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{2.26}$$

Đối chiếu cột M với bảng 2.2 ta được $I_p^T = \{\infty, 5, 3, 5, 6, 3, 3, 2, 5\}$, trong đó

∞ biểu thị cho chuỗi toàn 0.

Bảng 2.2: *Biến đổi D của chuỗi m*

$G_s(d)$	Chuỗi con	Dạng nhị phân	Chỉ số pha	$S_i(D)$
$1+d^2+d^3$	T^0Z	1001011	0	$1+d^2$
	T^1Z	0010111	1	d^2
	T^2Z	0101110	2	d
	T^3Z	1011100	3	1
	T^4Z	0111001	4	$d+d^2$
	T^5Z	1110010	5	$1+d$
	T^6Z	1100101	6	$1+d+d^2$
$1+d+d^2$	T^0W	011	0	d
	T^1W	110	1	1
	T^2W	101	2	$1+d$

2.4.4. Hàm Vết

a. Trường con và trường mở rộng

Số phần tử trong bất kì trường Galois nào cũng phải là lũy thừa của một số nguyên tố, và nó có một và chỉ một trường Galois $GF(p^s)$ với p^s phần tử. Trường $GF(p^s)$ có thể xem như một trường mở rộng của trường $GF(p)$, và trường $GF(p)$ có thể có dạng một trường con của $GF(p^s)$ bởi vì các phần tử của $GF(p)$ được mở rộng đến $GF(p^s)$, nhưng trường $GF(p^s)$ có thể còn được mở rộng đến một trường lớn hơn $GF[(p^s)^r]$. Nó quan trọng để phân biệt giữa các thành phần $GF(p)$, $GF(p^s)$ và $GF[(p^s)^r]$:

- Thành phần α của trường $GF(p)$ là một số nguyên trong tập $\{0,1,2,\cdots,p-1\}$, với phép cộng và nhân được tính mô-đun p.
- Bất kì thành phần α nào của trường $GF(p^s)$ đều là một đa thức có bậc $\leq s-1$ trên $GF(p)$, các hệ số của đa thức là số nguyên trong tập $\{0,1,2,\ldots,p-1\}$; các phép cộng và nhân được tính mô-đun với $(g(x),p)$, trong đó $g(x)$ là một đa thức tối giản có bậc s trên $GF(p)$.

- Bất kì thành phần α nào của trường $GF[(p^s)^r]$ đều là một đa thức có bậc $\leq r-1$ trên $GF(p^s)$, các hệ số của đa thức là các phần tử (các đa thức) trong $GF(p^s)$, các phép cộng và nhân được tính mô-đun $(f(x), g(x), p)$, trong đó $f(x)$ là một đa thức tối giản có bậc r trên $GF(p^s)$ và $g(x)$ là một đa thức tối giản bậc s trên $GF(p)$.

b. Hàm Vết

Gọi α là một phần tử trên $GF[(p^s)^r]$, sau đó vết liên quan đến trường con $GF(p^s)$ được định nghĩa như sau:

$$Tr_s^{rs}(\alpha) = \alpha + \alpha^{p^s} + \alpha^{p^{2s}} + \alpha^{p^{3s}} + \cdots + +\alpha^{p^{(r-1)s}} = \sum_{i=0}^{r-1} \alpha^{p^{is}}. \tag{2.27}$$

Hàm Vết có các tính chất sau:

- Thành phần $\alpha \in GF(p^s)$ là nghiệm của $x^{p^s} - x = 0,\ \alpha^{p^s} = \alpha$ có

$$(Tr_s^{rs}(\alpha))^{p^s} = Tr_s^{rs}(\alpha). \tag{2.28}$$

- Với mọi $\alpha \in GF[(p^s)^r]$ có

$$Tr_s^{rs}(\alpha) = Tr_s^{rs}(\alpha^{p^s}). \tag{2.29}$$

- Với mọi $\alpha, \beta \in GF[(p^s)^r]$, vì $(\alpha + \beta)^{p^{ks}} = \alpha^{p^{ks}} + \beta^{p^{ks}}$ với k là số nguyên có

$$Tr_s^{rs}(\alpha + \beta) = Tr_s^{rs}(\alpha) + Tr_s^{rs}(\beta). \tag{2.30}$$

- Cho $b \in GF(p^s)$, thì $b^{p^s} = b$ và $b^{p^{ks}} = b$, với bất kì $\alpha \in GF[(p^s)^r]$, có

$$Tr_s^{rs}(b\alpha) = b.Tr_s^{rs}(\alpha). \tag{2.31}$$

- $Tr_s^{rs}(\alpha) = b$, b là bất kỳ trên $GF(p^s)$ có p^{rs-s} kết quả thỏa mãn $\alpha \in GF[(p^s)^r]$.

- Nếu $s = 1$, thì $Tr_1^r(\alpha)$ ánh xạ các thành phần α của $GF(p^r)$ lên các thành

phần của $GF(p)$. Hơn nữa, nếu $m = rs$ và $\alpha \in GF(p^m)$, có tính chất bắc cầu

$$Tr_1^m(\alpha) = Tr_1^s(Tr_s^m(\alpha)). \tag{2.32}$$

c. Tìm thứ tự lồng ghép I_p^T theo hàm Vết

Cho m, n là 2 số nguyên dương và m là ước số của n, $T = (2^n - 1)/(2^m - 1)$ và α là phần tử nguyên tố của trường hữu hạn $GF(2^n)$.

Hàm vết của x là ánh xạ của $GF(2^q)$ xuống $GF(2^p)$ là

$$Tr_p^q(x) = \sum_{k=0}^{p/q-1} x^{2^{pk}}. \tag{2.33}$$

Thứ tự lồng ghép I_p^T được định nghĩa qua hàm vết Tr là

$$I_p^j = \begin{cases} i & Tr_p^q(\alpha^j) = \alpha^{Si} \quad \text{với } i = 0, 1, \cdots, 2^m - 2 \\ \alpha & Tr_p^q(\alpha^j) = 0 \qquad \text{với } i = 0, 1, \cdots, S - 1. \end{cases} \tag{2.34}$$

Để biểu diễn chuỗi được tạo ra trong phần trên có thể dùng các công cụ toán học như phép toán biến đổi D, hàm Vết, biểu diễn LFSR bằng ma trận trạng thái hữu hạn [22].

Phép biến đổi D là ngắn và dễ dàng thực hiện, nó chứa đầy đủ thông tin về trạng thái LFSR. Phép biến đổi D có thể được áp dụng cho bất kì chuỗi tuần hoàn nào có độ dài có thể phân tích thành dạng $L = T \times N = 2^n - 1$. 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 chuỗi.

2.5. Chuỗi trải phổ PN phi tuyến lồng ghép

Để nâng cao xác suất thỏa mãn tính chất giới hạn đẳng trị RIP về mặt lý thuyết giữa các cột của ma trận BPNSM phải có giá trị tương quan nhỏ. Chuỗi trải phổ PN phi tuyến lồng ghép có tính tương quan hoàn toàn phù hợp với điều kiện trên và tốt hơn nhiều so với các chuỗi PN truyền thống [22].

Phương pháp thiết kế chuỗi trải phổ PN lồng ghép phi tuyến dựa trên cấu trúc của chuỗi m (chuỗi cực đại tuyến tính có hàm tự tương quan ACF lý tưởng). Bước đầu tiên là tìm hàm vết của trường lớn $GF(2^n)$ xuống trường con $GF(2^m)$ dưới dạng một chuỗi dịch pha S (đó chính là trình tự sắp xếp các chuỗi con trong chuỗi lớn). Sau đó duy trì thứ tự sắp xếp chuỗi con như cũ nhưng thay bằng các chuỗi con khác để có chuỗi phi tuyến. Ngoài ra, có thể sử dụng kỹ thuật ghép các chuỗi con theo thời gian qua các mắt trễ và công cụ toán học mô tả là biến đổi D.

Khi thiết kế chuỗi nhị phân phi tuyến giả ngẫu nhiên các tiêu chí cần quan tâm bao gồm:

- Có đặc trưng thống kê tốt, thể hiện bằng việc các phần tử 0 và 1 trong chuỗi phải cân bằng và phân bố đồng đều cho mọi chuỗi.
- Với mỗi cặp chuỗi giả ngẫu nhiên, hàm tương quan chéo giữa chúng phải bé hơn giá trị cho trước δ trong một khoảng lệch pha nào đó.

Quy trình thiết kế các chuỗi nhị phân có vùng tương quan thấp bao gồm các bước sau:

- Bước 1: Tìm thứ tự lồng ghép I_p^T (chuỗi dịch pha) các chuỗi con có độ dài $N = 2^m - 1$ để tạo nên một chuỗi m lớn có độ dài $L = T.N = 2^n - 1$.
- Bước 2: Giữ nguyên giá trị lồng ghép nhưng thay các chuỗi con bằng các chuỗi nhị phân khác có hàm tương quan lý tưởng.

2.5.1. Phân hoạch chuỗi lớn

Lấy mẫu chuỗi $\{b_n\}$ với khoảng cách T, và sắp xếp chúng dưới dạng các cột của ma trận $M = N.T$:

$$N = \frac{2^n - 1}{T}, \tag{2.35}$$

$$M = \begin{vmatrix} a_0 & a_1 & \cdots & a_{T-1} \\ a_T & a_{T+1} & \cdots & a_{2T-1} \\ a_{2T} & a_{2T+1} & \cdots & a_{3T-1} \\ \vdots & \vdots & \cdots & \vdots \\ a_{(2^m-2)T} & a_{(2^m-2)T+1} & \cdots & a_{(2^m-1)T-1} \end{vmatrix}. \tag{2.36}$$

Xét về mặt biến đổi D, đấy chính là ghép $\{w_n\}$ theo thời gian,

$$\{a_{nT}\}\{a_{nT+1}\}\{a_{2nT+1}\}\cdots\{a_{n(2^m-2)T-1}\}. \tag{2.37}$$

Trên quan điểm hàm vết thấy rằng các cột của ma trận M là các pha của chuỗi con:

$$\{w_n\} = \{b_{nT}\} = Tr_1^n(\alpha^{Tn}) = Tr_0^n(\beta^n). \tag{2.38}$$

Trình tự sắp xếp các cột $\{w_n\}$ chính là thứ tự lồng ghép I_p^T.

2.5.2. Đánh giá chuỗi PN giả ngẫu nhiên lồng ghép phi tuyến

a. Hàm tự tương quan của chuỗi phi tuyến

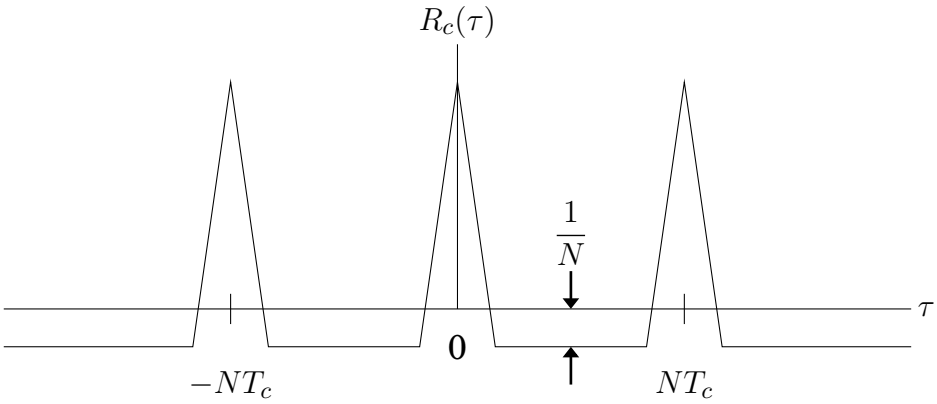
Do cấu trúc lồng ghép dựa trên trình tự pha I_p^T đã tạo nên hàm tự tương quan (ACF) lý tưởng của chuỗi m hình 2.3, còn các chuỗi con cân bằng chỉ đảm bảo được tính cân bằng của chuỗi m lớn. Hàm tương quan của chuỗi m được tính như sau:

$$R_c(\tau) = \frac{1}{NT_c}\int_0^{NT_c} c(t+\tau)\,c(t)\,dt\left(1+\frac{1}{N}\right)\Lambda_{T_c}(\tau) - \frac{1}{N}, \tag{2.39}$$

trong đó N là chu kì mã, T_c là độ rộng xung, $c = \pm 1$ trong trường hợp chuỗi lưỡng cực, $c = [0,1]$ trong trường hợp chuỗi nhị phân và $\Lambda_{T_c(\tau)}$ là hàm tam giác được xác định như sau:

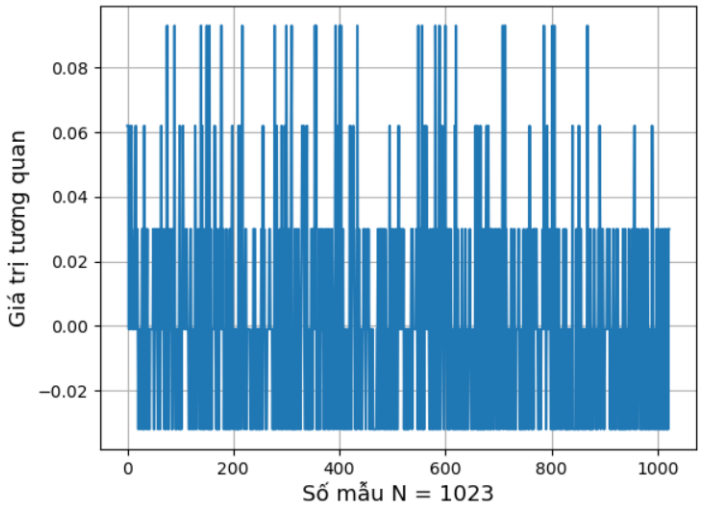
$$\Lambda_{T_c(\tau)} = \begin{cases} 1 - \dfrac{|\tau|}{T_c} & 0 \le |\tau| \le T_c \\ 0 & \neq \end{cases}. \tag{2.40}$$

Quy trình tính ACF của một chuỗi phân hoạch có cấu trúc lồng ghép dựa



Hình 2.3: *Hàm tự tương quan của chuỗi phi tuyến*

trên I_p^T khá đơn giản, nếu lưu ý rằng hàm tương quan phụ thuộc vào số các chuỗi con trùng nhau. Hình 2.4 thể hiện tương quan chéo giữa 2 chuỗi giả ngẫu nhiên lồng ghép phi tuyến. Có thể nhận xét với 2 chuỗi phi tuyến giả ngẫu nhiên khác nhau tương quan chéo của chúng rất bé, tính chất này giúp cho ma trận lấy mẫu BPNSM đảm báo thỏa mãn tính chất không kết hợp.



Hình 2.4: *Hàm tương quan chéo của chuỗi phi tuyến*

b. Độ phức tạp của chuỗi phi tuyến

Để đánh giá độ phức tạp của một chuỗi, người ta sử dụng khái niệm độ phức tạp tuyến tính (LC). Độ phức tạp tuyến tính của một chuỗi hữu hạn là độ dài của LFSR ngắn nhất tạo ra chuỗi đó hoặc là bậc đa thức sinh của LFSR đó.

Ý nghĩa vật lý của LC là nếu muốn khôi phục chuỗi tuyến tính bậc n, cần phải xác định đúng $2n$ bit liền nhau của chuỗi. Ví dụ: để khôi phục chuỗi m được tạo bởi đa thức sinh $g(d)$ bậc 6, ta cần xác định đúng 12 bit liền nhau của chuỗi đó. Còn để khôi phục lại chuỗi phi tuyến bậc 6, ta cần xác định đúng 24 bit liền nhau của chuỗi đó (tương đương với khôi phục chuỗi tuyến tính m có đa thức sinh $g(d)$ bậc 12).

Ngoài tính chất tự tương quan tốt, độ phức tạp tuyến tính của chuỗi phi tuyến tăng lên rất nhiều lần so với chuỗi tuyến tính [C1]. Tính chất này đảm bảo tính bảo mật dữ liệu khi thực hiện lấy mẫu nén với ma trận được tạo thành từ các chuỗi phi tuyến.

2.6. Xây dựng ma trận xác định BPNSM

Các ma trận BPNSM được đề xuất thuộc lớp ma trận xác định toàn phần có kích thước $(2^n - 1) \times 2^{n+1}$ với các phần tử mang giá trị $\{0,\ 1\}$, trong đó $n \geq 3$. Quá trình xây dựng ma trận BPNSM được trình bày như sau:

- Bước 1: Lựa chọn đa thức sinh $g_1(d)$, tìm thứ tự lồng ghép I_p^T bằng việc tính toán hàm Vết ánh xạ từ $GF(2^n)$ xuống $GF(2^m)$. Để tạo được chuỗi phi tuyến, giữ nguyên thứ tự lồng ghép I_p^T và thay thế chuỗi con bằng các chuỗi con khác tương ứng, chuỗi nhị phân phi tuyến giả ngẫu nhiên $\{b_n\}$ thu được có chiều dài $2^n - 1$. Dịch vòng chuỗi $\{b_n\}$ ta có 2^n các chuỗi phi tuyến giả ngẫu nhiên xác định. Đặt các chuỗi đó là vector cột của ma trận Φ_1, thu được một ma trận Φ_1 có kích thước $(2^n - 1) \times 2^n$, ma trận được

biểu diễn như sau:

$$\Phi_1 = \begin{bmatrix} b_0^0 & b_0^1 & \cdots & b_0^{2^n-1} \\ b_1^0 & b_1^1 & \cdots & b_1^{2^n-1} \\ \vdots & \vdots & \ddots & \vdots \\ b_{2^n-2}^0 & b_{2^n-2}^1 & \cdots & b_{2^n-2}^{2^n-1} \end{bmatrix}. \tag{2.41}$$

- Bước 2: Tương tự như vậy, việc tạo ra chuỗi giả ngẫu nhiên thứ 2 được thực hiện bằng việc lựa chọn một đa thức sinh $g_2(d)$ khác trong trường $GF(2^n)$. Lặp lại quy trình của Bước 1 được chuỗi $\{d_n\}$ có chiều dài $2^n - 1$ tương ứng thu được ma trận $\Phi_2 \in \mathbb{R}^{(2^n-1)\times 2^n}$. Ma trận Φ_2 có dạng như sau:

$$\Phi_2 = \begin{bmatrix} d_0^0 & d_0^1 & \cdots & d_0^{2^n-1} \\ d_1^0 & d_1^1 & \cdots & d_1^{2^n-1} \\ \vdots & \vdots & \ddots & \vdots \\ d_{2^n-2}^0 & d_{2^n-2}^1 & \cdots & d_{2^n-2}^{2^n-1} \end{bmatrix}. \tag{2.42}$$

- Bước 3: Ghép hai ma trận $\Phi_1 \in \mathbb{R}^{(2^n-1)\times 2^n}$ và $\Phi_2 \in \mathbb{R}^{(2^n-1)\times 2^n}$ dưới dạng mở rộng thêm cột để thu được ma trận BPNSM Φ có kích thước $(2^n-1)\times 2^{n+1}$. Ma trận lấy mẫu nén BPNSM có dạng sau:

$$\begin{aligned} \Phi &= [\Phi_1|\Phi_2] \\ &= \left[\begin{array}{cccc|cccc} b_0^0 & b_0^1 & \cdots & b_0^{2^n-1} & d_0^0 & d_0^1 & \cdots & d_0^{2^n-1} \\ b_1^0 & b_1^1 & \cdots & b_1^{2^n-1} & d_1^0 & d_1^1 & \cdots & d_1^{2^n-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{2^n-2}^0 & b_{2^n-2}^1 & \cdots & b_{2^n-2}^{2^n-1} & d_{2^n-2}^0 & d_{2^n-2}^1 & \cdots & d_{2^n-2}^{2^n-1} \end{array}\right]. \end{aligned} \tag{2.43}$$

Từ quá trình xây dựng ma trận lấy mẫu nén, có thể thấy rằng các ma trận BPNSM có tốc độ lấy mẫu $(2^n - 1)/2^{n+1} \approx 0,5$ lần so với tốc độ Nyquist và tốc độ này có thể điều chỉnh bằng cách thay đổi cách ghép nối các ma trận thành phần để phù hợp với tín hiệu đầu vào. Ma trận BPNSM $\Phi \in \mathbb{R}^{(2^n-1)\times 2^{n+1}}$ bao gồm hai ma trận con $\Phi_1 \in \mathbb{R}^{(2^n-1)\times 2^n}$ và $\Phi_2 \in \mathbb{R}^{(2^n-1)\times 2^n}$ trong đó mỗi ma trận tương ứng với một họ chuỗi phi tuyến giả ngẫu nhiên. Hai chuỗi phi tuyến

giả ngẫu nhiên được tạo ra với quy trình tương tự nhau, trong đó sự khác biệt nằm trong sự lựa chọn đa thức nguyên tố ban đầu thuộc trường $GF(2^n)$ do đó không mất đi tính tổng quát.

2.7. Tính chất không kết hợp của ma trận BPNSM

Tính chất không kết hợp là một tiêu chí quan trọng để mô tả hiệu quả của ma trận lấy mẫu nén. Đối với ma trận lấy mẫu khi giá trị không kết hợp của ma trận lấy mẫu giảm thì xác suất khôi phục tín hiệu tăng. Trong phần này luận án sẽ trình bày về tính chất không kết hợp của ma trận lấy mẫu BPNSM được đề xuất và so sánh tính chất không kết hợp của nó với các ma trận lấy mẫu ngẫu nhiên Gauss và Bernoulli tương ứng.

Để phân tích tính chất không kết hợp của ma trận BPNSM, luận án sử dụng khái niệm về tương quan chéo và tương quan cực đại đối với họ các chuỗi nhị phân được trình bày trong [71]. Giá trị tương quan chéo của hai chuỗi nhị phân khác nhau $a = (a_0, a_1, \cdots, a_\nu)$ và $b = (b_0, b_1, \cdots, b_\nu)$ có cùng chu kì ν được định nghĩa như sau:

$$R_{a,b}(\tau) = \sum_{i=0}^{\nu-1} (-1)^{a_i + b_{i+\tau}} \text{ với } 0 \leq \tau \leq \nu - 1, \tag{2.44}$$

trong đó $i + \tau$ được tính modulo với ν. Nếu hai chuỗi a và b cùng được tạo ra từ một chuỗi dịch vòng khi đó $R_{a,b}(\tau)$ là giá trị tự tương quan của a và b.

Với $S = \{s^{(0)}, s^{(1)}, \cdots, s^{(r-1)}\}$ là tập r các chuỗi nhị phân có chu kỳ ν. Giá trị $R_{\max} = \max \left| R_{s^{(i)}, s^j}(\tau) \right|$ với $0 \leq \tau \leq \nu - 1$ và $0 \leq i,\ j \leq r - 1$, trong đó $\tau \neq 0$ nếu $i = j$. Ở đây, R_{max} là giá trị lớn nhất trong số tất cả các giá trị tự tương quan và tương quan chéo của các chuỗi trong S. R_{max} còn được gọi là giá trị tương quan cực đại của S.

Để thuận tiện trong quá trình phân tích tính toán có thể tính toán giá trị tương quan chéo trong 2 trường hợp giá trị của n là chẵn và lẻ. Đối với n là lẻ, giá trị tương quan chéo của hai chuỗi nhị phân bất kỳ a và b là $R_{a,b}(\tau) \in$

$\{-1, -1 \pm 2^{(n+1)/2}\}$ và giá trị tương quan cực đại $R_{\max} = 1 + 2^{(n+1)/2}$.

Đối với n là chẵn giá trị tương quan chéo của hai chuỗi nhị phân bất kỳ a và b là $R_{a,b}(\tau) \in \{-1, -1 \pm 2^{n/2}, -1 \pm 2^{n/2+1}\}$ và giá trị tương quan cực đại $R_{\max}$ là $1 + 2^{n/2+1}$.

Áp dụng các phân tích phía trên để tính toán giá trị không kết hợp [98] cho ma trận Φ có kích thước $(2^n - 1) \times 2^{n+1}$ với $n \geq 3$ được tạo thành từ 2 chuỗi phi tuyến giả ngẫu nhiên.

Đối với ma trận $\Phi \in \mathbb{R}^{(2^n-1)\times 2^{n+1}}$ có

$$\mu(\Phi) = \max_{1 \leq i \neq j \leq 2^{n+1}} \frac{|\langle \phi_i, \phi_j \rangle|}{\|\phi_i\|_2 \cdot \|\phi_j\|_2}, \tag{2.45}$$

trong đó ϕ_i là cột thứ i của Φ. Với ϕ_i và ϕ_j là hai chuỗi phi tuyến giả ngẫu nhiên có độ dài $2^n - 1$. Do đó

$$\|\phi_i\|_2 = \|\phi_j\|_2 = (2^n - 1)^{1/2}, \tag{2.46}$$

ở đây ma trận BPNSM $\Phi = [\Phi_1 | \Phi_2]$ được ghép từ hai ma trận thành phần $\Phi_1 \in \mathbb{R}^{(2^n-1)\times 2^n}$ và $\Phi_2 \in \mathbb{R}^{(2^n-1)\times 2^n}$ được tạo ra bởi hai đa thức nguyên tố tương ứng. Phụ thuộc vào giá trị của i và j cách tính $\max_{1 \leq i \neq j \leq 2^{n+1}} |\langle \phi_i, \phi_j \rangle|$ có thể được phân thành 2 trường hợp.

- Trường hợp 1: Tính toán với các cột đều nằm ở cùng một ma trận thành phần Φ_1 hoặc Φ_2 khi đó $1 \leq i, j \leq 2^n$ hoặc $2^n + 1 \leq i, j \leq 2^{n+1}$.

  Trong trường hợp này, $\max_{1 \leq i \neq j \leq 2^{n+1}} |\langle \phi_i, \phi_j \rangle| = \max_{1 \leq i \neq j \leq 2^n} \left| \left\langle \phi_1^i, \phi_1^j \right\rangle \right|$
  $= \max_{1 \leq i \neq j \leq 2^n} \left| \left\langle \phi_2^i, \phi_2^j \right\rangle \right|$.

  Thừa nhận rằng $\{b_n^i\}$ và $\{b_n^j\}$ là hai chuỗi nhị phân bất kỳ được tạo ra từ $\{b_n\}$ trên trường $GF(2^n)$. Khi đó có tích vô hướng của 2 cột tương ứng trong ma trận lấy mẫu nén được tính như sau $\left\langle \phi_s^i, \phi_s^j \right\rangle = \sum_{t=0}^{2^n-2} (-1)^{b_t^i(s) + b_t^j(s)} = R_{b^i(s), b^j(s)}(0)$ với $s = 1, 2$.

Từ phân tích dựa trên biểu thức (2.44), đối với trường hợp n là lẻ thì:

$$\begin{aligned}\max_{1\le i\ne j\le 2^n}\left|\left\langle \phi_1^i,\phi_1^j\right\rangle\right| &= \max_{1\le i\ne j\le 2^n}\left|\left\langle \phi_2^i,\phi_2^j\right\rangle\right| \\ &= \max_{1\le i\ne j\le 2^n}\left|R_{b^i(2),b^j(2)}(0)\right| \\ &= \max\left|-1,-1\pm 2^{(n+1)/2}\right| \\ &= 1+2^{(n+1)/2}.\end{aligned} \tag{2.47}$$

Tương tự như vậy đối với n chẵn, có

$$\max_{1\le i\ne j\le 2^n}\left|\left\langle \phi_1^i,\phi_1^j\right\rangle\right| = \max_{1\le i\ne j\le 2^n}\left|\left\langle \phi_2^i,\phi_2^j\right\rangle\right| = 1+2^{n/2+1}. \tag{2.48}$$

- Trường hợp 2: Tính toán với các cột nằm ở cả hai ma trận thành phần $1\le i\le 2^n$ và $2^n+1\le j\le 2^{n+1}$.

  Trong trường hợp này, $\max_{1\le i,j\le 2^{n+1}}\left|\left\langle \phi^i,\phi^j\right\rangle\right|$ = $\max_{1\le i,j\le 2^n}\left|\left\langle \phi_1^i,\phi_2^j\right\rangle\right|$.

  Thừa nhận rằng $\{b_n^i\}$ và $\{d_n^j\}$ là hai chuỗi nhị phân bất kỳ được tạo ra từ $\{b_n\}$ và $\{d_n\}$ trên trường $GF(2^n)$ tương ứng. Có tích vô hướng $\left\langle \phi_1^i,\phi_2^j\right\rangle = \sum_{t=0}^{2^n-2}(-1)^{b_t^i(1)+d_t^j(2)} = R_{b^i(1),d^j(2)}(0)$.

  Hai chuỗi nhị phân $\{b_n\}$ và $\{d_n\}$ là 2 chuỗi có cùng chu kỳ do đó 2 chuỗi $\{b_n^i\}$ và $\{d_n^j\}$ được tạo ra từ 2 chuỗi gốc cũng có cùng chu kỳ. Tồn tại một số nguyên τ để $R_{b^i(1),d^j(2)}(0) = R_{b^i(1),d^k(1)}(\tau)$. Theo biểu thức (2.44), τ có nghĩa là sự thay đổi về trình tự pha giữa 2 chuỗi và có thể suy ra với trường hợp n lẻ thì:

$$\begin{aligned}\max_{1\le i,j\le 2^n}\left|\left\langle \phi_1^i,\phi_2^j\right\rangle\right| &= \max_{1\le i,j\le 2^n}\left|R_{b^i(1),d^j(2)}(0)\right| \\ &= \max\left|R_{b^i(1),d^k(1)}(\tau)\right| \\ &= 1+2^{(n+1)/2}.\end{aligned} \tag{2.49}$$

  Phân tích tương tự đối với n chẵn, có $\max_{1\le i,j\le 2^n}\left|\left\langle \phi_1^i,\phi_2^j\right\rangle\right| = 1+2^{n/2+1}$.

Kết hợp trường hợp 1 và 2 có:

$$\max_{1\leq i,j\leq 2^n}\left|\left\langle \phi_1^i,\phi_2^j\right\rangle\right| = \begin{cases} 1+2^{(n+1)/2} & \text{với } n \text{ lẻ} \\ 1+2^{n/2+1} & \text{với } n \text{ chẵn.} \end{cases} \tag{2.50}$$

Tính chất không kết hợp của ma trận $\mu(\Phi)$ được tính đối với trường hợp n chẵn:

$$\mu(\Phi) = \frac{1+2^{n/2+1}}{2^n-1}, \tag{2.51}$$

và đối với trường hợp n lẻ

$$\mu(\Phi) = \frac{1+2^{(n+1)/2}}{2^n-1}. \tag{2.52}$$

Giá trị không kết hợp được sử dụng để đánh giá sự đảm bảo cho quá trình lấy mẫu nén và khôi phục tín hiệu của ma trận BPNSM và được so sánh với giá trị không kết hợp của các ma trận khác.

2.8. So sánh đánh giá ma trận BPNSM

Để đánh giá hiệu quả của ma trận được thiết kế tính chất không kết hợp của các ma trận lấy mẫu nén thường được so sánh với tính chất không kết hợp của ma trận Gauss [5], [37], [56], [83]. Trong luận án cũng đánh giá hiệu quả của ma trận xác định BPNSM được tạo thành từ các chuỗi giả ngẫu nhiên trên trường hữu hạn dựa trên tính chất không kết hợp và so sánh tính chất không kết hợp của nó với ma trận lấy mẫu nén cùng kích thước với các phần tử được tạo thành theo phân bố Gauss.

Trong [48] trình bày các nghiên cứu và đánh giá về xác suất của 2 chuỗi ngẫu nhiên cũng như xác suất của ma trận Gauss thỏa mãn tính chất không kết hợp. Đối với $\{b_n^i\}_{i=1}^k$ và $\{d_n^i\}_{i=1}^k$ là hai chuỗi có các phần tử ngẫu nhiên độc lập tuyến tính thỏa mãn $\left|b_n^i\right| \leq \lambda$ hoặc $\left|b_n^i, d_n^i\right| \leq \lambda^2$. Khi đó

$$Pr\left(\left|\sum_{i=1}^{k} b_n^i d_n^i\right| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2k\lambda^4}\right), \tag{2.53}$$

trong trường hợp 2 chuỗi $\{b_n^i\}_{i=1}^k$ và $\{d_n^i\}_{i=1}^k$ là hai chuỗi có các phần tử ngẫu nhiên độc lập tuyến tính theo phân bố Gauss với phương sai σ^2 áp dụng kết quả của biểu thức (2.53) có thể suy ra:

$$Pr\left(\left|\sum_{i=1}^{k} b_n^i d_n^i\right| \geq t\right) \leq 2\exp\left(-\frac{t^2}{4\sigma^2(k\sigma^2 + t/2)}\right). \tag{2.54}$$

Trong trường hợp giá trị n là chẵn, giả thiết $\{b_n^i\}_{i=1}^{2^n-1}$ và $\{d_n^i\}_{i=1}^{2^n-1}$ là 2 vector cột bất kỳ của ma trận Gauss G. Từ biểu thức (2.54) với $k = 2^n - 1, t > \frac{1+2^{n/2+1}}{2^n-1}$, và $\sigma^2 = \frac{1}{2^n-1}$, có

$$Pr\left(\left|\sum_{i=1}^{2^n-1} b_n^i d_n^i\right| \geq t\right) \leq 2\exp\left(-\frac{(2^n-1)t^2}{4+2t}\right). \tag{2.55}$$

Đặt $z(n,t) = 2\exp\left\{-\frac{(2^n-1)t^2}{4+2t}\right\}$. Có thể nhận thấy rằng $z(n,t)$ tăng khi giá trị n giảm dần. Do đó, $z(n,t) \leq z(4,t)$ từ đó suy ra:

$$Pr\left(\left|\sum_{i=1}^{2^n-1} b_n^i d_n^i\right| \geq t\right) \leq z(4,t) = 2\exp\left(-\frac{15t^2}{4+2t}\right). \tag{2.56}$$

Giá trị $z(4,t)$ tăng khi t giảm dẫn đến $z(4,t) < z\left(4, \frac{1+2^{n/2+1}}{2^n-1}\right)$. Đặt $z_1(n) = z\left(4, \frac{1+2^{n/2+1}}{2^n-1}\right)$. Rõ ràng, $z_1(n)$ tăng với n giảm.

Do đó, có $z(4,t) < z_1(n) \leq z_1(4) \approx 2\exp(-1.0385)$ từ đó suy ra:

$$Pr\left(\left|\sum_{i=1}^{2^n-1} b_n^i d_n^i\right| \geq t\right) \leq z_1(4) \approx 2\exp(-1.0385) \approx 0.708 \tag{2.57}$$

Theo biểu thức (2.2) $\left|\sum_{i=1}^{2^n-1} b_n^i d_n^i\right|$ có thể đặc trưng cho tính chất không kết hợp $\mu(\mathbf{G})$ của ma trận G.

Cho $S = \{b_1, b_2, \cdots, b_{2^{n+1}}\}$ có

$$\mu(\mathbf{G}) = \max_{\{b_n^i\},\{d_n^i\}} \left\{\sum_{i=1}^{2^n-1} b_n^i d_n^i | \{b_n^i\} \subset S, \{d_n^i\} \subset S \backslash \{b_n^i\}\right\}, \tag{2.58}$$

và

$$Pr\left(\min_{\{b_n^i\},\{d_n^i\}}\left|\sum_{i=1}^{2^n-1} b_n^i d_n^i\right| \geq t\right) \leq 0.708^{\frac{|S|(|S|-1)}{2}} = 0.708^{2^n(2^{n+1}-1)}. \quad (2.59)$$

Đặt $\delta_b(n) = 0.708^{2^n(2^{n+1}-1)}$ với $n \geq 4$. Có thể suy ra:

$$Pr\left(\min_{\{b_n^i\},\{d_n^i\}}\left|\sum_{i=1}^{2^n-1} b_n^i d_n^i\right| \geq t\right) \geq 1 - \delta_b(n) \approx 1. \quad (2.60)$$

Từ đó suy ra, $\mu(\mathbf{G}) = \max_{\{b_n^i\},\{d_n^i\}} \sum_{i=1}^{2^n-1} b_n^i d_n^i \geq t$. Với $t > \dfrac{1+2^{n/2+1}}{2^n-1}$ và kết hợp với kết quả của biểu thức (2.51) có $\mu(\Phi) = \dfrac{1+2^{n/2+1}}{2^n-1}$ suy ra $\mu(\mathbf{G}) > \mu(\Phi)$ trong trường hợp n chẵn. Với các bước phân tích tương tự như trên, kết quả này cũng đúng với trường hợp n là lẻ .

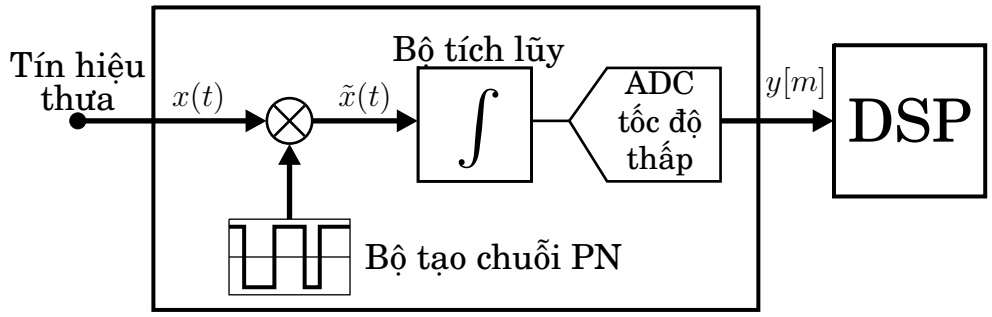
Có thể thấy rằng, so với các ma trận ngẫu nhiên với các phần tử được tạo ra theo phân bố Gauss, ma trận BPNSM được đề xuất có sự cân bằng tốt hơn giữa hiệu suất lấy mẫu, chi phí bộ nhớ, độ phức tạp tính toán và khả năng hiện thực hóa phần cứng của chúng.

Ma trận ngẫu nhiên phổ biến khác cũng được sử dụng để so sánh đánh giá với ma trận BPNSM đề xuất là ma trận Bernoulli. Ma trận Bernoulli $\mathrm{B} \in R^{M\times N}$ với các phần tử được tạo ra theo phân bố Bernoulli gồm 2 giá trị $[a_0, a1]$ có xác suất $Pr(a_0) = Pr(a_1) = 1/2$ [7]. Trong trường hợp $a_0 = 0, a_1 = 1$ và kích thước của ma trận Bernoulli B bằng với kích thước của ma trận đề xuất BPNSM thì 2 ma trận này gần như tương đương nhau. Nhưng ma trận Bernoulli cần thời gian để tạo ra lớn cũng như cần có bộ nhớ lưu trữ lớn hơn so với ma trận BPNSM đề xuất do tính chất của ma trận là ngẫu nhiên không có cấu trúc.

Trong phần mô phỏng luận án thực hiện việc so sánh đánh giá giữa 3 ma trận gồm ma trận Gauss, ma trận Bernoulli và ma trận đề xuất BPNSM để làm rõ hơn các nhận định trên.

2.9. Thực hiện ma trận lấy mẫu nén trên phần cứng

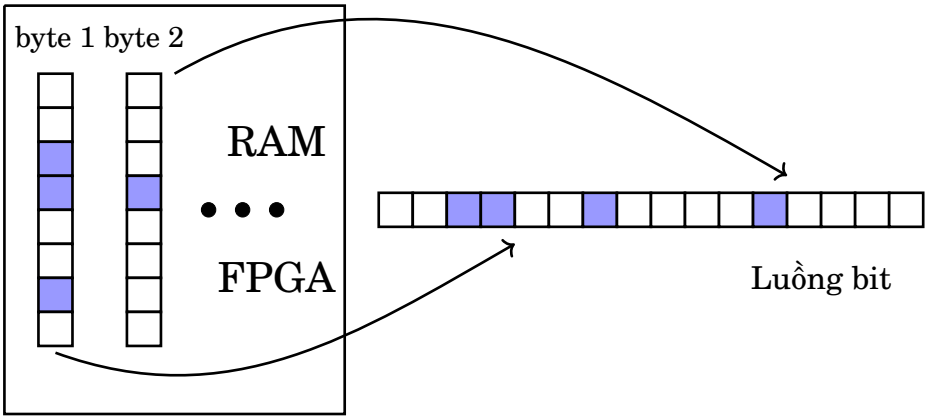
Trong phần này luận án giới thiệu một kỹ thuật thực hiện quá trình lấy mẫu nén áp dụng trên phần cứng. Trong [97] trình bày sơ đồ chuyển đổi tín hiệu tương tự thành thông tin (AIC) để thu được tín hiệu thưa không giới hạn băng tần. Hệ thống thực hiện được bằng cách nhân tín hiệu thưa với một chuỗi giả ngẫu nhiên tốc độ cao để trải tín hiệu trên toàn bộ phổ của chuỗi giả ngẫu nhiên như mô tả trong hình 2.5. Sau đó, tín hiệu ngẫu nhiên hóa được đưa qua bộ lọc thông thấp và bộ khử xung răng cưa để lấy mẫu với tần số lấy mẫu thấp hơn nhiều so với tần số Nyquist.



Hình 2.5: *Mô hình lấy mẫu nén băng rộng sử dụng ADC tốc độ thấp*

Trong sơ đồ hình 2.5 ma trận BPNSM Φ được tạo ra từ *Bộ tạo chuỗi PN* và *Bộ tích lũy*, khi đó bộ tạo chuỗi phi tuyến giả ngẫu nhiên sẽ đưa các giá trị trong vector hàng của ma trận Φ lên luồng bit để thực hiện phép nhân với tín hiệu $x(t)$. Mặc dù bộ tạo chuỗi ngẫu nhiên hoạt động với tốc độ bằng hoặc cao hơn tốc độ Nyquist, nhưng bộ ADC trong sơ đồ lại hoạt động với tốc độ thấp hơn. Điều này làm cho phần cứng của bộ AIC ít phức tạp hơn so với sử dụng bộ ADC để lấy mẫu với tốc độ Nyquist đối với tín hiệu $x(t)$ vì việc xây dựng bộ tạo tín hiệu ngẫu nhiên và bộ tích lũy đơn giản hơn so với việc chế tạo ADC tốc độ cao. Bộ AIC hoạt động ở tốc độ lấy mẫu thấp hơn nhiều so với tốc độ Nyquist, và nó phụ thuộc vào kích thước của ma trận lấy mẫu nén. Trong trường hợp cụ thể thực hiện với ma trận BPNSM Φ khi đó tốc độ của bộ AIC ≈ 0.5 so với tốc độ Nyquist.

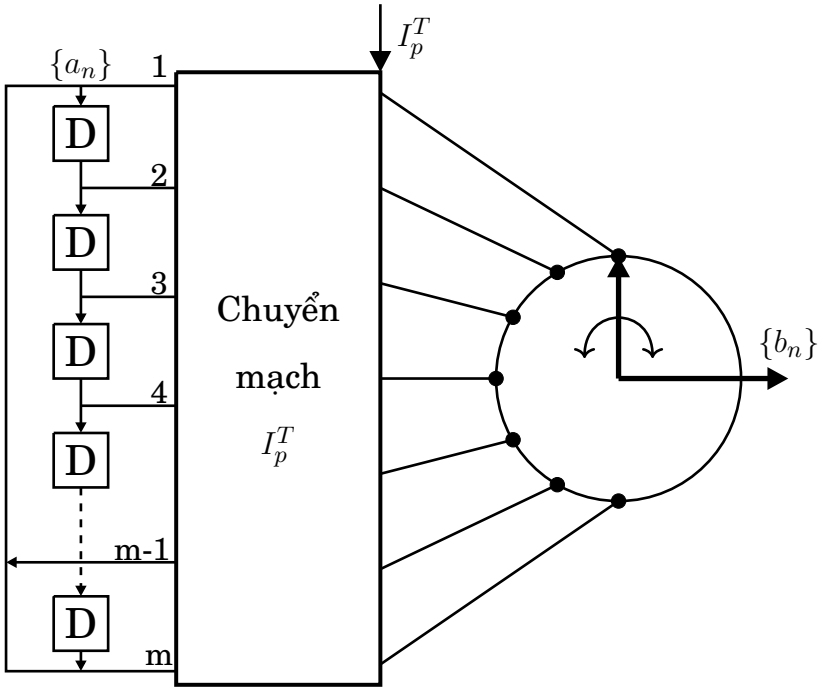
Việc tạo ma trận lấy mẫu có thể đạt được tốc độ rất cao bằng việc thực hiện trên một phần cứng FPGA đơn giản. Ma trận sau khi xây dựng được lưu trữ dưới dạng các byte dữ liệu trong bộ nhớ Flash của FPGA. Trong quá trình lấy mẫu, dữ liệu trong bộ nhớ được đọc bởi một xung đồng hồ tốc độ thấp và đưa lên trên luồng dựa theo cơ chế tra bảng (Look up Table) hỗ trợ bởi ngôn ngữ lập trình phần cứng. Sau đó, chuỗi byte được chuyển đổi thành chuỗi bit bằng cách sử dụng mạch song song nối tiếp như mô tả trong hình 2.6. Với kỹ thuật này, ma trận có thể được thiết kế với kích thước rất lớn, và nó cũng có thể được cấu hình động để lấy mẫu nén thích nghi. Kích thước của ma trận chỉ bị giới hạn bởi dung lượng bộ nhớ của FPGA.



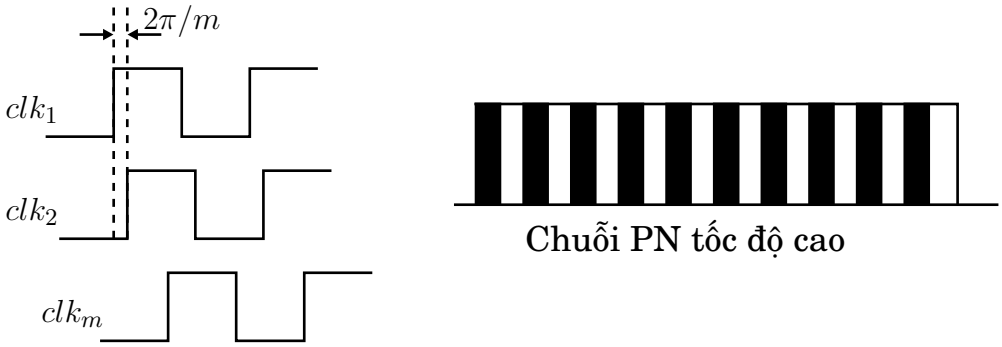
Hình 2.6: *Mô hình chuyển đổi từ byte trong bộ nhớ thành luồng bit*

Sau khi đọc mẫu từ trong bộ nhớ để có thể nâng cao tốc độ của luồng bit trong luận án cũng đề xuất sử dụng một bộ chuyển mạch cơ khí với m tiếp điểm như hình 2.7. Khi đó với tần số của các đầu ra song song là F sẽ tạo được một luồng bit nối tiếp với tần số mF.

Hình 2.8 biểu diễn chuỗi giả ngẫu nhiên tốc độ cao ở đầu ra của bộ chuyển mạch. Chuỗi giả ngẫu nhiên này được sử dụng để tạo ra ma trận lấy mẫu nén. Với mô hình phần cứng như đã trình bày ma trận lấy mẫu nén sẽ được tạo ra với thời gian rất ngắn giúp cho tốc độ lấy mẫu nén có thể được nâng cao một cách dễ dàng.



Hình 2.7: *Lồng ghép các chuỗi dịch*



Hình 2.8: *Giản đồ xung đầu ra sau chuyển mạch*

2.10. Tổng kết chương

Từ những phân tích và đánh giá trên có thể nhận thấy rằng ma trận lấy mẫu nén được biến đổi từ các chuỗi giả ngẫu nhiên trên trường hữu hạn có một số đặc điểm sau đây:

- Chi phí bộ nhớ thấp: BPNSM bao gồm các phần tử 0 và 1. Vì vậy BPNSM chỉ yêu cầu $M \times N$ bit để lưu trữ tất cả các phần tử. So với ma trận ngẫu nhiên Gauss ma trận này giảm yêu cầu về bộ nhớ, do đó mang lại hiệu

quả lưu trữ tốt hơn. Tính năng này làm cho BPNSM khả thi khi áp dụng trong các lĩnh vực thông tin và truyền thông có băng thông hạn chế. Hơn thế nữa, ma trận có thể tái tạo lại từ một số các đa thức thuộc trường hữu hạn nên không cần phải gửi toàn bộ giá trị các phần tử của ma trận, thay vào đó chỉ cần gửi đi các đa thức tạo ra ma trận, việc này có lợi ích lớn khi thực hiện đối với các bài toán truyền thông.

- Độ phức tạp tính toán thấp: Ma trận BPNSM được đề xuất cần ít tài nguyên hơn cho quá trình tính toán, thu thập và phục hồi lại tín hiệu gốc. Để thu thập và khôi phục lại tín hiệu thưa, các phép toán số học của BPNSM là cộng và trừ, trong khi đối với ma trận ngẫu nhiên Gauss yêu cầu cộng, trừ và nhân.

- Dễ dàng thực hiện với phần cứng: Như đã trình bày trong phần (2.4.2), việc triển khai BPNSM cực kỳ dễ dàng nhờ các cấu trúc LFSR, do đó nó rất khả thi khi áp dụng trên phần cứng ví dụ như FPGA, khác với các ma trận có cấu trúc ngẫu nhiên như Gauss và Bernoulli yêu cầu phần cứng cao trong quá trình tạo ra các số ngẫu nhiên.

- Ma trận lấy mẫu nén BPNSM có hạn chế là không thể được tạo ra với kích thước tùy ý do các phần tử của ma trận BPNSM được tạo ra từ một đa thức sinh trên trường $GF(2^n)$. Do đó không thể lấy mẫu tín hiệu có kích thước tùy ý, nếu sử dụng trong lĩnh vực viễn thông thì hạn chế này có thể bỏ qua vì các khung dữ liệu thường có kích thước 2^n.

Trong chương 4 luận án sẽ trình bày các mô phỏng ứng dụng mà sử dụng ma trận BPNSM được tạo ra từ các chuỗi giả ngẫu nhiên trên trường hữu hạn đối với tín hiệu đầu vào 1 chiều và 2 chiều để làm rõ hơn các nhận xét và đánh giá trong chương này.

CHƯƠNG 3. ĐỀ XUẤT THUẬT TOÁN KHÔI PHỤC TÍN HIỆU ĐƯỢC LẤY MẪU NÉN DRMP

Lấy mẫu nén là một tập hợp các phương pháp biểu diễn tín hiệu trên cơ sở một số phép đo giới hạn và sau đó khôi phục tín hiệu từ các phép đo này. Vấn đề làm thế nào để khôi phục hiệu quả tín hiệu gốc từ dữ liệu nén đóng một vai trò quan trọng trong kỹ thuật lấy mẫu nén. Đã có nhiều nghiên cứu về các thuật toán tái tạo như trong [14], [18], [72], [96]. Trong chương này luận án nghiên cứu và phân tích một số loại thuật toán dựa trên các kỹ thuật mà nó thực hiện. Từ đó, xây dựng một thuật toán cải tiến phù hợp với mục tiêu đặt ra của luận án. Kết quả nghiên cứu của chương 3 được trình bày trong [J1].

3.1. Chỉ tiêu đánh giá thuật toán khôi phục

Trong thực tế vấn đề cơ bản của các thuật toán khôi phục tín hiệu được lấy mẫu nén là làm thế nào khôi phục lại tín hiệu thưa x từ các phép đo có nhiễu

$$y = \Phi x + w. \tag{3.1}$$

Các nghiên cứu về thuật toán đều hướng tới mục tiêu có thể khôi phục nhanh, chính xác và ổn định tín hiệu thưa x từ y. Việc thiết kế các thuật toán khôi phục tín hiệu được lấy mẫu nén có thể phụ thuộc vào nhiều tiêu chí khác nhau. Một số chỉ tiêu đánh giá quan trọng được liệt kê như sau.

- **Tỉ số nén CR:** là tỷ số giữa số phép đo và số lượng mẫu trong tín hiệu gốc. Nó có thể được tính là $CR = M/N$ trong đó M là số chiều của vector lấy mẫu và N là số chiều của vector tín hiệu. Tỉ số này thể hiện rằng các tín hiệu có số chiều lớn có thể được khôi phục từ các vector có số chiều $M \ll N$.

- **Lỗi khôi phục:** là tiêu chí được sử dụng để đánh giá sự khác biệt giữa tín hiệu ban đầu và tín hiệu được khôi phục. Tiêu chí về lỗi khôi phục có thể được biểu diễn như sau [85]

$$E_x = \frac{\|x - \hat{x}\|}{\|x\|}, \tag{3.2}$$

trong đó E_x là lỗi khôi phục, x là tín hiệu gốc và $\hat{x}$ là tín hiệu khôi phục. Nó thể hiện giá trị lỗi của một thuật toán khôi phục.

Một phương pháp khác cũng thường được sử dụng để đánh giá lỗi khôi phục của thuật toán là sai số toàn phương trung bình (MSE). Giá trị này được sử dụng để đánh giá sự biến đổi của lỗi đối với thuật toán khôi phục theo thời gian. Nó được sử dụng như một mô hình dự đoán về ước lượng lỗi đối với một thuật toán khôi phục [85]. Sai số toàn phương trung bình để thể hiện sự sai khác giữa tín hiệu khôi phục và tín hiệu ban đầu. Cả lỗi khôi phục và lỗi toàn phương trung bình gần như tương tự nhau.

- **Tốc độ:** Một số ứng dụng lấy mẫu nén làm việc với các tín hiệu có số chiều lớn và yêu cầu đáp ứng nhanh theo thời gian thực. Do đó, các thuật toán khôi phục tín hiệu cần phải sử dụng tài nguyên tính toán tối thiểu. Trong luận án thời gian xử lý của thuật toán được sử dụng để đánh giá tốc độ.

- **Tương quan và hiệp phương sai:** Hệ số tương quan là giá trị thể hiện sự giống nhau giữa tín hiệu khôi phục và tín hiệu gốc. Nó đánh giá mức độ giống nhau của hai tín hiệu và nó nằm trong khoảng từ -1 đến 1. Giá trị tương quan > 0 thể hiện rằng tín hiệu gốc và tín hiệu khôi phục có tương quan thuận. Giá trị tương quan < 0 thể hiện rằng tín hiệu gốc và tín hiệu khôi phục có tương quan nghịch. Giá trị tương quan bằng không thể hiện rằng không mối quan hệ giữa hai tín hiệu này.

Giá trị hiệp phương sai là một thước đo thống kê tương ứng với sự tương quan giữa tín hiệu được khôi phục và tín hiệu gốc. Nó được biểu diễn

bằng biểu thức

$$cov = E([x - E(x)][\hat{x} - E(\hat{x})]), \tag{3.3}$$

trong đó $E(.)$ biểu thị kỳ vọng, x biểu thị tín hiệu ban đầu, và $\hat{x}$ biểu thị tín hiệu được khôi phục. Cả giá trị tương quan và giá trị hiệp phương sai đều phụ thuộc và mối quan hệ giữa tín hiệu ban đầu và tín hiệu được khôi phục và có thể coi 2 giá trị này đều thuộc cùng một thước đo để đánh giá đối với hiệu quả của thuật toán.

- **Độ phức tạp:** Độ phức tạp phản ánh hiệu quả của thuật toán khi thực hiện với lượng dữ liệu lớn. Nó có thể là sự phức tạp về mặt tính toán, phức tạp trong việc thiết kế ma trận lấy mẫu nén, thu được tín hiệu có số chiều lớn, thời gian xử lý của thuật toán lớn hoặc có yêu cầu quá cao về cấu hình phần cứng. Các thuật toán có độ phức tạp lớn có thể không được xem xét ngay cả khi chúng cung cấp kết quả chính xác [92].

Từ khi lĩnh vực lấy mẫu nén được đề xuất có khá nhiều các nghiên cứu về thuật toán khôi phục tín hiệu từ các mẫu nén đã được công bố [1], [29], [63], [72], [90], [96], [104], [106]. Trong nội dung của luận án nghiên cứu và trình bày thuật toán khôi phục tham lam MP [104] và các thuật toán cải tiến phổ biến của nó, từ đó xây dựng thuật toán DRMP là một thuật toán cải tiến dựa trên thuật toán MP.

3.2. Các thuật toán lặp lại tham lam

Các thuật toán lặp lại tham lam được sử dụng rộng rãi trong các ứng dụng lấy mẫu nén [35] do chúng có độ phức tạp tính toán thấp và khả năng khôi phục nhanh. Các thuật toán loại này thực hiện quá trình khôi phục lại tín hiệu gốc qua từng bước lặp đến khi thỏa mãn điều kiện dừng.

Trọng tâm của nghiên cứu về lấy mẫu nén là thiết kế các thuật toán có thể khôi phục được nhanh chóng và chính xác tín hiệu ban đầu từ một số các

mẫu nén, chúng sẽ là tiền đề để có thể ứng dụng kỹ thuật lấy mẫu nén trong thực tế. Hiện tại, trong nhóm các thuật toán tham lam thì thuật toán đuổi khớp tham lam (MP) và các thuật toán phái sinh được phát triển trên thuật toán này là phổ biến nhất [27], [35].

3.2.1. Thuật toán đuổi khớp - MP

Matching Pursuit (MP) - thuật toán đuổi khớp, được đặt tên và giới thiệu với cộng đồng xử lý tín hiệu bởi Mallat và Zhang [63], là một thuật toán tham lam lặp đi lặp lại phân rã một tín hiệu thành một tổ hợp tuyến tính của các phần tử từ một bộ dữ liệu đã xây dựng. Trong thuật toán đuổi khớp, bộ dữ liệu này là ma trận lấy mẫu nén $\Phi \in \mathbb{R}^{M\times N}$ và có thể khôi phục lại tín hiệu thưa ban đầu x từ mẫu nén y và ma trận lấy mẫu.

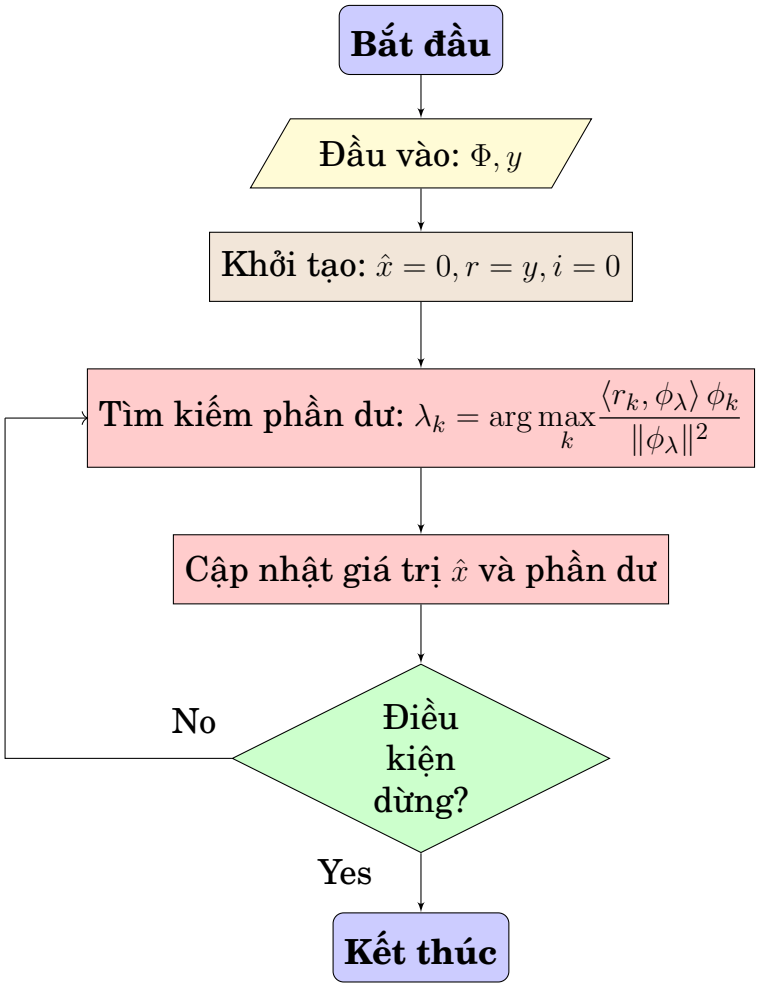
Thuật toán đuổi khớp có lưu đồ thực hiện được mô tả trong hình 3.1. Khối lượng tính toán chính trong thuật toán MP là tìm phần dư $r \in \mathbb{R}^M$ và phần còn lại đại diện cho phần chưa được biết đến của các mẫu nén. Ở mỗi lần lặp lại của thuật toán, một vector hay một cột của ma trận lấy mẫu nén sẽ được chọn mà thỏa mãn tiêu chí có tương quan lớn nhất đối với phần dư r:

$$\lambda_k = \arg\max_k \frac{< r_k, \phi_\lambda > \phi_k}{\|\phi_\lambda\|^2}. \tag{3.4}$$

Cột của ma trận được chọn từ biểu thức (3.4) sẽ là một biểu diễn tốt hơn của tín hiệu, khi đó một hệ số mới λ_k được cộng vào tín hiệu khôi phục trung gian. Do đó, phần dư và tín hiệu ước lượng sẽ được cập nhật lại tại bước này theo biểu thức sau:

$$\begin{aligned} r_k &= r_{k-1} - \frac{< r_{k-1}, \phi_{\lambda_k} > \phi_{\lambda_k}}{\|\phi_{\lambda_k}\|^2}, \\ \hat{x}_{\lambda_k} &= \hat{x}_{\lambda_k} + < r_{k-1}, \phi_{\lambda_k} >, \end{aligned} \tag{3.5}$$

và quá trình như vậy được lặp lại đến khi phần dư r trở nên nhỏ hơn một giá trị đặt trước và nó được gọi là tiêu chí dừng của thuật toán. Thuật toán đuổi khớp được trình bày như sau:



Hình 3.1: *Lưu đồ thuật toán đuổi khớp*

Thuật toán: Thuật toán đuổi khớp

Đầu vào: Ma trận đo Φ, vectơ mẫu nén y

Đầu ra: Tín hiệu gần đúng $\hat{x}$

Khởi tạo: $\hat{x}_0 = 0, r = y, i = 0$

Lặp:

1. $\lambda_k \leftarrow \arg\max_k \frac{\langle r_i, \phi_\lambda \rangle \phi_k}{\|\phi_\lambda\|^2}, \quad k = [1:N]$ {Tìm kiếm tín hiệu dư}
2. $\hat{x}_i \leftarrow \hat{x}_{i-1} + \langle r_i, \phi_{\lambda_k} \rangle$ {Cập nhật hệ số lớn nhất}
3. $r_i \leftarrow r_{i-1} - \frac{\langle r_i, \phi_{\lambda_k} \rangle \phi_{\lambda_k}}{\|\phi_{\lambda_k}\|^2}$ {Cập nhật phần dư}
4. $i \leftarrow i + 1$

Tiêu chí dừng

return $\hat{x} \leftarrow \hat{x}_i$

Kết thúc.

Mặc dù thuật toán đuổi khớp trực quan và có thể ước lượng chính xác tín hiệu, nhưng nó có hai nhược điểm chính: (i) không đảm bảo về lỗi khôi phục, nó không khai thác cấu trúc và đặc điểm từ ma trận lấy mẫu nén Φ; (ii) số

lần lặp cần thiết khá lớn. Độ phức tạp tính toán của thuật toán đuổi khớp là $\mathcal{O}(M \times N \times T)$ [33], trong đó T là số lần lặp của thuật toán đuổi khớp.

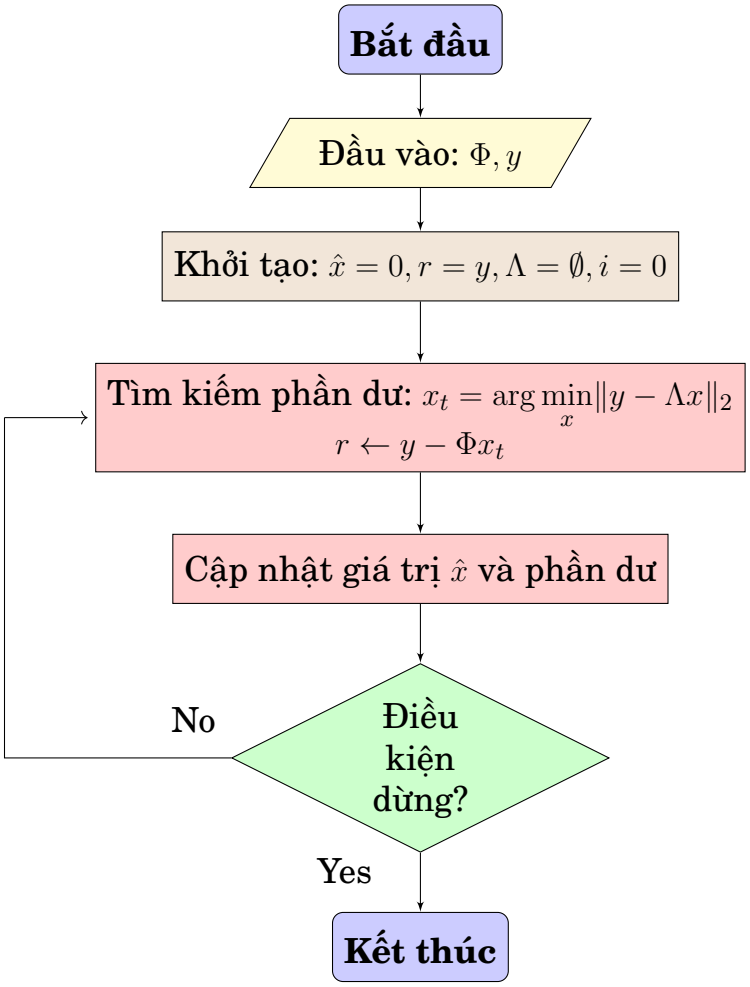
3.2.2. Thuật toán đuổi khớp trực giao - OMP

Thuật toán đuổi khớp không đảm bảo về lỗi khôi phục và độ phức tạp tính toán lớn nên không khả thi đối với nhiều ứng dụng trong thực tế [6], [64], [65], bởi sự phức tạp của thuật toán đuổi khớp tăng tuyến tính theo số lần lặp T. Một số các thuật toán sửa đổi dựa trên thuật toán đuổi khớp có thể giới hạn số lần lặp lại tối đa đối với thuật toán này và giúp nó khả thi hơn khi áp dụng trong thực tế. Ý tưởng trong thuật toán đuổi khớp trực giao là tại bất kỳ lần lặp thứ k nào, thay vì trừ đi phần tử được tìm kiếm trong ma trận lấy mẫu nén Φ có tương quan lớn nhất đối với phần dư r, tại bước này của thuật toán sẽ tính toán ước lượng bình phương tối thiểu của phần dư r cần tính toán lên ma trận con của ma trận lấy mẫu nén Φ. Bước này đảm bảo phần dư hoàn toàn trực giao với ma trận con được chọn, và các ma trận con sẽ chỉ được chọn một lần trong quá trình khôi phục. Do đó, đại lượng này đại diện tốt hơn cho phần dư chưa biết đến, nếu thuật toán OMP khôi phục một tín hiệu có $K - sparse$ thì nó có thể khôi phục chính xác sau K lần lặp. Nếu Λ là ma trận con được tạo thành bởi các cột của Φ đã chọn tại bước thời gian t, quá trình thực hiện có thể được mô tả như sau:

$$\begin{aligned} x_t &= \arg\min_x \|y - \Lambda x\|_2, \\ \hat{\alpha}_t &= \Phi x_t, \\ r_t &= y - \hat{\alpha}_t. \end{aligned} \tag{3.6}$$

Các bước này được lặp lại cho đến khi kết quả của thuật toán là hội tụ. Thuật toán này được gọi là thuật toán đuổi khớp trực giao (OMP) [79]. Tropp và Gilbert [96] đã chứng minh rằng thuật toán OMP có thể được sử dụng để khôi phục một tín hiệu thưa với xác suất cao từ các mẫu nén. Thuật toán hội tụ nhiều nhất sau K lần lặp, trong đó K là độ thưa của tín hiệu được lấy mẫu

nén, nhưng độ phức tạp tính toán của thuật toán đuổi khớp trực giao được bổ sung ở mỗi lần lặp. Do đó, độ phức tạp tín toán của thuật toán OMP có thể được biểu diễn là $\mathcal{O}(M \times N \times K)$. Lưu đồ của thuật toán đuổi khớp trực giao OMP được mô tả như trong hình 3.2.



Hình 3.2: *Lưu đồ thuật toán đuổi khớp trực giao*

Thuật toán: Thuật toán đuổi khớp trực giao

Đầu vào: Ma trận đo Φ, vectơ đo lường y

Đầu ra: Tín hiệu khôi phục $\hat{x}$

Khởi tạo: $x^0 \leftarrow 0,\ r^0 \leftarrow y,\ \Lambda^0 = \emptyset,\ i \leftarrow 0$

Lặp:

1. $b^{i+1} \leftarrow \Phi^T r^i$ {Tìm kiếm tín hiệu dư}
   $\lambda = \underset{j \notin \Lambda^i}{\operatorname{argmax}} |b_j^{i+1}|$
   $\Lambda^{i+1} \leftarrow \Lambda^i \cup \{\lambda\}$
2. $x^{i+1} \leftarrow \arg\min_x \|y - \Lambda^{i+1} x^i\|_2$ {cập nhật tín hiệu}
3. $r^{i+1} \leftarrow y - \Phi x^{i+1}$ {cập nhật phần dư}
4. $i \leftarrow i + 1$

Tiêu chí dừng

return $\hat{x} \leftarrow x^i$

Kết thúc.

Mặc dù thuật toán OMP có tốc độ khôi phục nhanh và đã được chứng minh là có thể khôi phục lại tín hiệu thưa một cách chính xác, nhưng các điều kiện đảm bảo để thuật toán OMP có thể khôi phục kém hơn so với các thuật toán dựa trên tối ưu hóa lồi [25], [96]. Đặc biệt, các điều kiện đảm bảo khôi phục lại tín hiệu thưa là không đồng nhất, tức là không thể chỉ ra rằng một phép đo duy nhất với ma trận lấy mẫu nén có số hàng $M = CK \log N$ có thể được sử dụng để khôi phục lại mọi tín hiệu có độ thưa K, mặc dù có thể đạt được những điều kiện đảm bảo thống nhất như vậy khi chấp nhận số phép đo nhiều hơn [25]. Một vấn đề khác đối với thuật toán OMP là khả năng chống nhiễu, không có mối quan hệ giữa biên độ của nhiễu trong tín hiệu sau khôi phục bằng thuật toán OMP với biên độ của nhiễu đầu vào. Tuy nhiên, thuật toán OMP vẫn là một thuật toán hiệu quả để khôi phục lại các tín hiệu được lấy mẫu nén, đặc biệt là khi tín hiệu có độ thưa K thấp [73].

3.2.3. Thuật toán lấy mẫu nén đuổi khớp - CoSaMP

Các thuật toán đuổi khớp tham lam (MP và OMP) làm giảm độ phức tạp tính toán trong quá trình khôi phục lại tín hiệu thưa so với với các thuật toán khôi phục dựa trên tối ưu hóa, nhưng làm mất đi các điều kiện đảm bảo đồng nhất trong quá trình phục hồi và yêu cầu ma trận lấy mẫu nén phải có số

hàng lớn.

Trong khoảng thời gian này đã có các nghiên cứu để phát triển các thuật toán tham lam như các thuật toán OMP chính quy [73], Theo đuổi đối sánh lấy mẫu nén (CoSaMP) [72] và Thuật toán đuổi khớp trong không gian con [23] nhằm thu hẹp khoảng cách giữa tính đồng nhất và độ phức tạp. Tính chất giới hạn đẳng trị RIP được phát triển cho các thuật toán dựa trên tối ưu hóa ℓ_1 cũng đóng vai trò trung tâm đối với các thuật toán tham lam này. Do đó, nếu ma trận Φ thỏa mãn tính chất RIP bậc K, điều này thể hiện rằng mọi tập con gồm K cột của ma trận là một xấp xỉ trực giao. Tính chất này được sử dụng để chứng minh khả năng hội tụ tốt của các các thuật toán lặp lại tham lam.

CoSaMP là một thuật toán tham lam cải tiến tiếp cận theo hướng trên. Cải tiến mới trong thuật toán CoSaMP không giống như MP, OMP và StOMP, các chỉ số mới được thêm vào trong tín hiệu ước lượng và bị xóa khỏi tập hợp các chỉ số đã chọn hiện tại. Ngược lại, các thuật toán theo đuổi tham lam như MP và OMP lại xuất phát từ thực tế rằng một chỉ số đã chọn (một vector thành phần trong bộ dữ liệu chọn trước từ ma trận lấy mẫu nén Φ vẫn còn trong biểu diễn của tín hiệu cho đến bước cuối cùng). Thuật toán CoSaMP được mô tả như trong hình 3.3.

Thuật toán: Thuật toán đuổi khớp mẫu nén CoSaMP

Đầu vào: Ma trận đo Φ, vectơ đo lường y, tín hiệu thưa K

Đầu ra: Tín hiệu gần đúng $\hat{x}$

Khởi tạo: $x^0 = 0, \quad r = y, \quad i = 0$

Thực hiện các bước sau cho đến khi đáp ứng tiêu chí dừng:

Gán: $e \leftarrow \Phi^* r$ {ước lượng phần dư}

Tìm kiếm: $\Omega \leftarrow \text{supp}(T(e, 2K))$ {ước lượng thành phần tín hiệu còn lại}

Kết hợp: $T \leftarrow \Omega \cup \text{supp}(x^{i-1})$ {hợp nhất}

Ước lượng: $b|_T \leftarrow \Phi_T^\dagger y, \quad b|_{T^c} \leftarrow 0$

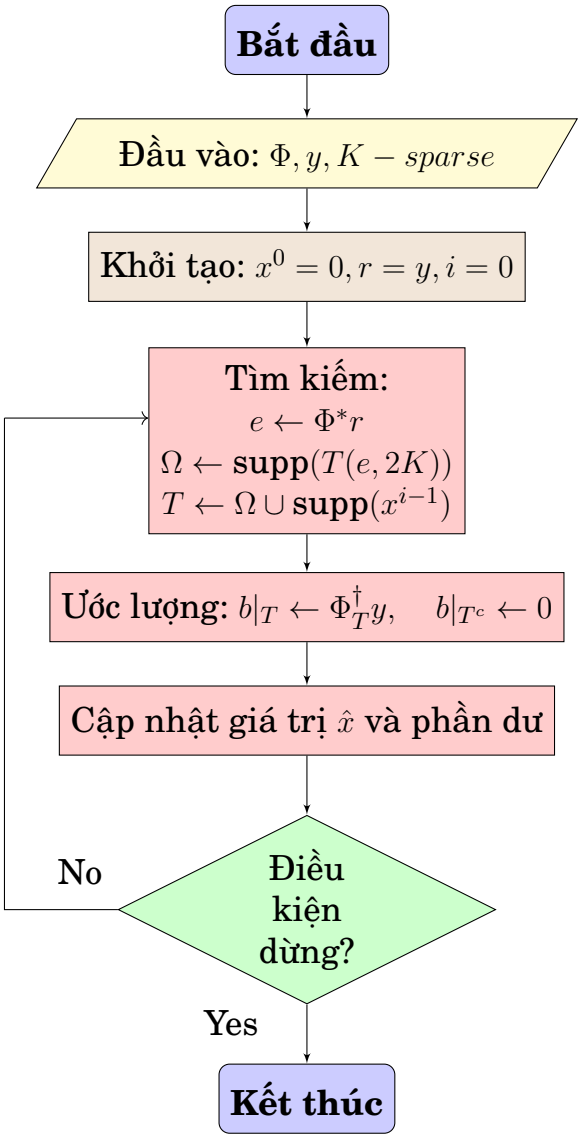
Cập nhật: $x^i \leftarrow T(b, K)$ {tín hiệu ước lượng}

$r \leftarrow y - \Phi x^i$ {update phần dư}

$i \leftarrow i + 1$

return $\hat{x} \leftarrow x^i$

Kết thúc.



Hình 3.3: *Lưu đồ thuật toán CoSaMP*

Như được trình bày trong [72], các vấn đề tính toán quan trọng đối với CoSaMP là quá trình tính toán với phần dư của tín tín hiệu, và phương pháp được sử dụng là các phép chiếu trên không gian con trong các bước ước lượng tín hiệu. Theo các giả định chung, độ phức tạp tính toán của thuật CoSaMP có thể được biểu diễn là $\mathcal{O}(M \times N)$, và nó không phụ thuộc vào mức độ thưa của tín hiệu ban đầu. Điều này thể hiện sự cải tiến của thuật toán CoSaMP so với cả các thuật toán tham lam cũng như các thuật toán dựa trên tối ưu hóa lồi.

Trong khi thuật toán CoSaMP được cho là đại diện cho các thuật toán hiện

đại có hiệu suất cao trong khôi phục lại tín hiệu được lấy mẫu nén, nó vẫn còn tồn tại nhược điểm là: thuật toán yêu cầu phải biết trước về độ thưa thớt K của tín hiệu cần lấy mẫu. Nếu lựa chọn không chính xác về độ thưa của tín hiệu đầu vào có thể dẫn đến các điều kiện đảm bảo khôi phục kém hơn kể cả khi so với các thuật toán trước khi cải tiến như thuật toán OMP. Giới hạn ổn định của thuật toán CoSaMP gắn liền với lỗi khi lựa chọn tham số về độ thưa của tín hiệu cần lấy mẫu, và độ thưa của tín hiệu chủ yếu được đưa ra bởi các phương pháp thống kê.

3.3. Thuật toán cải tiến DRMP

Từ nghiên cứu và phân tích về số một thuật toán đã được trình bày ở trên, trong luận án trình bày về một thuật toán cải tiến dựa trên các thuật toán kiểu tham lam. Vấn đề khôi phục tín hiệu được lấy mẫu nén được đưa về bài toán tìm cực tiểu với các điều kiện ràng buộc đối với tín hiệu x:

$$\min_{x} \ell(x), \text{ với điều kiện } \|x\|_{0,D} \leq K, \tag{3.7}$$

ở đây, $\|x\|_{0,D}$ là một ma trận định chuẩn, chuẩn này sẽ được sử dụng để đo lường tính chất thưa của tín hiệu x mà có liên quan đến ma trận lấy mẫu D, K là tham số để xác định mức độ thưa của ước lượng $\ell(x)$ trong đó nó được coi là hàm mục tiêu và được sử dụng để đo lường giá trị tương quan giữa các mẫu nén và thông số cần ước lượng.

Điều kiện đối với $\ell(x)$ phải là một hàm trơn, nhưng không cần thiết phải là một hàm lồi, khi đó bài toán khôi phục có thể giải quyết bằng các kỹ thuật ước lượng và tối ưu. Ví dụ, trong bài toán lấy mẫu nén tuyến tính hồi quy [21] nếu muốn khôi phục lại tín hiệu thưa $\hat{x} \in \mathbb{R}^N$ từ các mẫu nén $y \in \mathbb{R}^M : y = \Phi\hat{x} + w$ với Φ là ma trận lấy mẫu nén có kích thước là $M \times N$ và w là một vector nhiễu cộng. Hàm mục tiêu $\ell(x)$ được mô tả dưới dạng một hàm số bậc hai: $\ell(x) = \|y - \Phi x\|_2^2$ theo định chuẩn $\ell_0 - norm$.

Các thuật toán được nghiên cứu trong lĩnh vực lấy mẫu nén [14], [18], [72],

[96] đều hướng tới ứng dụng trong thực tế. Việc lựa chọn thuật toán nào phụ thuộc vào từng ứng dụng cụ thể và phụ thuộc vào ma trận lấy mẫu nén cũng như các chỉ số về mức độ thưa và độ lớn của vector tín hiệu cần lấy mẫu. Từ các phân tích và tìm hiểu về các thuật toán khôi phục luận án tập trung vào phân tích, đề xuất một thuật toán cải tiến dựa trên thuật toán đuổi khớp MP. Thuật toán được xây dựng với mục tiêu giảm khối lượng tính toán và lỗi khôi phục xảy ra tại các bước so với thuật toán gốc MP. Phần thực nghiệm cũng sử dụng các kết quả đã được công bố với thuật toán MP để tiến hành mô phỏng, so sánh và đánh giá dựa trên một số tiêu chí cơ bản được trình bày trong mục (3.1).

Trong thuật toán đuổi khớp mục (3.2.1) tại các bước tìm kiếm phần dư và cập nhật giá trị, thuật toán này cần phải tìm một tập hợp con Γ của các vector cột trong tập ma trận lấy mẫu D mà thỏa mãn điều kiện tích vô hướng của một vector đối với tập mở rộng D_Γ có giá trị cực đại. Tập con này thường khó tính toán, đặc biệt khi kích thước của D lớn. Lý do là cần phải tìm kiếm tất cả sự kết hợp có thể có các tập con của D để có thể tìm ra một lựa chọn tốt nhất [26]. Hơn thế nữa thuật toán đuổi khớp không khai thác cấu trúc và đặc điểm của tập D.

Trên thực tế, để khắc phục những hạn chế trên cần chú ý đến các tính chất đặc biệt của tập D. Ví dụ, khi D là một ma trận trực giao, Γ có thể được chọn đơn giản bởi điều kiện đẳng trị dựa trên không gian được mở rộng bởi tất cả các cột của D và chọn $2K$ cột lớn nhất dựa trên điều kiện đẳng trị này. Ngoài ra, khi thực hiện khôi phục ma trận cấp thấp, U là ma trận và Γ có thể là được tính bằng cách lấy các giá trị kỳ dị (SVD) của U và lựa chọn các vector kì dị có độ lớn $2K$ về bên trái và phải mà liên quan lớn nhất đến điểm giá trị kì dị $2K$ [75]. Trong phần tiếp theo, luận án đề xuất thuật toán cải tiến từ thuật toán gốc MP gọi là DRMP, thuật toán cải tiến này giảm khối lượng tính toán và giảm lỗi xảy ra ở mỗi bước trong quá trình khôi phục tín hiệu được lấy mẫu nén trên cơ sở xem xét một số điều kiện đặc biệt đối với tập D.

3.3.1. Xây dựng thuật toán DRMP

Thuật toán DRMP được xây dựng với các bước tương tự như thuật toán đuổi khớp trong (3.2.1). Sự khác biệt giữa hai thuật toán là ở bước tìm kiếm phần dư và bước cập nhật giá trị. Trong đó, thuật toán DRMP đã thay thế điều kiện đẳng trị bằng giá trị tích vô hướng của nó.

Trong mô hình lấy mẫu nén x là tín hiệu được lấy mẫu nén và được biểu diễn bởi công thức $x = \sum_{i \in R} \alpha_i d_i$, trong đó α_i là thừa số của x trong tập D, và gọi không gian con chứa $\hat{x}$ trong biểu thức (3.7) là T. Đối với một vectơ x đã cho, đặt $P_{D_T} x$ là phép chiếu trực giao của x lên không gian được tạo bởi các cột của ma trận D_T

$$P_{D_T} x = \underset{u \in D_T}{\operatorname{argmin}} \|x - u\|_2, \tag{3.8}$$

ở đây định chuẩn $\ell_2 - norm$ được sử dụng để tìm kiếm x. Ký hiệu C_{D_Γ} là không gian được tạo bởi các cột của D_Γ, khi đó

$$C_{D_\Gamma} = \{x : x = D_\gamma \alpha, \alpha \in \mathbb{R}^{|\Gamma|}\}, \tag{3.9}$$

ở đây, C_{D_Γ} là một tập lồi.

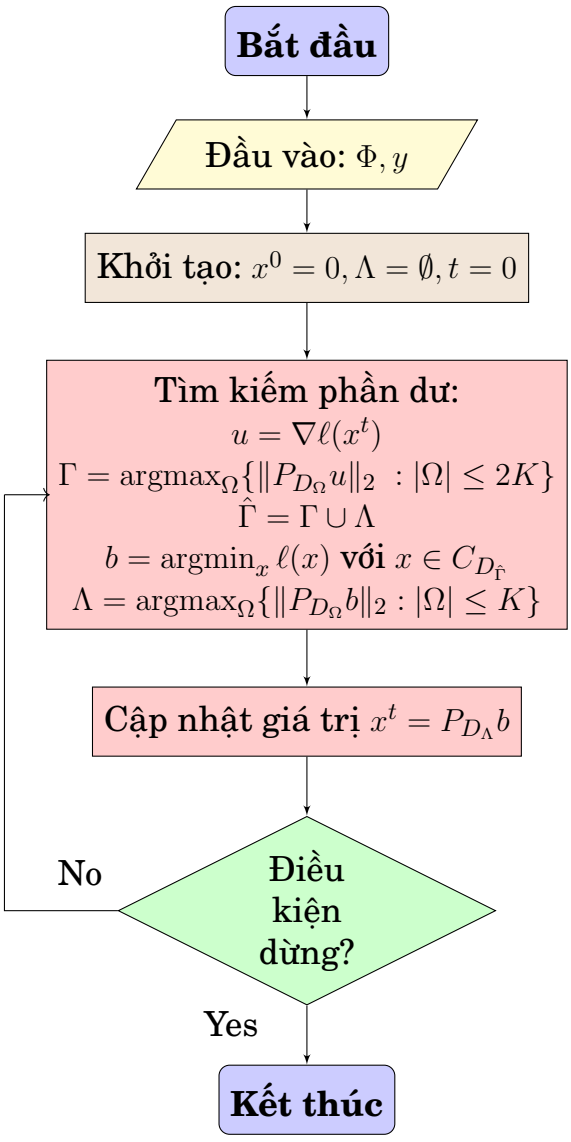
Thuật toán DRMP được xây dựng với các bước gần như tương đương với thuật toán đuổi khớp MP. Để thuật toán DRMP hội tụ cần phải có điều kiện đối với tập D. Giả sử rằng D là hữu hạn và D thỏa mãn tính chất giới hạn đẳng trị RIP [17]:

$$(1 - \delta_K) \|\alpha\|_2^2 \leq \|D\alpha\|_2^2 \leq (1 + \delta_K) \|\alpha\|_2^2, \tag{3.10}$$

với mỗi vectơ α tại K. Trong trường hợp D là ma trận trực giao với kích thước là $n \times n$, thì thuật toán DRMP sẽ tương đương với thuật toán gốc MP.

Sự khác biệt giữa hai thuật toán là ở bước tìm kiếm phần dư và cập nhật giá trị. Tại bước tìm kiếm không gian mở rộng Γ xây dựng từ D được tính thông qua việc lấy biến đổi *gradient* của hàm mất mát ở mỗi lần lặp và chọn

không gian con theo hướng tối đa hóa giá trị của biến đổi *gradient*. Tại bước cập nhật định chuẩn $\ell_2 - norm$ được sử dụng trên không gian mở rộng C_{D_Γ} thay cho việc tính tích vô hướng của toàn bộ các cột trong ma trận lấy mẫu nén như đối với thuật toán MP. Thuật toán DRMP được mô tả như trong sơ đồ hình 3.4.



Hình 3.4: *Lưu đồ thuật toán DRMP*

Thuật toán: Đuổi khớp cải tiến DRMP

Đầu vào: K và điều kiện dừng.

Đầu ra: x^t

Khởi tạo: $\Lambda = \varnothing, x^0,$ và $t = 0.$

Lặp:

Gán: $u = \nabla \ell(x^t)$

Tìm kiếm: $\Gamma = \text{argmax}_\Omega \{\|P_{D_\Omega} u\|_2 \ : |\Omega| \leq 2K\}$

Kết hợp: $\hat{\Gamma} = \Gamma \cup \Lambda$

Ước lượng: $b = \text{argmin}_x \, \ell(x)$ với $x \in C_{D_{\hat{\Gamma}}}$

Loại bỏ: $\Lambda = \text{argmax}_\Omega \{\|P_{D_\Omega} b\|_2 : |\Omega| \leq K\}$

Cập nhật: $x^{t+1} = P_{D_\Lambda} b$

$t = t + 1$

Tiêu chí dừng

Kết thúc.

Bảng 3.1 trình bày sự cải tiến của thuật toán DRMP so với thuật toán MP tại các bước tìm kiếm và cập nhật.

Trong phần (3.3.2) hiệu năng của thuật toán DRMP được đánh giá thông qua một mô hình toán học và chỉ ra rằng, đối với thuật toán được cải tiến, lỗi trong quá trình khôi phục lại tín hiệu lấy mẫu nén sẽ giảm dần sau mỗi một bước lặp, quá trình tính toán sẽ đơn giản hơn so với sử dụng giá trị tích vô hướng được tính toán từ các cột của ma trận lấy mẫu. Tuy nhiên, điều này chỉ có thể được áp dụng khi ma trận D thỏa mãn tính chất giới hạn đẳng trị RIP. Trong trường hợp, D là ma trận trực giao khi đó hai thuật toán có hiệu năng tương đương.

3.3.2. Hiệu năng của thuật toán DRMP

Để đánh giá hiệu quả của thuật toán DRMP trong phần này xây dựng một mô hình toán học với tham số là giá trị lỗi khôi phục của thuật toán xảy ra tại mỗi lần lặp. Gọi tín hiệu khôi phục $\hat{x}$ là nghiệm của biểu thức (3.7) và $\lambda(\hat{x}) \triangleq \max_i |\langle \nabla \ell(\hat{x}), d_i \rangle|$ là giá trị lỗi của tín hiệu khôi phục trung gian trong mỗi bước của thuật toán, lỗi ước lượng ở bước $(t+1)$ được ràng buộc bởi:

$$\left\|x^{t+1} - \hat{x}\right\|_2 \leq \gamma \left\|x^t - \hat{x}\right\|_2 + \lambda(\hat{x})\sqrt{K}\left(\frac{\sqrt{2}+1}{\rho^-_{4K}} + \frac{3}{2\sqrt{\rho^-_{4K}\rho^+_{4K}}}\right), \tag{3.11}$$

Bảng 3.1: *Bảng so sánh thuật toán cải tiến và thuật toán gốc MP*

	MP	DRMP
Điều kiện đầu vào	Không tính đến điều kiện của ma trận lấy mẫu	Ma trận lấy mẫu phải thỏa mãn điều kiện giới hạn đẳng trị RIP
Bước tìm kiếm	$\lambda_k = \arg\max_k \frac{<r_k, \phi_\lambda>\phi_k}{\|\phi_\lambda\|^2}$. Thuật toán MP tìm kiếm các thành phần của tín hiệu từ các cột của ma trận lấy mẫu giữa trên tương quan lớn nhất với vector mẫu nén.	$u = \nabla\ell(x^t)$ $\Gamma = \text{argmax}_\Omega\{\|P_{D_\Omega}u\|_2 \ : \|\Omega| \leq 2K\}$ $b = \text{argmin}_x\, \ell(x)$ với $x \in C_{D_{\hat{\Gamma}}}$ Tìm kiếm không gian mở rộng Γ xây dựng từ D được tính thông qua việc lấy biến đổi *gradient* của hàm mất mát ở mỗi lần lặp và chọn không gian con theo hướng tối đa hóa giá trị của biến đổi *gradient*.
Bước cập nhật	$r_k = r_{k-1} - \frac{< r_{k-1}, \phi_{\lambda_k} > \phi_{\lambda_k}}{\|\phi_{\lambda_k}\|^2}$ $\hat{x}_{\lambda_k} = \hat{x}_{\lambda_k} + < r_{k-1}, \phi_{\lambda_k} >$ Giá trị phần dư tìm được sẽ được cộng vào tín hiệu khôi phục trung gian, đồng thời phần dư mới cũng được cập nhật ở bước này.	$\Lambda = \text{argmax}_\Omega\{\|P_{D_\Omega}b\|_2 : \|\Omega| \leq K\}$ $\hat{\Gamma} = \Gamma \cup \Lambda$ $x^t = P_{D_\Lambda}b$ Sử dụng định chuẩn $\ell_2 - norm$ trên không gian mở rộng C_{D_Γ} thay cho việc tính tích vô hướng của toàn bộ các cột trong ma trận lấy mẫu nén như đối với thuật toán MP.

trong đó, γ là tỉ số suy giảm sau các bước và

$$\gamma \triangleq \sqrt{2\frac{1+\delta}{(1-\delta)^2}\frac{\rho_K^+((1+\delta)\rho_{2K}^+ - (1-\delta)\rho_{2K}^-)}{(\rho_{4K}^-)^2}}. \tag{3.12}$$

Tín hiệu lấy mẫu x là một vector $K - sparse$ trong tập D, và gọi không gian con chứa x là T, ký hiệu là $x = D_T\alpha_T$. Đối với bất kỳ vector y nào có kích thước

là n, có:

$$\begin{aligned}|\langle x,y\rangle| &= \left|\left\langle \sum_{i\in T}\alpha_i d_i, y\right\rangle\right| = \left|\sum_{i\in T}\langle \alpha_i d_i, y\rangle\right| \le \sum_{i\in T}|\alpha_i|\,|\langle d_i,y\rangle| \\ &= \max_{i\in T}|\langle d_i,y\rangle|\,\|\alpha_T\|_1 \le \max_{i\in T}|\langle d_i,y\rangle|\sqrt{K}\,\langle \alpha_T\rangle_2 \\ &\le \max_{i\in T}|\langle d_i,y\rangle|\sqrt{K}\frac{1}{(1-\delta_K)}\|D_T\alpha_T\|_2,\end{aligned} \tag{3.13}$$

trong đó, bất đẳng thức cuối cùng là dựa trên tập D thỏa mãn tính chất giới hạn đẳng trị RIP. Suy ra:

$$|\langle x,y\rangle| \le \max_{i\in T}|\langle d_i,y\rangle|\,\frac{\sqrt{K}}{\sqrt{(1-\delta_K)}}\|x\|_2. \tag{3.14}$$

Giả sử b là một vector thu được từ bước ước lượng trong thuật toán DRMP tại lần lặp lại thứ t, và $\hat{\Gamma}$ là tập hợp đạt được ở bước kết hợp thứ t. Tín hiệu khôi phục được tại bước này $\hat{x} = \hat{x}_{T\setminus\hat{\Gamma}} + \hat{x}_{T\cap\hat{\Gamma}}$ và sai số ước lượng là:

$$\|b-\hat{x}\|_2 \le \sqrt{\frac{\rho_K^+}{\rho_{4K}^-}}\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2 + \frac{5\lambda\sqrt{K}}{2\rho_{4K}^-\sqrt{1-\delta_{4K}}}. \tag{3.15}$$

Để giải thích cho biểu thức (3.15), áp dụng điều kiện $D-RSC$ (***D** - restricted strong convexity*) và $D-RSS$ (***D** - restricted strong smoothness*) được trình bày trong [3], [74] và biểu thức (3.14), nhận được:

$$\begin{aligned}\rho_{4K}^-\|b-\hat{x}\|_2^2 + \langle\nabla\ell(\hat{x}), b-\hat{x}\rangle &\le^{(a)} \ell(b)-\ell(\hat{x}) \le^{(b)} \ell(\hat{x}_{T\cup\hat{\Gamma}})-\ell(\hat{x}) \\ &= \ell(\hat{x}_{T\cap\hat{\Gamma}})-\ell(\hat{x})-\left\langle\nabla\ell(\hat{x}),\hat{x}_{T\cup\hat{\Gamma}}-\hat{x}\right\rangle+\left\langle\nabla\ell(\hat{x}),\hat{x}_{T\cup\hat{\Gamma}}-\hat{x}\right\rangle \\ &\le^{(c)} \rho_K^+\left\|\hat{x}_{T\setminus\hat{\Gamma}}-\hat{x}\right\|_2^2+\left\langle\nabla\ell(\hat{x}),\hat{x}_{T\setminus\hat{\Gamma}}\right\rangle \\ &\le^{(d)} \rho_K^+\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|^2+\max_{i\in T\setminus\hat{\Gamma}}|\langle\nabla\ell(\hat{x}),d_i\rangle|\frac{\sqrt{K}}{\sqrt{1-\delta_K}}\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2 \\ = \rho_K^+\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2^2+\frac{\lambda\sqrt{K}}{\sqrt{1-\delta_K}}\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2 &\le \rho_K^+\left(\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2+\frac{\lambda\sqrt{K}}{2\rho_K^+\sqrt{1-\delta_K}}\right)^2.\end{aligned} \tag{3.16}$$

Bất đẳng thức (a) đạt được từ điều kiện $D-RSC$; bất đẳng thức (b) đạt được theo thuật toán, $\ell(b)$ là điểm cực tiểu của hàm mục tiêu $\ell(x)$ với $x\in span(D_{\hat{\Gamma}})$, bất đẳng thức (c) đạt được từ các điều kiện giả định của $D-RSS$. Cuối cùng, bất đẳng thức (d) đạt được từ biểu thức (3.14) đã nêu trên. Đồng thời, có:

$|supp_D(b-\hat{x})| \leq |\hat{\Gamma} \cup T| \leq 4K$ về bên trái của bất đẳng thức có thể được giới hạn bởi áp dụng điều kiện trên:

$$\begin{aligned}\rho^-_{4K} \|b-\hat{x}\|_2^2 + \langle \nabla \ell(\hat{x}), b-\hat{x}\rangle &\geq \rho^-_{4K} \|b-\hat{x}\|_2^2 - \lambda \frac{\sqrt{4K}}{\sqrt{1-\delta_{4K}}} \|b-\hat{x}\|_2 \\ &\geq \rho^-_{4K} \left(\|b-\hat{x}\|_2 - \frac{\lambda\sqrt{4K}}{2\rho^-_{4K}\sqrt{1-\delta_{4K}}} \right)^2 - \frac{\lambda^2 K}{\rho^-_{4K}(1-\delta_{4K})}.\end{aligned} \tag{3.17}$$

Kết hợp các bất đẳng thức trên, thu được:

$$\begin{aligned}\left(\|b-\hat{x}\|_2 - \frac{\lambda\sqrt{K}}{\rho^-_{4K}\sqrt{1-\delta_{4K}}} \right)^2 \leq \frac{\rho^+_K}{\rho^-_{4K}} \left(\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2 + \frac{\lambda\sqrt{K}}{2\rho^+_K\sqrt{1-\delta_K}} \right)^2 \\ + \frac{\lambda^2 K}{(\rho^-_{4K})^2(1-\delta_{4K})}.\end{aligned} \tag{3.18}$$

Rút gọn:

$$\begin{aligned}\|b-\hat{x}\|_2 - \frac{\lambda\sqrt{K}}{\rho^-_{4K}\sqrt{1-\delta_{4K}}} &\leq \sqrt{\frac{\rho^+_K}{\rho^-_{4K}}} \left(\left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2 + \frac{\lambda\sqrt{K}}{2\rho^+_K\sqrt{1-\delta_K}} \right) \\ + \frac{\lambda\sqrt{K}}{\rho^-_{4K}\sqrt{(1-\delta_{4K})}} \|b-\hat{x}\|_2 &\leq \sqrt{\frac{\rho^+_K}{\rho^-_{4K}}} \left\|\hat{x}_{T\setminus\hat{\Gamma}}\right\|_2 + \frac{5\lambda\sqrt{K}}{2\rho^-_{4K}\sqrt{(1-\delta_{4K})}},\end{aligned} \tag{3.19}$$

kết quả cuối cùng chính là biểu thức (3.15).

Kí hiệu R là hỗ (giá) của vector $\Delta \triangleq x^t - \hat{x}$ tới tổ hợp D và $\lambda \triangleq \max_i |\langle \nabla \ell(\hat{x}), d_i\rangle|$ rút ra biểu thức sau:

$$\begin{aligned}\sqrt{\frac{\rho^+_{2K}-\rho^-_{2K}}{2\rho^-_{4K}}} \left\|x^{t-1}-\hat{x}\right\|_2 + \frac{\lambda\sqrt{2K}}{\rho^-_{4K}} \left\|\Delta_{R\setminus\Gamma}\right\|_2 \leq \sqrt{\frac{1+\delta}{(1-\delta)^2}\frac{(1+\delta)\rho^+_{2K}-(1-\delta)\rho^-_{2K}}{2\rho^-_{4K}}} \|\Delta\|_2 \\ + \lambda\sqrt{K}\frac{1+\delta}{1-\delta}\left(\frac{1}{\sqrt{2}\rho^-_{4K}} + \frac{1}{\sqrt{2\rho^-_{4K}}} \right),\end{aligned} \tag{3.20}$$

Để giải thích cho biểu thức (3.20), kí hiệu $\Delta = \hat{x} - x^t$ và $supp_D(\Delta) = R$ có

$\Delta = \sum_{i \in R} \beta_i d_i = \Delta_{R \cap \Gamma} + \sum_{i \in R \backslash \Gamma} \beta_i d_i$. Áp dụng điều kiện $D - RSC$ đối với x^t, có:

$$\begin{aligned} \ell(x^t + \Delta) - \ell(x^t) - \rho_{2K}^{-} \|\Delta\|_2^2 &\geq \left\langle \nabla \ell(x^t), \Delta \right\rangle = \left\langle \nabla \ell(x^t), \sum_{i \in R} \beta_i d_i \right\rangle \\ &= \left\langle \nabla \ell(x^t), \Delta_{R \cap \Gamma} \right\rangle + \sum_{i \in R \backslash \Gamma} \left\langle \nabla \ell(x^t), \beta_i d_i \right\rangle \\ &\geq \left\langle \nabla \ell(x^t), \Delta_{R \cap \Gamma} \right\rangle - \sum_{i \in R \backslash \Gamma} |\beta_i| \left\langle \nabla \ell(x^t), d_i \right\rangle, \end{aligned} \tag{3.21}$$

các tính chất của tập con R nhỏ hơn $2K$, trong khi đó $|\Gamma| = 2K$. Điều này suy ra $|R \backslash \Gamma| \leq |\Gamma \backslash R|$. Kết hợp với cấu trúc tập hợp con Γ, có:

$$\left| \left\langle \nabla \ell(x^t), d_i \right\rangle \right| \leq \left| \left\langle \nabla \ell(x^t), d_j \right\rangle \right|, \tag{3.22}$$

với $i \in R \backslash \Gamma$ và $j \in \Gamma \backslash R$. Đặt vectơ $g \in \mathbb{R}^p$ thỏa mãn các điều kiện sau:

- Các phần tử nằm ngoài tập hợp $\Gamma \backslash R$ khi đó được thiết lập bằng 0.
- Đối với mỗi phần tử $i \in R \backslash \Gamma$, chọn một phần tử $j \in \Gamma \backslash R$ và đặt:

$$g_j = -|\beta_i| \frac{\left\langle \nabla \ell(x^t), d_i \right\rangle}{|\langle \nabla \ell(x^t), d_i \rangle|}. \tag{3.23}$$

- Các phần tử trong tập hợp $\Gamma \backslash R$ mà không có được chọn bởi các bước ở trên khi đó được đặt bằng 0. Có tối đa $|\Gamma \backslash R| - |R \backslash \Gamma|$ hệ số tại bước này.

Do đó:

$$\sum_{i \in R \backslash \Gamma} \beta_i^2 = \sum_{j \in \Gamma \backslash R} g_j^2. \tag{3.24}$$

Đối với mỗi cặp số, $i \in R \backslash \Gamma$ và $j \in \Gamma \backslash R$ đã chọn trong biểu thức (3.23), có: $|g_j| = |\beta_i|$. Rút gọn biểu thức:

$$-|\beta_j| \left| \left\langle \nabla \ell(x^t), d_i \right\rangle \right| \geq -|\beta_j| \left| \left\langle \nabla \ell(x^t), d_j \right\rangle \right| = \left\langle \nabla \ell(x^t), g_j d_j \right\rangle. \tag{3.25}$$

Ở đây, biểu thức (3.25) tuân theo việc xây dựng từ $g_{\Gamma\setminus R}$. Do đó có:

$$-\sum_{j\in R\setminus\Gamma}\left\langle\nabla\ell(x^t),g_jd_j\right\rangle\geq\sum_{j\in\Gamma\setminus R}\left\langle\nabla\ell(x^t),g_jd_j\right\rangle. \tag{3.26}$$

Kết hợp với biểu thức (3.21) có:

$$\begin{aligned}\ell(x^t+\Delta)-\ell(x^t)-\rho_{2k}^-\|\Delta\|_2^2&\geq\left\langle\nabla\ell(x^t),\Delta_{R\cap\Gamma}\right\rangle+\sum_{j\in\Gamma\setminus R}\left\langle\nabla\ell(x^t),g_jd_j\right\rangle\\&=\left\langle\nabla\ell(x^t),\Delta_{R\cap\Gamma}\right\rangle+\left\langle\nabla\ell(x^t),\sum_{j\in\Gamma\setminus R}g_jd_j\right\rangle=\left\langle\nabla\ell(x^t),z\right\rangle.\end{aligned} \tag{3.27}$$

Đặt $z\triangleq\Delta_{R\cap\Gamma}+\sum_{j\in\Gamma\setminus R}g_jd_j$, trong trong đó: z là vectơ rời rạc của tập D và giá trị là Γ. Vế bên phải của bất đẳng thức là giới hạn trên $D-RSC$.

$$\left\langle\nabla\ell(x^t),z\right\rangle\geq\ell(x^{t-1}+z)-\ell(x^t)-\rho_{2K}^+\|z\|_2^2. \tag{3.28}$$

Do đó có:

$$\rho_{2K}^+\|z\|_2^2-\rho_{2K}^-\|\Delta\|_2^2\leq\ell(x^t+z)-\ell(x^{t-1}+\Delta)=\ell(x^t+z)-\ell(x^*). \tag{3.29}$$

Từ định nghĩa của z và điều kiện tập hợp D thỏa mãn tính chất giới hạn đẳng trị RIP:

$$\begin{aligned}\|z\|_2^2=\left\|\sum_{R\cap\Gamma}\beta_id_i+\sum_{j\in\Gamma\setminus R}g_jd_j\right\|_2^2&\leq(1+\delta_{2K})\left(\sum_{R\cap\Gamma}\beta_i^2+\sum_{j\in\Gamma\setminus R}g_j^2\right)\\&=(1+\delta_{2K})\left(\sum_{R\cap\Gamma}\beta_i^2+\sum_{j\in R\setminus\Gamma}\beta_i^2\right)=(1+\delta_{2K})\left(\sum_{i\in R}\beta_i^2\right)\\&\leq\frac{1+\delta_{2K}}{1-\delta_{2K}}\left\|\sum_{i\in R}\beta_id_i\right\|_2^2=\frac{1+\delta_{2K}}{1-\delta_{2K}}\|\Delta\|_2^2.\end{aligned} \tag{3.30}$$

Do đó, vế bên trái của biểu thức (3.29) được giới hạn bởi:

$$\rho_{2K}^+\|z\|_2^2-\rho_{2K}^-\|\Delta\|_2^2\leq\left(\frac{1+\delta_{2K}}{1-\delta_{2K}}\rho_{2K}^+-\rho_{2K}^-\right)\|\Delta\|_2^2, \tag{3.31}$$

về bên phải của biểu thức (3.29) được giới hạn dưới bởi $D - RSC$.

$$\ell(x^t + z) - \ell(x^*) \geq \langle \nabla \ell(\hat{x}), z - \Delta \rangle + \rho_{4K}^- \|z - \Delta\|_2^2 . \tag{3.32}$$

Từ $supp_D(z) = \Gamma$ và $supp_D(\Delta) = R$, có $supp_D(z-\Delta) \in R \cap \Gamma$ và $|supp_D(z-\Delta)| \leq 4K$.

Biểu diễn $z - \Delta = D_{R\cap\Gamma}\alpha_{R\cap\Gamma} = \sum_{i\in R\cap\Gamma} d_i\alpha_i$, có:

$$\begin{aligned}
&\ell(x^t + z) - \ell(x^*) \geq -\sum_{i\in R\cap\Gamma} \alpha_i \langle \nabla L(x^*), d_i \rangle + \rho_{4K}^- \|z - \Delta\|_2^2 \\
&\geq -\lambda \sum_{i\in R\cap\Gamma} |\alpha_i| + \rho_{4K}^- \|z - \Delta\|_2^2 \geq -\lambda\sqrt{4K} \|\alpha_{R\cap\Gamma}\|_2 + \rho_{4K}^- \|z - \Delta\|_2^2 \\
&\geq -\lambda\sqrt{4K}\frac{1}{\sqrt{1-\delta_{4K}}} \|D_{R\cap\Gamma}\alpha_{R\cap\Gamma}\|_2 + \rho_{4K}^- \|z - \Delta\|_2^2 \\
&\geq -\frac{\lambda\sqrt{4K}}{\sqrt{1-\delta_{4K}}} \|z - \Delta\|_2 + \rho_{4K}^- \|z - \Delta\|_2^2 \\
&= \rho_{4K}^- \left(\|z - \Delta\|_2 - \frac{\lambda\sqrt{K}}{\rho_{4K}^- \sqrt{1-\delta_{4K}}} \right)^2 - \frac{\lambda K}{\rho_{4K}^-}.
\end{aligned} \tag{3.33}$$

Từ kiến trúc của vectơ z và g, kết hợp với điều kiện RIP của D, có:

$$\begin{aligned}
&\geq (1-\delta_{4K}) \left(\sum_{j\in\Gamma\setminus R} g_j^2 + \sum_{i\in R\setminus\Gamma} \beta_i^2 \right) \|z - \Delta\|_2^2 \\
&= \left\| \sum_{j\in\Gamma\setminus R} g_j d_j - \sum_{i\in R\setminus\Gamma} \beta_i d_i \right\|_2^2 \\
&= 2(1-\delta_{4K}) \sum_{i\in R\setminus\Gamma} \beta_i^2 \geq 2\frac{1-\delta_{4K}}{1+\delta_{4K}} \left\| \sum_{i\in R\setminus\Gamma} \beta_i d_i \right\|_2^2 \\
&= 2\frac{1-\delta_{4K}}{1+\delta_{4K}} \left\| \Delta_{R\setminus\Gamma} \right\|_2^2 .
\end{aligned} \tag{3.34}$$

Ở đây, đẳng thức thứ hai trong biểu thức (3.34) thỏa mãn biểu thức (3.24). Kết hợp với bất đẳng thức trong biểu thức (3.34), có:

$$\ell(x^t + z) - \ell(x^*) \geq \rho_{4K}^- \left(\sqrt{2\frac{1-\delta_{4K}}{1+\delta_{4K}}} \left\| \Delta_{R\setminus\Gamma} \right\|_2 - \frac{\lambda\sqrt{K}}{\rho_{4K}^- \sqrt{1-\delta_{4K}}} \right)^2 - \frac{\lambda^2 K}{\rho_{4K}^-}. \tag{3.35}$$

Kết hợp kết quả ở trên với các biểu thức (3.29) và (3.30), có:

$$\begin{aligned}&\left(\frac{1+\delta_{4K}}{1-\delta_{4K}}\rho_{2K}^{+}-\rho_{2K}^{-}\right)\|\Delta\|_2^2\\ &\geq \rho_{4K}^{-}\left(\sqrt{2\frac{1-\delta_{4K}}{1+\delta_{4K}}}\left\|\Delta_{R\backslash\Gamma}\right\|_2-\frac{\lambda\sqrt{K}}{\rho_{4K}^{-}\sqrt{1-\delta_{4K}}}\right)^2-\frac{\lambda^2K}{\rho_{4K}^{-}}.\end{aligned} \tag{3.36}$$

Rút gọn:

$$\begin{aligned}\sqrt{2\frac{1-\delta_{4K}}{1+\delta_{4K}}}\left\|\Delta_{R\backslash\Gamma}\right\|_2 &\leq \frac{1}{\sqrt{\rho_{4K}^{-}}}\sqrt{\left(\frac{1-\delta_{4K}}{1+\delta_{4K}}\rho_{2K}^{+}-\rho_{2K}^{-}\right)\|\Delta\|_2^2+\frac{\lambda^2K}{\rho_{4K}^{-}}}+\frac{\lambda\sqrt{K}}{\rho_{4K}^{-}\sqrt{1-\delta_{4K}}}\\ &\leq \sqrt{\frac{1}{\rho_{4K}^{-}}\left(\frac{1-\delta_{4K}}{1+\delta_{4K}}\rho_{2K}^{+}-\rho_{2K}^{-}\right)}\|\Delta\|_2+\frac{2\lambda\sqrt{K}}{\rho_{4K}^{-}\sqrt{1-\delta_{4K}}}.\end{aligned} \tag{3.37}$$

Rút gọn:

$$\begin{aligned}\left\|\Delta_{R\backslash\Gamma}\right\|_2 \leq &\sqrt{\frac{1+\delta}{(1-\delta)^2}\frac{(1+\delta)\rho_{2K}^{+}-(1-\delta)\rho_{2K}^{-}}{2\rho_{4K}^{-}}}\|\Delta\|_2\\ &+\lambda\sqrt{K}\frac{1+\delta}{1-\delta}\left(\frac{1}{\sqrt{2}\rho_{4K}^{-}}+\frac{1}{\sqrt{2\rho_{4K}^{-}}}\right),\end{aligned} \tag{3.38}$$

kết quả cuối cùng chính là biểu thức (3.20).

Sử dụng kết quả từ hai biểu thức (3.15) và (3.20), biểu thức (3.11) có thể là được chứng minh như sau,

$$\left\|x^{t+1}-\hat{x}\right\|_2 \leq \|b-\hat{x}\|_2+\left\|b-x^{t+1}\right\|_2 \leq 2\|b-\hat{x}\|_2. \tag{3.39}$$

Sử dụng biểu thức (3.15) từ lưu đồ của thuật toán, giá $\lambda \triangleq \max_i |\langle \nabla \ell(\hat{x}), d_i\rangle|$ của vector x^t phụ thuộc vào tập hợp $\hat{\Gamma}$. Do đó $x^t_{T\backslash\hat{\Gamma}}=0$ và nhận được:

$$\left\|\hat{x}_{T\backslash\hat{\Gamma}}\right\|_2=\left\|(\hat{x}-x^t)_{T\backslash\hat{\Gamma}}\right\|_2 \leq \left\|(\hat{x}-x^t)_{T\backslash\Gamma}\right\|_2=\left\|(\hat{x}-x^t)_{R\backslash\Gamma}\right\|_2, \tag{3.40}$$

ở đây, R là ký hiệu của giá $x^t-\hat{x}$. Từ bất đẳng thức đầu tiên trong biểu thức (3.40) nhận được $\Gamma \subset \hat{\Gamma}$, do đó $T\backslash\hat{\Gamma} \subset T\backslash\Gamma$. Từ bất đẳng thức thứ hai trong (3.40) nhận được $T \subset R$. Kết hợp với biểu thức (3.20) suy ra biểu thức (3.11) thể hiện lỗi ước lượng ở mỗi bước lặp của thuật toán giảm dần thỏa mãn. Có thể nhận xét rằng với điều kiện tập D là hữu hạn và thỏa mãn tính chất giới

hạn đẳng trị RIP, thuật toán DRMP sẽ giảm dần lỗi trong mỗi một bước lặp của quá trình khôi phục tín hiệu được lấy mẫu nén.

3.4. Tổng kết chương

Trong chương này đã trình bày một thuật toán cải tiến dựa trên thuật toán đuổi khớp MP để khôi phục các tín hiệu được lấy mẫu nén. Sự khác biệt giữa hai thuật toán là ở bước tìm kiếm phần dư và bước cập nhật giá trị. So với thuật toán đuổi khớp MP thuật toán cải tiến DRMP có các đặc điểm sau:

- Thuật toán cải tiến DRMP đơn giản hơn, trong bước tìm kiếm phần dư thay vì phải tìm kiếm trên toàn bộ không gian ma trận lấy mẫu nén thuật toán DRMP chỉ tìm kiếm trong một không gian con theo hướng có biến đổi $gradien$ lớn nhất.
- Thuật toán cải tiến DRMP được chứng minh có giá trị lỗi giảm sau mỗi bước lặp của quá trình khôi phục, tính chất này giúp cho thuật toán hội tụ nhanh hơn dẫn đến nó có thể thực hiện nhanh hơn so với thuật toán gốc MP.
- Thuật toán cải tiến DRMP có hạn chế là nó chỉ hiệu quả khi ma trận lấy mẫu nén thỏa điều kiện giới hạn đẳng trị RIP. Nếu điều kiện này không thỏa mãn thuật toán sẽ tương đương với thuật toán gốc.

Trong chương 4 luận án sẽ thực hiện đánh giá so sánh hiệu năng của thuật toán DRMP với thuật toán gốc MP và thuật toán OMP với tín hiệu được lấy mẫu nén sử dụng ma trận BPNSM, ma trận Gauss và ma trận Bernoulli.

CHƯƠNG 4. ĐỀ XUẤT MÔ HÌNH LẤY MẪU NÉN

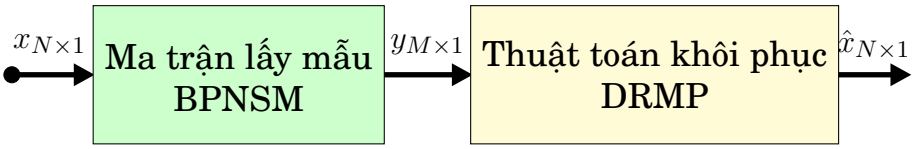
Nội dung chương trình bày quá trình mô phỏng, đánh giá hiệu năng của mô hình lấy mẫu nén sử dụng ma trận lấy mẫu nén BPNSM và thuật toán khôi phục DRMP. Tín hiệu đầu vào đối với mô hình gồm tín hiệu 1 chiều và tín hiệu 2 chiều. Kết quả ứng dụng mô hình lấy mẫu nén đề xuất được trình bày trong [J2], [J3], [J4].

4.1. Mở đầu

Trong những năm gần đây khoa học kỹ thuật phát triển mạnh mẽ, đặc biệt trong lĩnh vực viễn thông và công nghệ thông tin, lượng thông tin được trao đổi ngày càng nhiều dẫn đến hạ tầng viễn thông và công nghệ thông tin luôn phải đổi mới và nâng cấp để có thể đáp ứng nhu cầu về trao đổi thông tin của người dùng [46], [52], [62], [89], [102], [108]. Các tín hiệu phổ biến được trao đổi giữa các hệ thống này phổ biến là tín hiệu 1 chiều và tín hiệu 2 chiều. Tín hiệu 1 chiều là tín hiệu có dạng một hàm với biến số là thời gian $x = f(t)$, các tín hiệu trong hệ thống vô tuyến, hệ thống truyền dẫn quang đều là các tín hiệu một chiều. Tín hiệu 2 chiều là tín hiệu được biểu diễn bởi một hàm của 2 biến số độc lập, các tín hiệu hình ảnh là các tín hiệu 2 chiều. Các tín hiệu có tính chất thưa hoặc thưa trên một hệ cơ sở trực giao là phổ biến trong tự nhiên [34], [50], [88] do đó mô hình lấy mẫu nén rất có tiềm năng khi được áp dụng đối với các dạng tín hiệu này.

Để đánh giá hiệu quả của ma trận lấy mẫu nén các nghiên cứu, thiết kế ma trận thường so sánh với 2 loại ma trận điển hình là ma trận với các phần tử được tạo thành theo phân bố Gauss và Bernoulli [5], [37], [56], [83]. Trong luận án cũng thực hiện mô phỏng, so sánh hiệu quả của ma trận lấy mẫu BPNSM với 2 loại ma trận trên. Đối với thuật toán DRMP được cải tiến từ

thuật toán gốc MP, luận án cũng thực hiện so sánh đánh giá hiệu quả của thuật toán DRMP với thuật toán gốc MP [63] và một thuật toán cải tiến khác rất phổ biến là thuật toán OMP [96]. Các tiêu chí được sử dụng để đánh giá hiệu quả của ma trận và thuật toán đối với tín hiệu 1 chiều gồm: thời gian thực hiện, hệ số tương quan (1.8), sai số trung bình tuyệt đối MAE (1.9). Đối với tín hiệu 2 chiều sử dụng các tham số: thời gian thực hiện, sai số toàn phương trung bình MSE (1.10), tham số PSNR (1.11).



Hình 4.1: *Mô hình lấy mẫu nén đề xuất*

Trong luận án đề xuất mô hình lấy mẫu nén sử dụng ma trận lấy mẫu BPNSM và thuật toán cải tiến DRMP, được biểu diễn trong hình 4.1. Thuật toán DRMP được đề xuất trong luận án có độ phức tạp tính toán thấp hơn và sai số giảm sau mỗi một bước lặp so với thuật toán gốc MP với điều kiện ma trận lấy mẫu đầu vào phải thỏa mãn tính chất RIP (được thể hiện thông qua tính chất không kết hợp). Trong luận án ma trận BPNSM đã được chứng minh có tính chất không kết hợp nhỏ do đó mô hình đề xuất có tính khả thi.

4.2. Mô phỏng đánh giá mô hình với tín hiệu 1 chiều

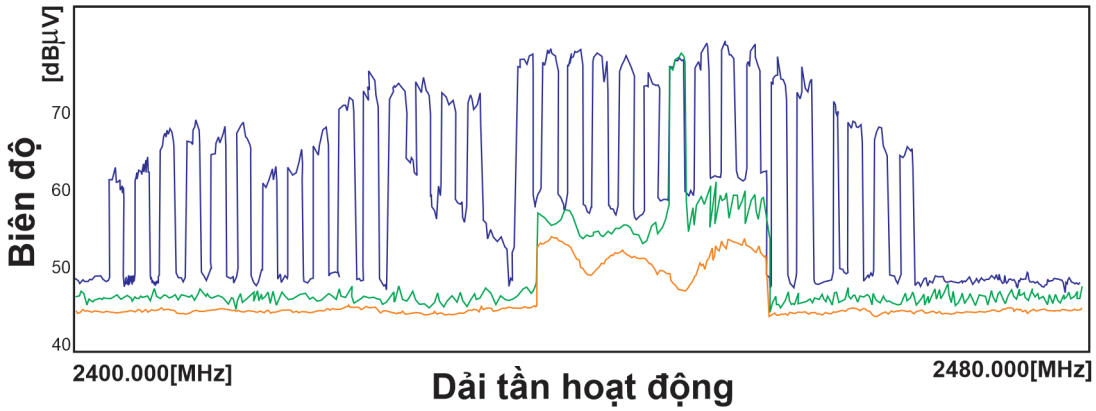
Tín hiệu vô tuyến là một trong các tín hiệu 1 chiều rất phổ biến trong thực tế. Tín hiệu vô tuyến trong giao tiếp giữa các thiết bị truyền nhận thường có băng thông lớn và được phân chia thành nhiều kênh tần số [28]. Việc thu phổ tín hiệu vô tuyến băng rộng cần tần số lấy mẫu cao dẫn đến yêu cầu tốc độ xử lý nhanh và phần cứng phức tạp. Trong các hệ thống điển hình, thu phổ tín hiệu được thực hiện bằng việc lấy mẫu theo tiêu chuẩn Nyquist sử dụng bộ ADC [93]. Ví dụ, để có thể thu phổ tín hiệu trong khoảng từ 200 MHz thì bộ ADC phải lấy mẫu với tần số lấy mẫu 400 MHz hoặc cao hơn dẫn đến tốc độ luồng dữ liệu thu được lên tới 3.2 Gbps (với bộ ADC 8 bit). Việc thực hiện

thu tín hiệu vô tuyến dựa trên tiêu chuẩn Nyquist như vậy là một khó khăn đối với việc xử lý dữ liệu trong thời gian thực bởi tốc độ dữ liệu quá cao và độ phức tạp tính toán lớn. Các kỹ thuật truyền thống như nhận biết phổ năng lượng, khá đơn giản và không yêu cầu biết trước thông tin về tín hiệu. Tuy nhiên, nó nhạy với nhiễu và chỉ phù hợp với tín hiệu băng hẹp nên khó áp dụng trong trường hợp này.

Trong quá trình giao tiếp tại một thời điểm, đối với các bộ thu phát vô tuyến sử dụng một số loại điều chế phổ biến như OFDM, FHSS, DSSS, chỉ trao đổi dữ liệu trên một số kênh tần số cố định [28], sự xuất hiện của tín hiệu là thưa trên toàn bộ dải tần số dành cho việc liên lạc này. Do vậy, tín hiệu truyền thông giữa các bộ thu phát vô tuyến là các tín hiệu thưa trong miền tần số. Dựa vào đặc tính này, có thể thu phổ tín hiệu vô tuyến băng rộng trong xử lý tín hiệu vô tuyến sử dụng kỹ thuật lấy mẫu nén nhằm khắc phục khó khăn trong việc xử lý dữ liệu nêu trên. Kỹ thuật này lấy mẫu và khôi phục tín hiệu thưa giảm đáng kể tốc độ dữ liệu và độ phức tạp tính toán.

Để mô phỏng và đánh giá hiệu năng của mô hình lấy mẫu nén đề xuất, luận án sử dụng bộ tín hiệu vô tuyến được lấy mẫu từ các loại Flycam khác nhau [4]. Trong đó, tín hiệu đưa vào mô hình đề xuất là tín hiệu thu được từ Flycam DJI Mavic 2 được cấu hình điều khiển ở dải tần 2.4 GHz. Tín hiệu điều khiển sử dụng kỹ thuật điều chế FHSS/DSSS với 32 kênh tần số có độ rộng mỗi kênh là 1.4 MHz nằm trong dải 2400 MHz-2470 MHz. Các tín hiệu vô tuyến trong [4] được thực hiện ở khoảng cách gần trong không gian tự do có tỉ số $SNR \geq 20dB$.

Ví dụ bộ điều khiển giao tiếp với Flycam Mavic 2 Pro ở dải tần số từ 2.4 GHz – 2.47 GHz sử dụng kỹ thuật trải phổ nhảy tần với độ rộng 1.4 MHz một kênh tần số như trong hình 4.2. Ở thời điểm t, chỉ một vài kênh M đang được sử dụng do đó $M \ll N$. Coi S như là tổ hợp các kênh đang được sử dụng. Khi đó tất cả các kênh còn lại không thuộc tập hợp S, các kênh đó tương ứng với tín hiệu bằng 0. Và coi tín hiệu lấy mẫu trên toàn dải tần số đó là một tín



Hình 4.2: *Phổ tần số sử dụng của Flycam Mavic pro*

hiệu có độ thưa S trong miền tần số. Do đó tín hiệu nhận được biểu diễn bởi:

$$x(t) = \sum_{n \in S} s_n(t) * h_n(t) + w(t), \tag{4.1}$$

trong miền tần số tín hiệu này có thể được biểu diễn như sau:

$$X(f) = \sum_{n \in S} D_h * s_n(f) + w(f), \tag{4.2}$$

ở đây, $s_n(t)$ là tín hiệu nằm trên kênh tần số thứ n của Flycam, $h_n(t)$ là đáp ứng của kênh và $w(t)$ là nhiễu cộng, D_h là ma trận độ lợi kênh [8]. Khi đó $X(f)$ là thưa trong miền tần số và có thể áp dụng kỹ thuật lấy mẫu nén để lấy mẫu tín hiệu này.

Tín hiệu trong miền tần số được nhân với ma trận lấy mẫu để tạo thành vector lấy mẫu nén. Ma trận lấy mẫu Φ là ma trận BPNSM có kích thước $M \times N$ được sử dụng để lấy mẫu với một vài phép đo từ tín hiệu thưa này, vector lấy mẫu nén khi đó được biểu diễn bởi

$$y(f) = \Phi X(f), \tag{4.3}$$

trong đó $y(f)$ là ánh xạ của tín hiệu nhận được trong miền tần số và $y(f) = [f_1, f_2, ...f_N]$ là vector $N \times 1$, $y(f)$ là vector $M \times 1$, Φ là ma trận lấy mẫu BPNSM $M \times N$. Tín hiệu được khôi phục lại từ tổ hợp các phép đo $y(f)$ bằng cách sử

dụng thuật cải tiến DRMP.

4.2.1. Ma trận lấy mẫu tín hiệu 1 chiều

Ma trận lấy mẫu BPNSM có kích thước $(2^n - 1) \times 2^{n+1}$ được tạo từ các chuỗi PN lồng ghép phi tuyến có đa thức sinh trên trường $g(2^n)$ với $n = 10$.

Xây dựng ma trận:

Chọn đa thức sinh thứ nhất: $g(d) = 1 + d^3 + d^{10}$ trên trường $GF(2^{10})$. $T = L/N = (2^n - 1)/(2^m - 1) = 1023/31 = 33$.

Tìm thứ tự lồng ghép I_p^T bằng việc tính toán hàm Vết ánh xạ $GF(2^{10})$ xuống $GF(2^5) = 1 + d^2 + d^5$.

Với α là nghiệm của $g(d) : g(d) = 0 \rightarrow 1 + \alpha^3 = \alpha^{10}$.

Tính toán hàm Vết, có: $Tr_m^n = \sum_{k=0}^{\frac{n}{m}-1} x^{2^{mk}} = x + x^{32}$.

Quá trình tính toán được thực hiện bằng phần mềm sử dụng ngôn ngữ Python trong phụ lục 1 và cho ra kết quả bảng 4.1 là bảng các giá trị nghiệm α và bảng 4.2 là giá trị của hàm Vết. Tìm kiếm các thứ tự sắp xếp từ 2 bảng 4.1 và 4.2 thu được thứ tự lồng ghép I_p^T.

Với j chạy từ 0 đến $T-1 = 32$, tính được thứ tự pha: $I_p^T = \{\infty, 23, 15, 20, 30, 10, 9, 17, 29, 13, 20, 21, 18, 21, 3, 9, 27, 12, 26, 21, 9, 7, 11, 11, 5, 22, 11, 4, 6, 27, 18, 14, 23\}$.

Để có chuỗi phi tuyến, giữ nguyên thứ tự I_p^T và thay thế các chuỗi con bằng các chuỗi con khác tương ứng.

Một số tính chất của chuỗi PN phi tuyến được tạo ra từ đa thức sinh $g(d) = 1 + d^3 + d^{10}$ trên trường $GF(2^{10})$.

- Phổ của chuỗi PN lồng ghép phi tuyến thể hiện sự phân bố đồng đều của các tần số trong chuỗi như trong hình 4.3.
- Hàm tự tương quan của chuỗi PN lồng ghép phi tuyến đa thức sinh có bậc 10 là một hàm nhọn như hình 4.4
- Hàm tương quan chéo của 2 chuỗi PN lồng ghép phi tuyến đa thức sinh

Bảng 4.1: *Bảng* α^{Ti}

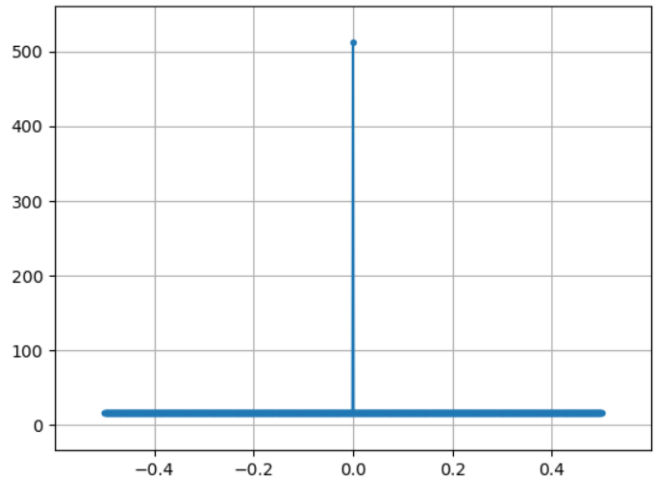
i	α^{Ti}	i	α^{Ti}
0	1	1	$\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^9$
2	$1+\alpha+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^8$	3	$\alpha^2+\alpha^3+\alpha^8+\alpha^9$
4	$\alpha^3+\alpha^4+\alpha^5+\alpha^9$	5	$\alpha+\alpha^3+\alpha^4+\alpha^6$
6	$\alpha+\alpha^8+\alpha^9$	7	$1+\alpha+\alpha^2+\alpha^3$
8	$1+\alpha+\alpha^3+\alpha^4+\alpha^6$	9	$\alpha+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^8$
10	$\alpha^5+\alpha^6+\alpha^8$	11	$1+\alpha+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^9$
12	$\alpha+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^9$	13	$\alpha+\alpha^3+\alpha^4+\alpha^5+\alpha^8$
14	$1+\alpha^2+\alpha^4+\alpha^6$	15	$1+\alpha^3+\alpha^4+\alpha^6+\alpha^8+\alpha^9$
16	$1+\alpha^5+\alpha^6+\alpha^8$	17	$1+\alpha+\alpha^3+\alpha^4+\alpha^5+\alpha^8$
18	$1+\alpha^3+\alpha^4+\alpha^5+\alpha^9$	19	$1+\alpha^2+\alpha^4+\alpha^5+\alpha^9$
20	$1+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^9$	21	$1+\alpha+\alpha^8+\alpha^9$
22	$1+\alpha+\alpha^5+\alpha^6+\alpha^9$	23	$\alpha^2+\alpha^4+\alpha^5+\alpha^8$
24	$\alpha+\alpha^5+\alpha^6+\alpha^9$	25	$\alpha^3+\alpha^4+\alpha^6+\alpha^8+\alpha^9$
26	$1+\alpha^2+\alpha^3+\alpha^8+\alpha^9$	27	$\alpha^2+\alpha^4+\alpha^6$
28	$1+\alpha^2+\alpha^4+\alpha^5+\alpha^8$	29	$\alpha+\alpha^2+\alpha^3$
30	$1+\alpha+\alpha^2+\alpha^4+\alpha^5+\alpha^9$	31	

có bậc 10 được biểu diễn trong hình 4.5 thể hiện rằng với 2 chuỗi phi tuyến giả ngẫu nhiên hầu như không có tương quan tuyến tính với nhau.

Sau khi tạo được chuỗi PN phi tuyến bậc 10 từ qui trình được trình bày trong mục (2.6) có ma trận Φ_1. Tương tự với các bước trên nhưng với đa thức $g(d) = 1+d+d^3+d^4+d^{10}$ trên trường $GF(2^{10})$ và ánh xạ xuống $GF(2^5) = 1+d^3+d^5$ xây dựng được ma trận Φ_2 và tổ hợp 2 ma trận Φ_1 và Φ_2 lại với nhau thu được ma trận lấy mẫu nén Φ. Ma trận này có kích thước $[1023 \times 2048]$ và được sử dụng để lấy mẫu tín hiệu vô tuyến của Flycam DJI Mavic trong [4]. Trong phần tiếp theo luận án trình bày các kết quả của quá trình thử nghiệm khôi phục lại tín hiệu vô tuyến lấy mẫu nén từ Flycam.

Bảng 4.2: *Bảng* $Tr(\alpha)$

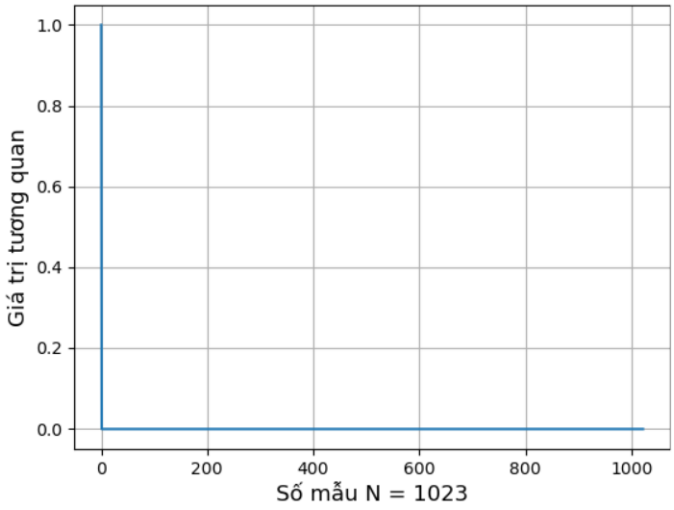
j	$Tr(\alpha^j)$	j	$Tr(\alpha^j)$
0	0	1	$\alpha^2+\alpha^4+\alpha^5+\alpha^8=\alpha^{23.33}$
2	$1+\alpha^3+\alpha^4+\alpha^6+\alpha^8+\alpha^9=\alpha^{15.33}$	3	$1+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^9=\alpha^{20.33}$
4	$1+\alpha+\alpha^2+\alpha^4+\alpha^5+\alpha^9=\alpha^{19.33}$	5	$\alpha^5+\alpha^6+\alpha^8=\alpha^{10.33}$
6	$\alpha+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^8=\alpha^{9.33}$	7	$1+\alpha+\alpha^3+\alpha^4+\alpha^5+\alpha^8=\alpha^{17.33}$
8	$\alpha+\alpha^2+\alpha^3=\alpha^{29.33}$	9	$\alpha+\alpha^3+\alpha^4+\alpha^5+\alpha^8=\alpha^{13.33}$
10	$1+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^9=\alpha^{20.33}$	11	$1+\alpha+\alpha^8+\alpha^9=\alpha^{21.33}$
12	$1+\alpha^3+\alpha^4+\alpha^5+\alpha^9=\alpha^{18.33}$	13	$1+\alpha+\alpha^8+\alpha^9=\alpha^{21.33}$
14	$\alpha^2+\alpha^3+\alpha^8+\alpha^9=\alpha^{3.33}$	15	$\alpha+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^8=\alpha^{9.33}$
16	$\alpha^2+\alpha^4+\alpha^6=\alpha^{27.33}$	17	$\alpha+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^9=\alpha^{12.33}$
18	$1+\alpha^2+\alpha^3+\alpha^8+\alpha^9=\alpha^{26.33}$	19	$1+\alpha+\alpha^8+\alpha^9=\alpha^{21.33}$
20	$\alpha+\alpha^2+\alpha^3+\alpha^5+\alpha^6+\alpha^8=\alpha^{9.33}$	21	$1+\alpha+\alpha^2+\alpha^3=\alpha^{7.33}$
22	$1+\alpha+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^9=\alpha^{11.33}$	23	$1+\alpha+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^9=\alpha^{11.33}$
24	$\alpha+\alpha^3+\alpha^4+\alpha^6=\alpha^{5.33}$	25	$1+\alpha+\alpha^5+\alpha^6+\alpha^9=\alpha^{22.33}$
26	$1+\alpha+\alpha^2+\alpha^4+\alpha^6+\alpha^8+\alpha^9=\alpha^{11.33}$	27	$\alpha^3+\alpha^4+\alpha^5+\alpha^9=\alpha^{4.33}$
28	$\alpha+\alpha^8+\alpha^9=\alpha^{6.33}$	29	$\alpha^2+\alpha^4+\alpha^6=\alpha^{27.33}$
30	$1+\alpha^3+\alpha^4+\alpha^5+\alpha^9=\alpha^{18.33}$	31	$1+\alpha^2+\alpha^4+\alpha^6=\alpha^{14.33}$
32	$\alpha^2+\alpha^4+\alpha^5+\alpha^8=\alpha^{23.33}$	33	



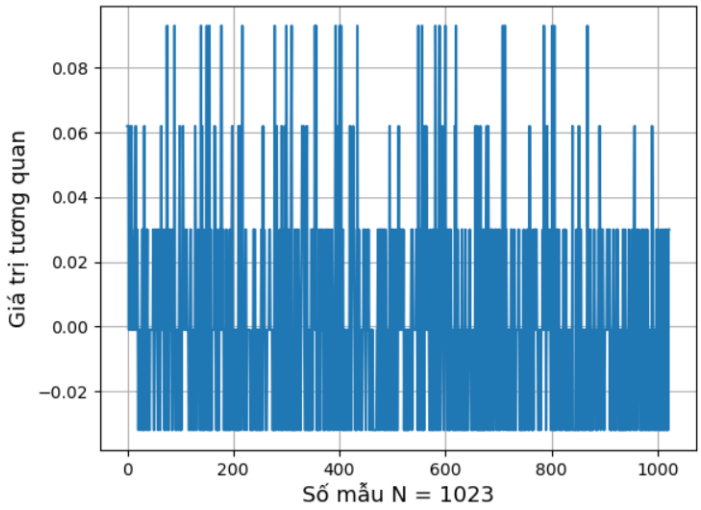
Hình 4.3: *Phổ của chuỗi PN lồng ghép phi tuyến*

4.2.2. Khôi phục tín hiệu 1 chiều

Để đánh giá hiệu quả của mô hình lấy mẫu nén đề xuất luận án tiến hành thực nghiệm với các kịch bản sau.



Hình 4.4: *Hàm tự tương quan của chuỗi PN lồng ghép phi tuyến*

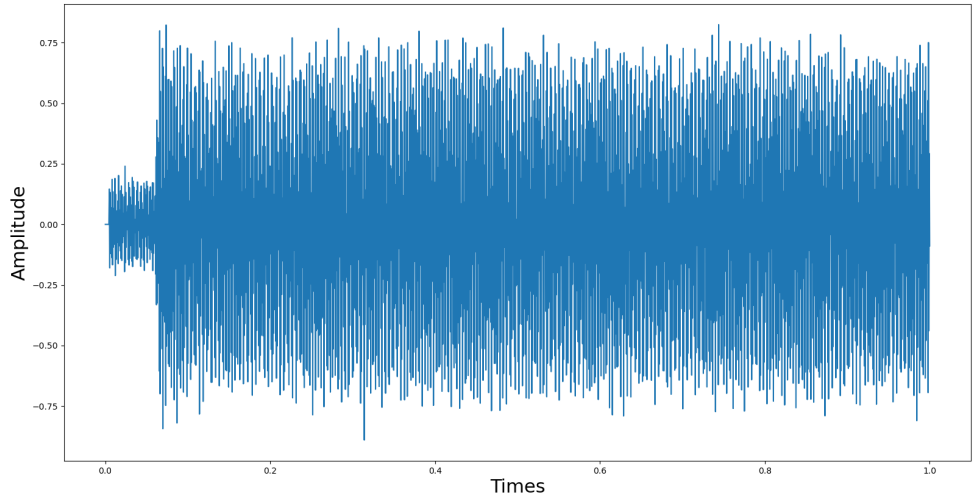


Hình 4.5: *Hàm tương quan chéo của 2 chuỗi PN lồng ghép phi tuyến*

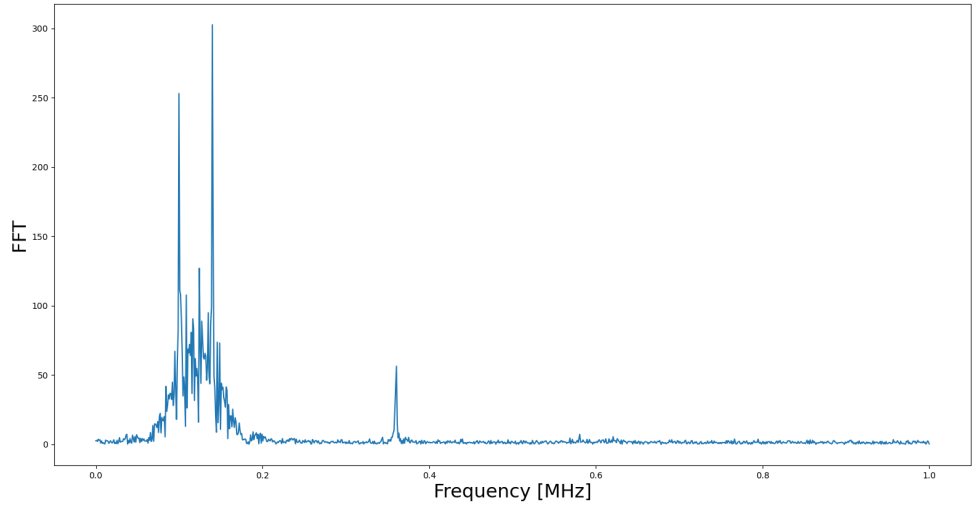
- Sử dụng các thuật toán DRMP, OMP và MP kết hợp với 3 ma trận lấy mẫu nén BPNSM, Gauss, Bernoulli đối với đầu vào là tín hiệu gốc được lựa chọn trong [4] có tỉ số $SNR \geq 20dB$ và tín hiệu gốc cộng thêm nhiễu có tỉ số $SNR \geq 15dB$ để mô phỏng hiệu quả của mô hình trong trường hợp tín hiệu có SNR thấp.

Ma trận lấy mẫu BPNSM có kích thước $[1023 \times 2048]$. Tín hiệu vô tuyến được chia thành các đoạn có độ dài tương ứng là 2048, hệ số nén trong mô hình này

là ≈ 0.5. Hình 4.6 thể hiện tín hiệu vô tuyến thu được từ Flycam trong miền thời gian và phổ tần số của nó. Trong hình 4.6a có thể nhận thấy tín hiệu rất dày đặc trong miền thời gian, nhưng trong hình 4.6b có thể nhận thấy tín hiệu là thưa trong miền tần số.



(a)

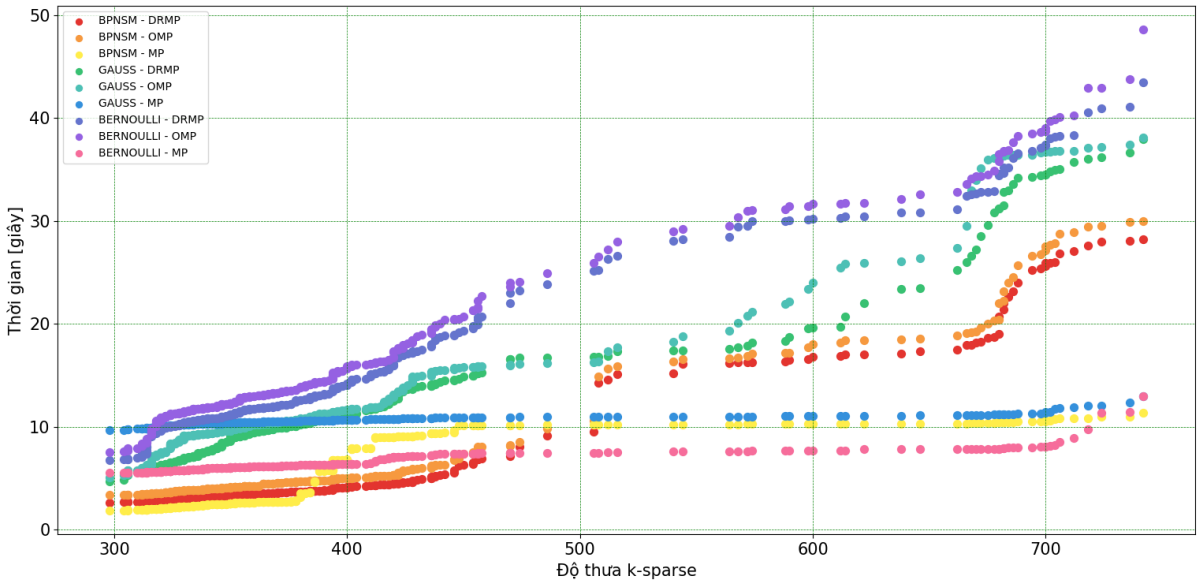


(b)

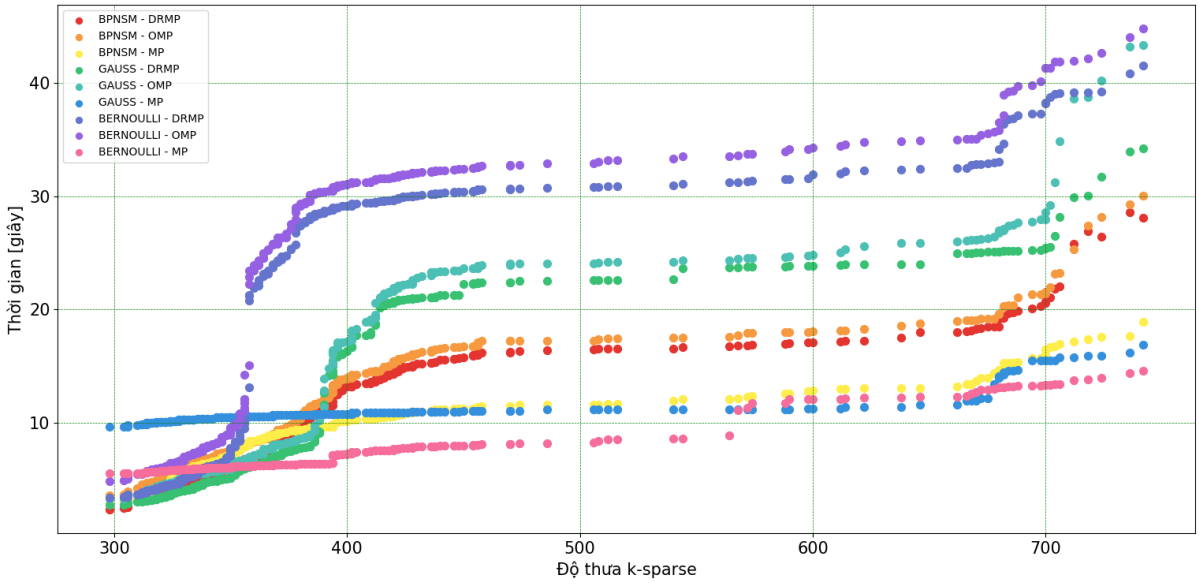
Hình 4.6: *Dạng tín hiệu của Flycam: 4.6a là tín hiệu gốc trong miền thời gian, 4.6b là biến đổi FFT của nó*

Trong thực nghiệm này 289 mẫu tín hiệu có kích thước 1×2048 được chọn lọc từ các mẫu tín hiệu vô tuyến trong [4] và được sắp xếp lại với giá trị $K - sparse$ tăng dần. Giá trị $K - sparse$ của tín hiệu vô tuyến này không thay đổi tuyến tính mà phân bố trong dải từ $300 - 740$. Dưới đây là một số kết quả

khi thực hiện mô hình lấy mẫu nén trên máy tính ASUS Core I5 Ram 8G với phiên bản Python 3.9.



Hình 4.7: *Thời gian thực hiện trong trường hợp không cộng nhiễu*

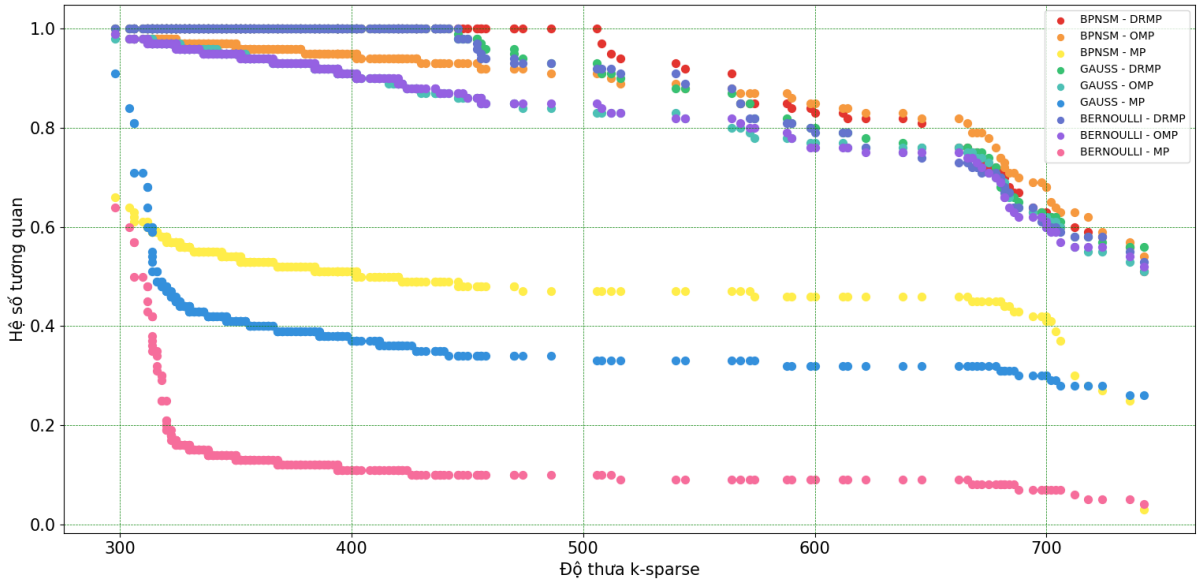


Hình 4.8: *Thời gian thực hiện trong trường hợp cộng nhiễu*

Hình 4.7 là đồ thị dạng điểm biểu diễn thời gian thực hiện của quá trình lấy mẫu nén và khôi phục lại tín hiệu vô tuyến với 3 thuật toán DRMP, OMP, MP và 3 ma trận tương ứng là BPNSM, Gauss và Bernoulli. Trục hoành của

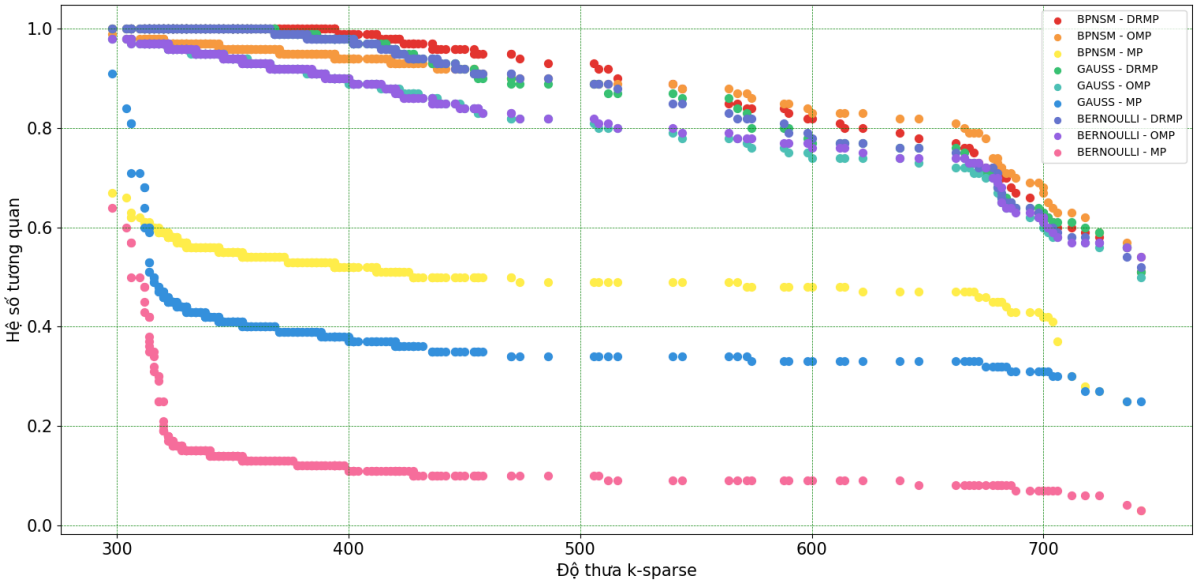
đồ thị biểu diễn các giá trị về độ thưa của 289 mẫu và được sắp xếp theo giá trị độ thưa tăng dần. Trong đó có thể nhận thấy thuật toán DRMP và thuật toán OMP gần như tương đương khi kết hợp với cùng một loại ma trận. Thời gian thực hiện khi kết hợp ma trận BPNSM với 2 thuật toán DRMP và OMP nhỏ hơn khi kết hợp với 2 ma trận Gauss và Bernoulli. Trường hợp thuật toán MP khi kết hợp với cả 3 ma trận đều có thời gian thực hiện nhanh nhưng không thể hiện rằng thuật toán này có hiệu quả cao. Thuật toán MP có thời gian khôi phục nhanh nhưng các chỉ tiêu về hệ số tương quan và sai số trung bình tuyệt đối MAE đều kém.

Hình 4.8 biểu diễn thời gian thực hiện trong trường hợp có cộng nhiễu vào tín hiệu gốc. Các nhận định cũng tương tự trong trường hợp không cộng nhiễu nhưng khi độ thưa $K - sparse > 350$ thời gian tăng nhanh đột biến.



Hình 4.9: *Hệ số tương quan trong trường hợp không cộng nhiễu*

Hình 4.9 biểu diễn hệ số tương quan của tín hiệu sau khôi phục với tín hiệu gốc được lấy mẫu nén. Trong trường hợp tín hiệu gốc không cộng nhiễu thuật toán DRMP kết hợp với ma trận BPNSM có hệ số tương quan tốt nhất trong tất cả các trường hợp. Khi độ thưa $K - sparse$ của tín hiệu vượt quá 500 hệ số tương quan trong trường hợp kết hợp thuật toán DRMP và ma trận

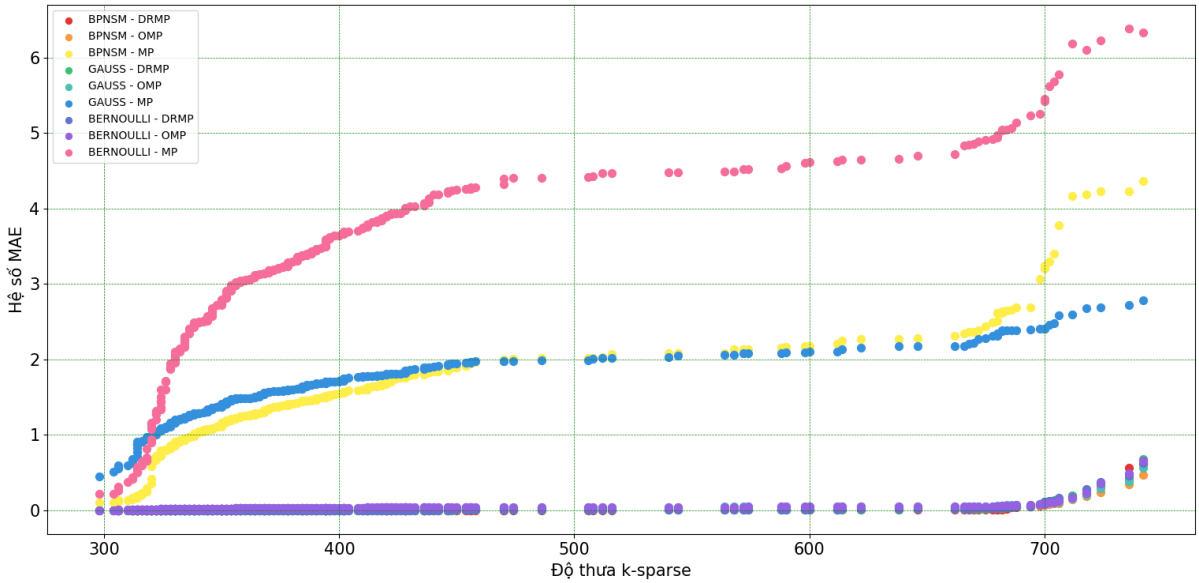


Hình 4.10: *Hệ số tương quan trong trường hợp cộng nhiễu*

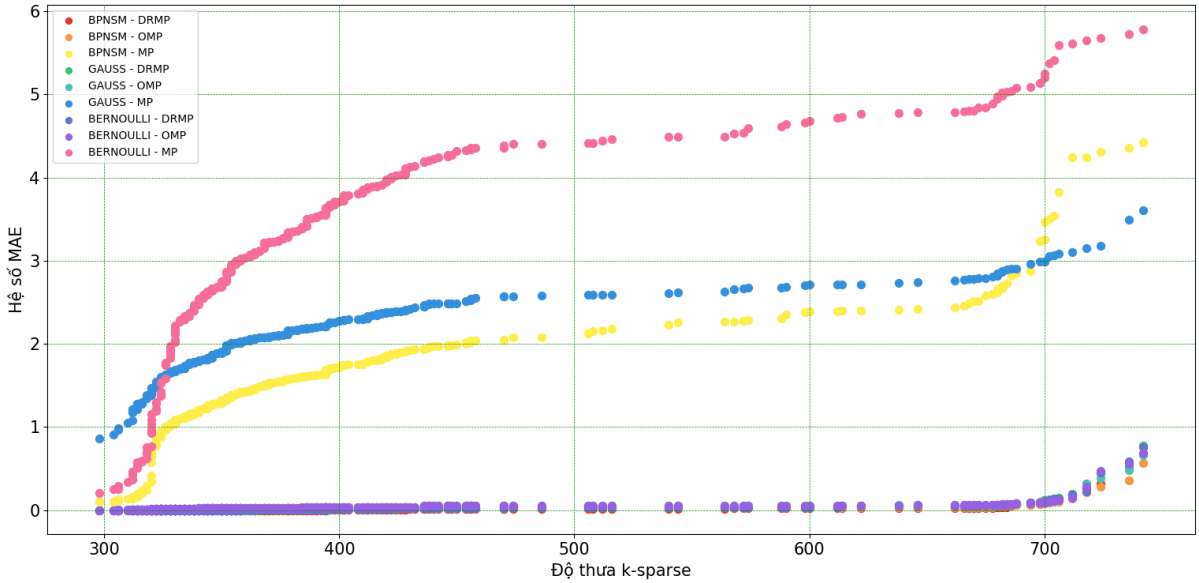
BPNSM bắt đầu giảm. Thuật toán DRMP kết hợp với 2 ma trận Gauss và Bernoulli cũng cho hệ số tương quan tốt nhưng giá trị tương quan bắt đầu giảm với độ thưa $K - sparse > 450$. Tín hiệu khôi phục bằng thuật toán MP có hệ số tương quan thấp đối với cả 3 ma trận, trong đó việc kết hợp thuật toán MP với ma trận Bernoulii cho kết quả kém nhất.

Hình 4.10 biểu diễn hệ số tương quan của tín hiệu sau khôi phục với tín hiệu gốc trong trường hợp cộng nhiễu vào tín hiệu gốc. Các nhận định cũng tương tự trong trường hợp không cộng nhiễu nhưng hệ số tương quan bắt đầu giảm với độ thưa $K - sparse < 400$.

Hình 4.11 biểu diễn sai số trung bình tuyệt đối MAE của tín hiệu được khôi phục sử dụng thuật toán của 3 thuật toán và 3 ma trận so với tín hiệu gốc. Sai số MAE khi sử dụng thuật toán MP là cao nhất trong các thuật toán và đặc biệt cao trong trường hợp thuật toán MP kết hợp với ma trận Bernoilli. Sai số trung bình tuyệt đối tăng khi độ thưa của tín hiệu được lấy mẫu nén tăng. Hình 4.12 biểu diễn sai số trung bình tuyệt đối MAE trong trường hợp cộng nhiễu vào tín hiệu gốc cũng cho các nhận xét tương tự trong trường hợp không cộng nhiễu nhưng hệ số MAE lớn hơn trong tất cả các trường hợp. Hai



Hình 4.11: *Hệ số MAE trong trường hợp không cộng nhiễu*



Hình 4.12: *Hệ số MAE trong trường hợp cộng nhiễu*

thuật toán DRMP và OMP cho sai số MAE gần như tương đương nhau trong cả 2 trường hợp không cộng nhiễu và có cộng nhiễu vào tín hiệu gốc.

Từ các thực nghiệm trên, có thể thấy rằng, đối với cả hai trường hợp không cộng nhiễu và có cộng nhiễu vào tín hiệu gốc ma trận BPNSM được đề xuất cung cấp độ chính xác tái tạo tốt hơn so với ma trận Gauss và Bernoulli.

Thuật toán DRMP được đề xuất cho độ chính xác tốt hơn nhiều so với thuật

toán gốc MP và là tốt nhất trong trường hợp kết hợp giữa thuật toán DRMP và ma trận lấy mẫu BPNSM.

4.3. Mô phỏng đánh giá mô hình với tín hiệu 2 chiều

Trong những năm gần đây lượng thông tin hình ảnh phát sinh, trao đổi trên mạng ngày càng nhiều và phổ biến. Có các bức ảnh thể hiện tính chất thưa trong miền thời gian như ảnh thiên văn, ảnh y tế [50], và phần lớn các bức ảnh còn lại thể hiện tính chất thưa thông qua một cơ sở biến đổi như biến đổi DCT [54]. Do đó trong luận án sử dụng một số bức ảnh đa cấp xám phổ biến như hình 4.13 trong lĩnh vực xử lý ảnh để đánh giá hiệu quả của mô hình lấy mẫu nén được đề xuất.

(a) *Cameraman*

(b) *Lena*

(c) *Phantom*

(d) *Bone*

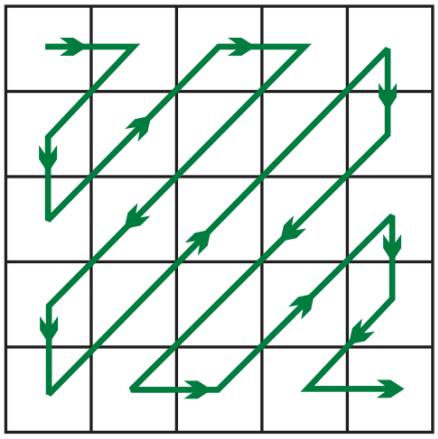
(e) *Saturn*

(f) *Fly*

Hình 4.13: *Ảnh thử nghiệm*

Về mặt trực giác các bức ảnh trong hình 4.13 không phải tín hiệu thưa, để có thể thực hiện với mô hình lấy mẫu nén, các bức ảnh này được biến đổi

DCT [51] và chuyển đổi thành dữ liệu 1 chiều sử dụng ma trận *zigzag* như trong hình 4.14.



Hình 4.14: *Ma trận zigzag*

Ma trận DCT thu được của bức ảnh được biểu diễn theo biểu thức sau:

$$x_{DCT}(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} x_{pixel}(u,v) \cos\left[\frac{(2u+1)i\pi}{2N}\right] \cos\left[\frac{(2v+1)j\pi}{2N}\right], \tag{4.4}$$

trong đó, $C(x) = \begin{cases} \frac{1}{\sqrt{2}} & \text{Nếu } x = 0 \\ 1 & \text{Nếu } x = 1. \end{cases}$

Chuyển đổi x_{DCT} thành tín hiệu một chiều để thực hiện lấy mẫu nén chúng ta thu được

$$x(n) = Z.x_{DCT}, \tag{4.5}$$

trong đó, Z là ma trận *zigzag*.

4.3.1. Ma trận lấy mẫu ảnh số

Trong kịch bản thử nghiệm đối với lấy mẫu nén ảnh số luận án thực hiện với các bức ảnh có kích thước 512×512. Ma trận lấy mẫu BPNSM được tạo thành từ 2 đa thức sinh bậc 9 là $g(d) = 1+d+d^3+d^4+d^9$ và $g(d) = 1+d+d^2+d^7+d^9$ trên trường $GF(2^9)$. Quá trình xây dựng ma trận lấy mẫu tương tự như trong

phần (4.2.1), các ma trận đều được tạo ra bằng một phần mềm tạo ma trận sử dụng ngôn ngữ Python (phụ lục 1) và chỉ thay đổi điều kiện thiết lập đối với đa thức đầu vào. Ma trận BPNSM thu được với kích thước 511×1024 được sử dụng để làm ma trận lấy mẫu. Để so sánh đánh giá hiệu quả của mô hình lấy mẫu nén, các trường hợp thực nghiệm cũng được thực hiện như trong mục (4.2.2).

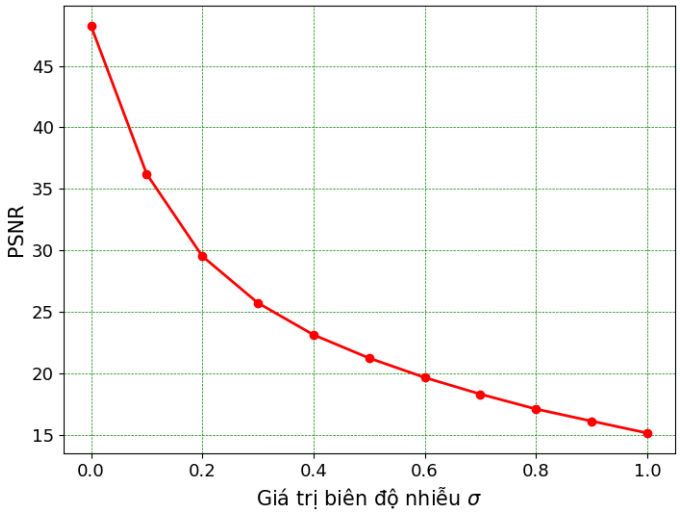
4.3.2. Khôi phục lại ảnh gốc

Luận án sử dụng 6 hình ảnh đa cấp xám 8 bit có kích thước 512×512 như trong hình 4.13 làm tín hiệu đầu vào cho mô hình lấy mẫu nén. Các bức ảnh được chia thành các khối ảnh 32×32 sau đó thực hiện biến đổi DCT và chuyển thành tín hiệu 1-D trước khi thực hiện việc lấy mẫu nén. Quá trình lấy mẫu nén cũng được thực hiện với 2 trường hợp là có cộng nhiễu vào tín hiệu ảnh gốc và không cộng nhiễu đối với tín hiệu ảnh. Các giá trị điển hình cho tham số PSNR đối với ảnh 8 bit khi nén bị tổn hao là từ 30 đến 50dB [36]. Mô phỏng trong luận án thực hiện việc đánh giá ảnh hưởng của nhiễu tới mô hình bằng việc thay đổi giá trị nhiễu σ trong dải $[0 \div 1]$ và thu được kết quả thay đổi của tham số PSNR như trong hình 4.15. Kết quả giá trị PSNR $\approx$ 30dB với $\sigma = 0.2$. Do đó tham số nhiễu $\sigma = 0.2$ và trường hợp không có nhiễu tại $\sigma = 0$ được sử dụng để đánh giá so sánh giữa các ma trận và thuật toán.

Trong thực nghiệm này luận án áp dụng 3 thuật toán khôi phục là MP, OMP [80], và thuật toán DRMP được đề xuất trong luận án để thực hiện quá trình khôi phục lại tín hiệu ảnh được lấy mẫu nén bởi các ma trận được tạo ra trong mục (4.3.1). Để đánh giá hiệu suất của hình ảnh được khôi phục lại, luận án sử dụng các tham số đánh giá thị giác là PSNR, MSE và thời gian thực hiện.

Dưới đây là một số kết quả thực nghiệm khi thực hiện mô phỏng mô hình lấy mẫu nén trên máy tính ASUS Core I5 Ram 8G.

Từ dữ liệu các bảng 4.3, 4.4, 4.5 nhận thấy rằng:



Hình 4.15: *Đồ thị đánh giá 3 ma trận lấy mẫu*

Bảng 4.3: *Bảng PSNR(dB) trong trường hợp không cộng nhiễu*

Ảnh test	MP			OMP			DRMP		
	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM
Cameraman	6.56	2.68	5.72	28.91	29.32	36.56	31.29	30.30	35.90
Lena	8.86	5.88	8.37	36.56	33.98	37.45	34.73	33.01	36.16
Phantom	13.95	6.45	12.75	32.51	33.62	37.08	34.89	36.38	41.36
Bone	9.59	3.61	9.35	29.72	29.28	32.73	30.40	30.03	36.69
Saturn	14.03	11.47	13.09	42.43	42.92	45.21	41.59	42.07	41.09
Fly	6.25	4.30	5.05	29.72	29.28	32.73	39.80	41.63	38.84

Bảng 4.4: *Bảng MSE trong trường hợp không cộng nhiễu*

Ảnh test	MP			OMP			DRMP		
	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM
Cameraman	0.89	1.40	0.99	0.07	0.07	0.04	0.05	0.06	0.03
Lena	0.94	1.32	0.99	0.05	0.05	0.04	0.05	0.06	0.04
Phantom	0.84	2	0.97	0.10	0.09	0.06	0.08	0.06	0.04
Bone	0.94	1.87	0.97	0.09	0.10	0.07	0.09	0.09	0.04
Saturn	0.89	1.19	0.99	0.03	0.03	0.03	0.04	0.04	0.04
Fly	0.87	1.09	0.99	0.02	0.02	0.01	0.02	0.02	0.02

- Trong 3 thuật toán được lựa chọn để thực hiện khôi phục lại tín hiệu là ảnh đa cấp xám được lấy mẫu nén thì thuật toán MP có hiệu quả kém nhất. Thuật toán MP cho sai số lớn và thời gian xử lý lâu nhất so với 2 thuật toán còn lại.

Bảng 4.5: *Bảng thời gian xử lý trong trường hợp không cộng nhiễu*

Ảnh test	MP			OMP			DRMP		
	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM
Cameraman	54.22	86.86	83.83	2.48	4.35	2.34	2.99	6.89	1.59
Lena	53.77	94.56	91.73	2.43	4.686	2.32	2.28	6.14	1.15
Phantom	26.48	21.37	19.20	4.39	6.445	4.49	1.21	2.22	0.71
Bone	24.8	45.25	47.34	3.41	5.586	3.29	2.60	5.35	1.37
Saturn	22.17	36.10	37.82	3.52	6.635	3.47	0.82	1.8	0.42
Fly	53.11	104.98	81.36	2.48	6.42	2.40	1.56	3.94	0.62

- Giữa thuật toán OMP và thuật toán DRMP có thể nhận xét gần như tương đương nhau về thời gian xử lý cũng như tham số PSNR khi sử dụng ma trận lấy mẫu là ma trận Gauss và Bernoulli.
- Khi kết hợp thuật toán DRMP với ma trận lấy mẫu BPNSM có thể nhận thấy cả thông số về thời gian và thông số về độ sai lệch MSE đều giảm rõ rệt. Hiệu quả này cũng đúng khi khôi phục tín hiệu được lấy mẫu nén bằng ma trận BPNSM sử dụng thuật toán OMP.

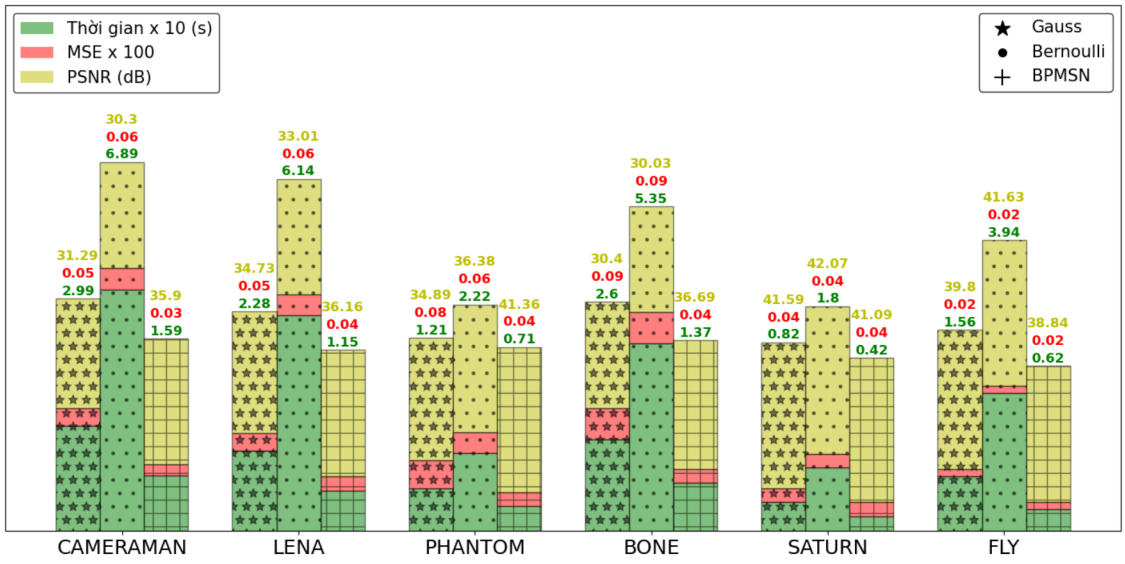
Hình 4.16 thể hiện một số hình ảnh đã được khôi phục trong trường hợp không cộng nhiễu. Bằng trực quan có thể thấy hình ảnh được lấy mẫu và khôi phục với mô hình đề xuất cho chất lượng tốt nhất trong các trường hợp.

(a) ***(b)*** ***(c)***

Hình 4.16: *Ảnh khôi phục bằng thuật toán DRMP. 4.16a sử dụng ma trận Gauss, 4.16b sử dụng ma trận Bernoulli, 4.16c sử dụng ma trận BPNSM*

Đồ thị trong hình 4.17 mô tả các thông số khi thực hiện lấy mẫu nén trong điều kiện không cộng nhiễu với 3 loại ma trận là Gauss, Bernoulli và ma

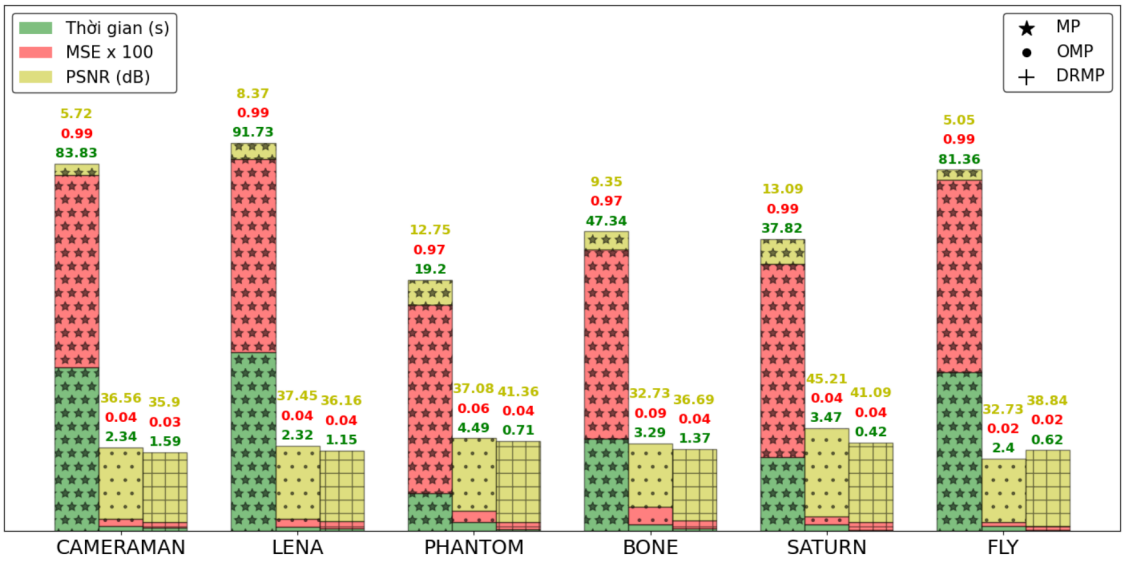
trận được đề xuất trong luận án BPNSM. Có thể nhận thấy với các tham số thị giác PSNR gần như tương đương thì khi thực hiện lấy mẫu nén sử dụng ma trận được thiết kế trong luận án cho thời gian thực hiện nhanh hơn và tỉ lệ lỗi giảm một cách đáng kể. Luận án cũng thực nghiệm với cùng một quá trình như vậy nhưng với tín hiệu được cộng thêm nhiễu để đánh giá khả năng chống nhiễu của thuật toán cũng như của ma trận lấy mẫu.



Hình 4.17: *Đồ thị đánh giá 3 ma trận lấy mẫu*

Đồ thị trong hình 4.18 mô tả các thông số khi thực hiện lấy mẫu nén với cùng một ma trận BPNSM nhưng được thực hiện với 3 thuật toán khác nhau MP, OMP và thuật toán được đề xuất DRMP. Có thể nhận thấy thuật toán MP có thời gian thực hiện lớn nhất, giá trị lỗi cao và tham số PSNR nhỏ khi kết hợp với cả 3 ma trận lấy mẫu là Gauss, Bernoulli và BPNSM. Thuật toán DRMP và thuật toán OMP cho giá trị lỗi và tham số PSNR gần như tương đương nhưng thuật toán DRMP thực hiện với thời gian nhanh hơn đáng kể. Trong luận án cũng thực hiện việc cộng nhiễu vào tín hiệu ảnh để đánh giá khả năng khôi phục của thuật toán đề xuất trong điều kiện có nhiễu.

Để đánh giá khả năng lấy mẫu nén ảnh số dưới tác động của nhiễu. Trong phần tiếp theo luận án tiến hành thực nghiệm với 3 thuật toán khôi phục là



Hình 4.18: *Đồ thị đánh giá 3 thuật toán lấy mẫu*

MP, OMP và DRMP, đối với tín hiệu ảnh có nhiễu được lấy mẫu sử dụng cả 3 ma trận Gauss, Bernoulli và BPNSM.

Bảng 4.6: *Bảng PSNR(dB) trong trường hợp cộng nhiễu*

Ảnh test	MP			OMP			DRMP		
	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM
Cameraman	6.89	5.77	5.77	16.80	24.35	30.68	19.67	25.39	33.37
Lena	9.54	8.4	8.4	16.96	25.21	32.58	20.22	27.22	34.10
Phantom	13.67	12.32	12.3	16.92	25.47	32.30	20.78	28.28	35.88
Bone	10.32	9.16	9.15	16.60	24.25	30.45	19.77	25.28	33.87
Saturn	13.67	12.62	12.62	17.13	25.84	33.60	21.40	29.74	36.22
Fly	6.28	5.17	5.17	17.21	25.89	33.68	21.03	29.08	35.45

Bảng 4.7: *Bảng MSE trong trường hợp cộng nhiễu*

Ảnh test	MP			OMP			DRMP		
	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM
Cameraman	0.86	0.98	0.98	0.27	0.12	0.06	0.2	0.10	0.04
Lena	0.86	0.98	0.99	0.37	0.14	0.06	0.25	0.11	0.05
Phantom	0.87	1.02	1.02	0.60	0.22	0.10	0.38	0.16	0.07
Bone	0.86	0.98	0.99	0.42	0.17	0.09	0.29	0.15	0.06
Saturn	0.93	1.04	1.05	0.62	0.23	0.09	0.38	0.15	0.07
Fly	0.86	0.98	0.98	0.25	0.09	0.04	0.16	0.06	0.03

Từ dữ liệu các bảng 4.6, 4.7, 4.8 nhận thấy rằng:

Bảng 4.8: *Bảng thời gian xử lý trong trường hợp cộng nhiễu (giây)*

Ảnh test	MP			OMP			DRMP		
	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM	Gauss	Bernoulli	BPNSM
Cameraman	50.92	89.18	83.71	2.54	4.34	2.43	2.87	3.48	2.85
Lena	51.16	93.73	123.48	2.57	4.28	2.31	4.57	3.17	2.40
Phantom	51.43	85.20	84.10	2.55	5.43	2.48	2.06	2.13	1.55
Bone	50.73	81.36	93.14	2.52	5.26	2.43	2.44	4.99	2.44
Saturn	52.20	164.9	93.93	2.56	4.77	2.49	2.14	4.20	1.29
Fly	52.48	88.26	165.6	2.53	5.66	2.63	1.65	4.87	1.77

- Trong 3 ma trận lấy mẫu khi tín hiệu được cộng nhiễu thì ma trận Gauss có hiệu quả kém nhất. Ma trận BPNSM có hiệu quả tốt nhất dựa trên 2 bảng 4.6, 4.7.

- Trong trường hợp cộng thêm nhiễu thuật toán MP cũng cho kết quả kém nhất trong 3 thuật toán cả về thời gian thực hiện, sai số trung bình tuyệt đối MSE và tham số PSNR. Giữa thuật toán OMP và thuật toán DRMP có thể nhận thấy thuật toán DRMP có các chỉ tiêu khôi phục tốt hơn trong trường hợp tín hiệu được cộng thêm nhiễu.

- Thời gian thực hiện của 2 thuật toán OMP và DRMP gần như tương đương nhau trong trường hợp cộng nhiễu. Khi so sánh thời gian thực hiện quá trình lấy mẫu nén với 3 ma trận Gauss, Bernoulli và ma trận BPNSM kết hợp với thuật toán DRMP thì thời gian thực hiện giữa ma trận Gauss và ma trận BPNSM là tương đương nhau, thời gian thực hiện với ma trận ma trận Bernolli là lâu nhất.

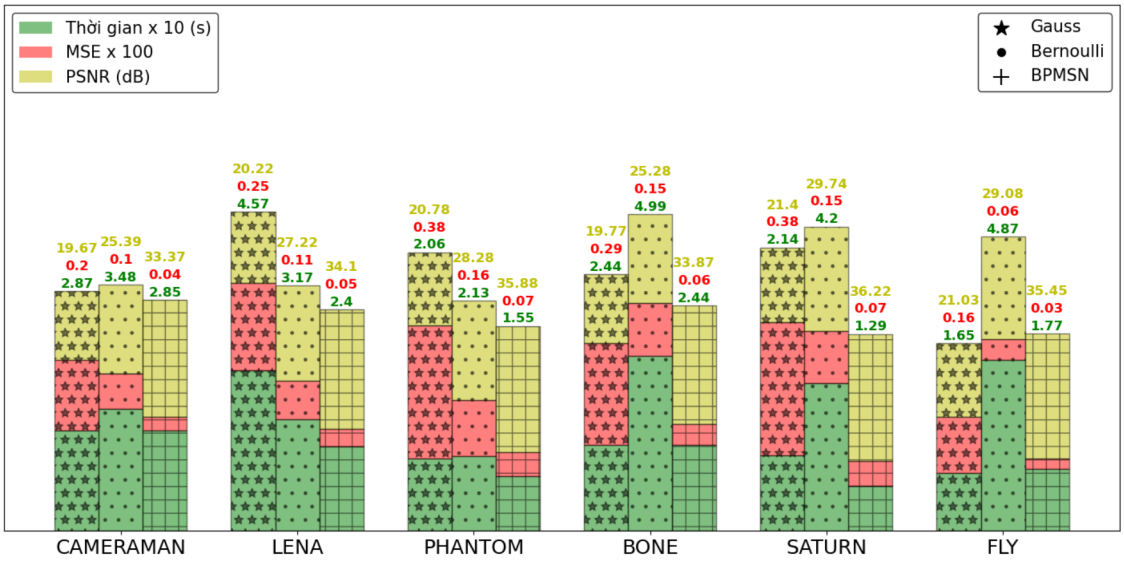
Hình 4.19 thể hiện một số hình ảnh đã được khôi phục trong trường hợp cộng nhiễu vào bức ảnh gốc. Bằng trực quan có thể nhận xét, hình ảnh được lấy mẫu và khôi phục bằng mô hình kết hợp giữa thuật toán DRMP và ma trận BPNSM cho chất lượng tốt nhất khi so với khi sử dụng thuật toán DRMP với 2 ma trận Gauss và Bernoulli.

Đồ thị trong hình 4.20 mô tả các thông số khi thực hiện lấy mẫu nén trong điều kiện cộng thêm nhiễu với 3 loại ma trận là Gauss, Bernoulli và ma trận

(a) *(b)* *(c)*

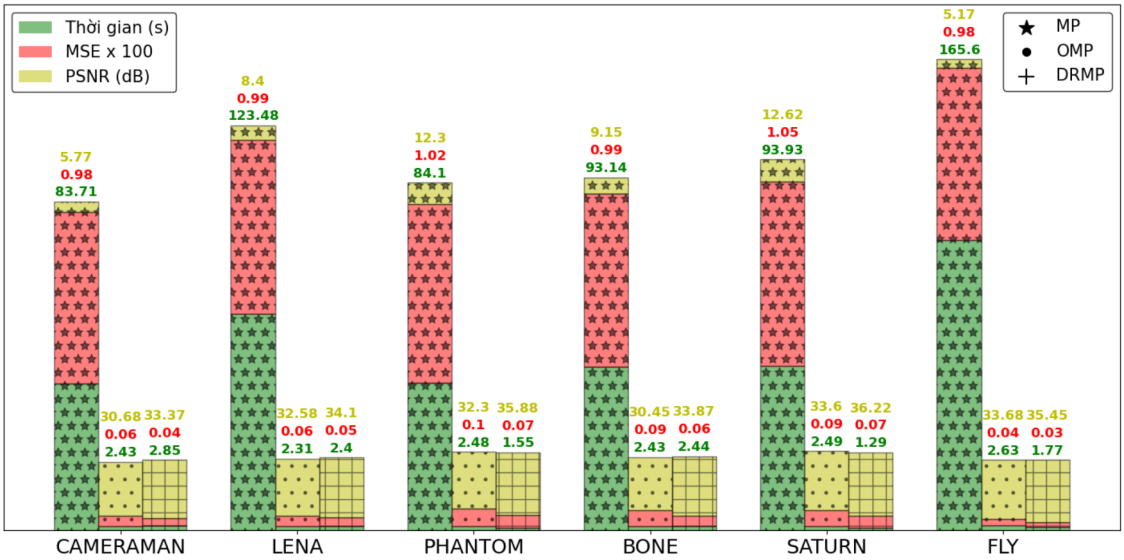
Hình 4.19: *Ảnh khôi phục bằng thuật toán DRMP trong trường hợp cộng thêm nhiễu. 4.19a sử dụng ma trận Gauss, 4.19b sử dụng ma trận Bernoulli, 4.19c sử dụng ma trận BPNSM*

được đề xuất trong luận án BPNSM. Quá trình khôi phục lại sử dụng thuật toán DRMP. Có thể dễ nhận thấy trong thực nghiệm này việc sử dụng ma trận lấy mẫu BPNSM có chỉ số lỗi MSE thấp hơn đáng kể so với 2 ma trận còn lại.



Hình 4.20: *Đồ thị đánh giá 3 ma trận lấy mẫu trong trường hợp cộng nhiễu*

Đồ thị trong hình 4.21 mô tả các thông số khi thực hiện lấy mẫu nén với cùng một ma trận BPNSM nhưng được thực hiện với 3 thuật toán MP, OMP và thuật toán được đề xuất DRMP trong trường hợp cộng nhiễu. Có thể thấy



Hình 4.21: *Đồ thị đánh giá 3 thuật toán lấy mẫu trong trường hợp cộng nhiễu*

thuật toán DRMP và thuật toán OMP cho kết quả gần như tương đương trong thực nghiệm này. Trong khi đó thuật toán MP có thời gian lâu, sai số MSE lớn và tham số PSNR nhỏ.

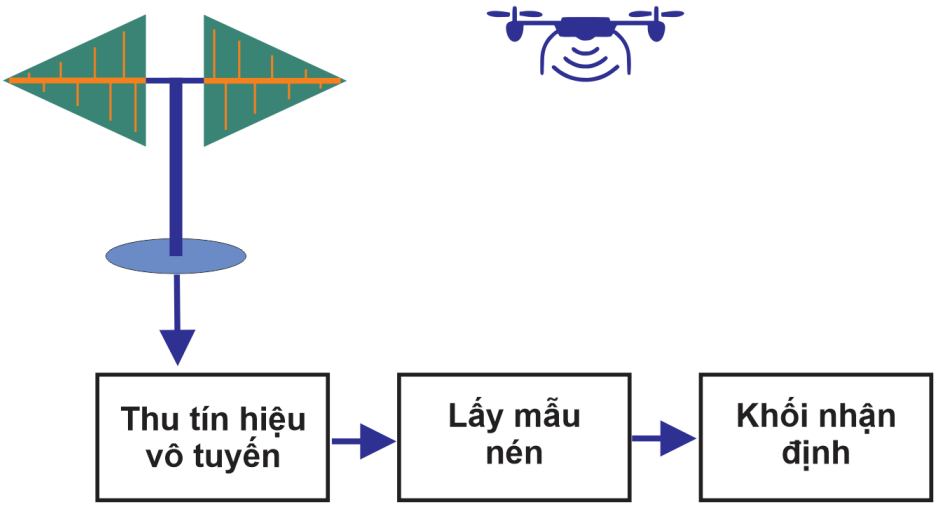
4.4. Ứng dụng mô hình lấy mẫu nén đề xuất

Từ các phân tích, mô phỏng đánh giá đối với mô hình lấy mẫu nén được đề xuất, để xem xét đến khả năng ứng dụng của mô hình lấy mẫu nén trong thực tế luận án cũng trình bày hai ứng dụng sử dụng lấy mẫu nén, ứng dụng thứ nhất là thu tín hiệu vô tuyến phát ra từ flycam và ứng dụng còn lại là lấy mẫu ảnh trong hệ thống camera giám sát.

4.4.1. Ứng dụng trong cảm nhận phổ băng rộng

Trong ví dụ này trình bày một phương pháp lấy mẫu tín hiệu vô tuyến thu được từ Flycam, ứng dụng trong thiết kế hệ thống thiết bị phát hiện, cảnh báo nhằm ứng phó kịp thời đối với những nguy cơ mà loại phương tiện này gây ra. Phương pháp này sử dụng kỹ thuật lấy mẫu nén để giảm khối lượng

tính toán trong xử lý tín hiệu vô tuyến băng rộng thu được. Một hệ thống máy thu được cũng xây dựng trên nền tảng vô tuyến cấu hình mềm (SDR) để thử nghiệm phương pháp này.



Hình 4.22: *Sơ đồ khối hệ thống thu tín hiệu vô tuyến từ Flycam*

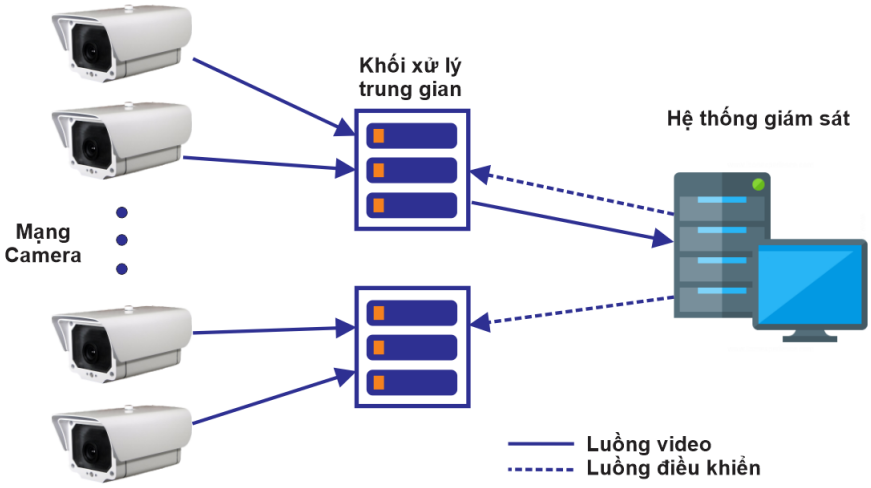
Hệ thống trong hình 4.22 sử dụng anten Yagi định hướng có độ rộng búp sóng là 70^o và hệ số tăng ích $9dBi$. Bộ thu tín hiệu vô tuyến sử dụng nền tảng vô tuyến cấu hình mềm USRP thực hiện việc lấy mẫu sau chuyển mạch anten. Bộ xử lý tín hiệu thực hiện quá trình lấy mẫu nén. Tín hiệu sau lấy mẫu nén được gửi đến bộ xử lý trung tâm để khôi phục lại tín hiệu gốc và đưa ra kết luận. Các kết quả chi tiết của ứng dụng này được trình bày trong [J3], [J4].

4.4.2. Ứng dụng lấy mẫu nén ảnh số

Các hệ thống giám sát video phân tán quy mô lớn ngày càng được sử dụng rộng rãi [53]. Ví dụ như các hệ thống giám sát giao thông, giám sát đường cao tốc hay giám sát cho nhà máy xí nghiệp. Các hệ thống như vậy đặt ra những thách thức về khả năng mở rộng, đáp ứng kịp thời cũng như khả năng bảo mật dữ liệu trên đường truyền. Do số lượng nguồn video trong các hệ thống

như vậy là rất lớn, băng thông cần thiết để truyền các luồng video giám sát có thể vượt khả năng đáp ứng của đường truyền, nhất là trong các điều kiện đường truyền không được ổn định. Trong các hệ thống này các tiêu chí xử lý đánh giá đối với chất lượng video thường dựa trên các thuật toán thị giác máy tính để tự động hóa các nhiệm vụ giám sát.

Trong phần này của luận án trình bày việc áp dụng kỹ thuật lấy mẫu nén trong quá trình xử lý tín hiệu video giám sát trước khi truyền đi. Kỹ thuật này chứng tỏ rằng tốc độ bit trên đường truyền giảm đi đáng kể trong khi đó chất lượng hình ảnh vẫn được đảm bảo khi đánh giá bằng các thuật toán thị giác máy tính sử dụng các tham số như PSNR, MSE, SSIM.



Hình 4.23: *Kiến trúc của hệ thống giám sát video phân tán [53]*

Có thể nhận thấy rằng trong các bức ảnh giám sát thì sự thay đổi giữa hình ảnh trước và sau là ít và không đáng kể so với dung lượng của toàn bức ảnh. Trong trường hợp này giả định rằng nội dung truyền đi trên đường truyền chỉ là sự sai khác giữa 2 khung hình như trong hình 4.24. Khi đó dữ liệu truyền đi chỉ là hình ảnh như trong hình 4.25, về trực quan là rất thưa và có thể nén được với tỉ số nén lớn bằng việc sử dụng kỹ thuật lấy mẫu nén. Kết quả của mô hình này có thể giảm đáng kể tốc độ bit trên đường truyền hình ảnh. Quá trình tiền xử lý đối với tín hiệu này chỉ là phép trừ ma trận

nên độ phức tạp tính toán là thấp. Ở phía nhận khôi phục lại tín hiệu thưa này và cộng với dữ liệu về hình ảnh nhận được liền trước, khi đó sẽ thu được bức ảnh gốc tại thời điểm hiện tại.

***(a)** Ảnh trước* *(b) Ảnh sau*

***Hình 4.24:** Hình ảnh giám sát*

***Hình 4.25:** Phần sai khác giữa 2 ảnh*

Giả sử có tín hiệu hình ảnh tại thời điểm k hiện tại ký hiệu là $v^k[i,j]$. Khi đó tín hiệu thưa thu được khi thực hiện phép trừ ma trận sẽ có dạng

$$x^k[i,j] = v^k[i,j] - v^{k-1}[i,j]. \tag{4.6}$$

Tín hiệu này có $K - sparse$ rất thấp và sẽ có hiệu quả cao khi thực hiện với mô hình lấy mẫu nén. Kết quả của ứng dụng được trình bày trong [J2].

4.5. Tổng kết chương

Từ các thực nghiệm với tín hiệu vô tuyến và tín hiệu ảnh được trình bày trong mục (4.4.1) và (4.4.2) có thể nhận xét:

- Ma trận xác định BPNSM có hiệu quả rõ rệt khi được so sánh với ma trận Gauss và ma trận Bernoulli. Sử dụng ma trận này hiệu quả cả về mặt thời gian và độ chính xác trong quá trình khôi phục lại tín hiệu.
- Thuật toán DRMP có thời gian xử lý gần như tương đương với thuật toán OMP nhưng cho kết quả có độ chính xác cao hơn và khả năng chống nhiễu tốt hơn so với thuật toán OMP và MP.
- Thuật toán DRMP được cải tiến từ thuật toán gốc MP và có hiệu quả tốt hơn so với thuật toán gốc MP cả về độ chính xác và thời gian thực hiện.
- Thuật toán gốc MP có thời gian khôi phục lớn do khối lượng tính toán lớn, lỗi khôi phục cao nhất là đối với các tín hiệu có giá trị $K - sparse$ cao và tỉ số SNR thấp.

Các thực nghiệm trong luận án này đều được thực hiện trên máy tính cá nhân nên chưa phát huy hết được khả năng tận dụng bộ nhớ của ma trận BPNSM. Nếu quá trình tạo ma trận được thực hiện trên các phần cứng khả trình như FPGA có thể làm tốc độ lấy mẫu được nâng cao hơn nữa so với việc sử dụng ma trận Gauss và ma trận Bernoulli.

KẾT LUẬN VÀ KIẾN NGHỊ

Nội dung luận án đã đạt được mục tiêu đề ra là nghiên cứu đề xuất một mô hình lấy mẫu nén gồm ma trận lấy mẫu nén xác định khả thi khi thực hiện với các hệ thống điện tử số, thuật toán khôi phục tín hiệu được lấy mẫu nén cải tiến từ thuật toán gốc MP. Toàn bộ các kiến thức nền tảng và các kết quả nghiên cứu đã được trình bày chi tiết trong luận án với bố cục gồm bốn chương: Chương 1 trình bày tổng quan về lấy mẫu nén, Chương 2 thiết kế ma trận lấy mẫu nén xác định BPNSM, Chương 3 trình bày thuật toán cải tiến DRMP, Chương 4 mô phỏng, đánh giá hiệu năng của ma trận và thuật toán đề xuất. Các kết quả đóng góp mới về khoa học của luận án có thể phân thành ba nhóm như sau:

1. Đề xuất phương pháp xây dựng ma trận lấy mẫu nén BPNSM.

Đề xuất một kỹ thuật xây dựng ma trận lấy mẫu nén BPNSM với các phần tử được tạo thành từ các chuỗi nhị phân phi tuyến. Các nghiên cứu trước đây chưa tập trung vào tính bảo mật và các kỹ thuật thực hiện ma trận trên phần cứng với tốc độ cao. Việc tạo ra ma trận từ các dãy phi tuyến giả ngẫu nhiên làm cho ma trận lấy mẫu nén có tính bảo mật cao, kỹ thuật tạo ra ma trận từ phần cứng FPGA kết hợp với hệ thống chuyển mạch giúp giảm thời gian tạo ra ma trận, giảm độ phức tạp phần cứng khi thực hiện lấy mẫu nén. Ma trận BPNSM được tạo ra từ các đa thức sinh trên trường Galois nên yêu cầu về lưu trữ thấp, trong các ứng dụng truyền thông thay vì truyền đi giá trị toàn bộ các phần tử trong ma trận thì đối với ma trận BPNSM chỉ cần truyền đi đa thức sinh để tạo ra ma trận đó.

2. Đề xuất thuật toán cải tiến DRMP dựa trên thuật toán gốc MP.

Đề xuất thuật toán cải tiến DRMP dựa trên thuật toán gốc MP. Thuật toán DRMP cải tiến so với thuật toán MP ở bước tìm kiếm phần dư và cập

nhật giá trị trong mỗi một bước lặp. Các thành phần của tín hiệu được tìm kiếm từ không gian con của ma trận lấy mẫu nén thông qua hướng tối đa hóa biến đổi *gradient* của hàm mục tiêu thay vì việc tính tích vô hướng của tất cả các cột trong ma trận lấy mẫu như đối với thuật toán gốc MP. Thông qua các phân tích đánh giá về mặt toán học cũng như mô phỏng thực nghiệm, thuật toán DRMP được đề xuất có lỗi trong quá trình khôi phục giảm dần sau mỗi lần lặp, quá trình tính toán ở mỗi bước lặp đơn giản hơn so với thuật toán gốc MP trong trường hợp ma trận lấy mẫu thỏa mãn điều kiện giới hạn đẳng trị RIP.

3. Đề xuất mô hình lấy mẫu nén gồm ma trận BPNSM và thuật toán cải tiến DRMP, so sánh đánh giá hiệu năng của mô hình đề xuất.

Từ ma trận BPNSM và thuật toán cải tiến DRMP luận án đã đề xuất mô hình lấy mẫu nén sử dụng 2 phần tử này. Hiệu năng của mô hình đề xuất được kiểm chứng thông qua mô phỏng với các nguồn tín hiệu đầu vào phổ biến trong thực tế gồm tín hiệu vô tuyến và tín hiệu ảnh đa cấp xám. Tính khả thi của mô hình được trình bày thông qua 2 ứng dụng là lấy mẫu nén tín hiệu vô tuyến từ Flycam và lấy mẫu nén hình ảnh trong hệ thống camera giám sát.

Kết quả nghiên cứu trong luận án phù hợp với những kết quả được trình bày trong phần thực nghiệm, đây cũng là định hướng để có thể mở rộng các nghiên cứu về sau. Hướng nghiên cứu tiếp theo của luận án sẽ tập trung vào lấy mẫu nén thích nghi để nâng cao hiệu quả cho quá trình tính toán và lưu trữ. Sử dụng các công cụ học máy, trí tuệ nhân tạo để tối ưu hóa ma trận lấy mẫu dựa trên một tập các dữ liệu từ tín hiệu đầu vào.

Ngoài ra, trong quá trình nghiên cứu nghiên cứu sinh đã lĩnh hội và tích lũy được các kiến thức về lý thuyết lấy mẫu nén, lý thuyết về đại số tuyến tính, các kỹ năng về tính toán và xử lý số tín hiệu. Đây là những công cụ không thể thiếu trong việc nghiên cứu về xử lý tín hiệu số, và có thể phục vụ lâu dài cho công tác nghiên cứu và giảng dạy của nghiên cứu sinh.

DANH MỤC CÁC CÔNG TRÌNH CÓ LIÊN QUAN ĐẾN LUẬN ÁN

[J1] Tran Vu Kien, Nguyen Ngoc Minh, Nguyen Le Cuong, "A Development toward Matching Pursuit Algorithm Aims To Reduce Calculation Mass in the Process of the Compessed Sampling and Errors in the Signal Recovery Process", Journal of Science & Technology 120 (2017) 072-077

[J2] Kien. T.V, An. B.L, Quynh. L.C, "Secure Separate Bit Plane Image Processing for Distributed Video Surveillance System (DVSS/DVC)", IJCSMC, Vol. 9, Issue. 9, September 2020, pg.61 - 72

[J3] Trần Vũ Kiên, Hoàng Văn Lợi, Nguyễn Lê Cường, "Ứng dụng kỹ thuật lấy mẫu nén trong việc thu tín hiệu vô tuyến để phát hiện máy bay không người lái siêu nhẹ", Tạp chí Nghiên cứu KH&CN quân sự, Số Đặc san Ra đa, 8 - 2021, ISSN 1859 - 1403.

[J4] Trần Vũ Kiên, Đỗ Anh Tú, Nguyễn Hải Quân, Nguyễn Lê Cường, "Một phương pháp ứng dụng mẫu nén và học máy để phát hiện Flycam trong môi trường có chồng lấn với tín hiệu WiFi", Tạp chí Nghiên cứu KH&CN quân sự, Số 82, 10 - 2022, ISSN 1859 - 1403.

[C1] Cuong Nguyen Le, Kien Tran Vu and Quynh Le Chi, "On The Desired Properties Of Linear Feedback Shift Register (LFSR) Based High-Speed PN-Sequence-Generator", ICTIS 2020.

TÀI LIỆU THAM KHẢO

[1] Abebe A. T. and Kang C. G., "Iterative Order Recursive Least Square Estimation for Exploiting Frame-Wise Sparsity in Compressive Sensing-Based MTC", in IEEE Communications Letters, vol. **20**, no. 5, pp. 1018-1021, (May 2016), doi: 10.1109/LCOMM.2016.2539255.

[2] Abo-Zahhad, Mohammed M., Aziza I. Hussein, and Abdelfatah M. Mohamed, "Compressive sensing algorithms for signal processing applications: A survey", International journal of communications, network and system sciences 8.06 (2015): 197.

[3] Agarwal A., Negahban S., and Wainwright M. J., "Fast global convergence of gradient methods for high-dimensional statistical recovery", Ann. Statist., **40**(5):, (2012) 2452-2482.

[4] Allahham, MHD Saria, et al, "DroneRF dataset: A dataset of drones for RF-based detection, classification and identification", Data in brief 26 (2019): 104313.

[5] Amini, Arash, and Farokh Marvasti, "Deterministic construction of binary, bipolar, and ternary compressed sensing matrices", IEEE Transactions on Information Theory 57.4 (2011): 2360-2370.

[6] Arjoune Youness, et al, "Compressive sensing: Performance comparison of sparse recovery algorithms", 2017 IEEE 7th annual computing and communication workshop and conference (CCWC). IEEE, (2017).

[7] Arjoune Y., Kaabouch N., El Ghazi H., & Tamtaoui A., "A performance comparison of measurement matrices in compressive sensing", International Journal of Communication Systems, **31**(10), e3576. doi:10.1002/dac.3576 (2018).

[8] Arjoune Y., Kaabouch N., “Wideband Spectrum Sensing: A Bayesian Compressive Sensing Approach”, Sensors (2018), 18, 1839; doi:10.3390/s18061839.

[9] Baraniuk R., “Compressive sensing”, IEEE Sig. Proc. Mag. **24** (2007), no. 4, 118-121.

[10] Baraniuk R., Davenport M., DeVore R., and Wakin M., “A simple proof of the restricted isometry property for random matrices”, Constructive Approximation **28** (2008), no. 3, 253-263.

[11] Baraniuk R., Cevher V., Duarte M. F., & Hegde C., “Model-based compressive sensing”, IEEE Transactions on information theory, **56**(4), (1982-2001).

[12] Berardinelli G., “Generalized DFT-s-OFDM waveforms without Cyclic Prefix”, IEEE Access (2017), 6, 4677–4689.

[13] Blumensath, Thomas, and Mike E. Davies, “Gradient pursuits”, IEEE Transactions on Signal Processing **56**.6 (2008): 2370-2382.

[14] Candès E. J., Romberg J., “L1-magic: Recovery of Sparse Signals via Convex Programming”, (2005), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.212.9120.

[15] Candès E. J., Romberg J., and Tao T., “Robust uncertainty principles: Ex act signal reconstruction from highly incomplete frequency information”, IEEE Transactions on Information Theory, vol. **52**, pp. 489—509, (Feb 2006).

[16] Candès E. J. and Romberg J., “Sparsity and incoherence in compressive sampling”, Inverse Prob. **23** (2007), no. 3, 969-985.

[17] Candès E. J., “The restricted isometry property and its implications for compressed sensing”, C. R. Acad. Sci. Paris, Ser. I 346 (2008) 589–592.

[18] Candès E.J., Recht B., “Exact matrix completion via convex optimization”, Foundations of Computational Mathematics 9(6), 717-772 (2009), http://link.springer.com/article/10.1007/s10208-009-9045-5.

[19] Candès E. J. and Plan Y., “A probabilistic and RIPless theory of compressed sensing”, Preprint: arXiv:1011.3854 (2010)

[20] Chen S., Donoho D., and Saunders M., “Atomic decomposition by basis pursuit”, SIAM Journal on Scientific Computing, vol. **20**, no. 1, pp. 33—61, (Jul 2006).

[21] Chen, Bo, John Paisley, and Lawrence Carin. “Sparse linear regression with beta process priors”, 2010 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, (2010).

[22] Cuong Nguyen Le , Thang Pham Xuan, Quynh Le Chi , “On the Comparative Study of Some Mathematical Tools for Specific Sequences Design”, Journal of information Engineering and Applications Vol.**5**, No.12, (2015), p.1-10.

[23] Dai W. and Milenkovic O., “Subspace pursuit for compressive sensing signal reconstruction”, IEEE Trans. Inform. Theory, 55(5):22308211;2249, (2009).

[24] Daubechies, Ingrid, et al. “Iteratively reweighted least squares minimization for sparse recovery”, Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences 63.1 (2010): 1-38.

[25] Davenport M. A. and Wakin M., “Analysis of orthogonal matching pursuit using the restricted isometry property”, IEEE Trans. Inform. Theory, 56(9):43958211;4401, (2010).

[26] Davenport M. A.,Needell D., and Wakin M. B., “Signal space cosamp for sparse recovery with redundant dictionaries”, Preprint at http://arxiv.org/abs/1208.0353, (2012).

[27] David L. Donoho, Yaakov Tsaig, Iddo Drori, Jean-Luc Starck, “Sparse Solution of Underdetermined Linear Equations by Stagewise Orthogonal Matching Pursuit”, IEEE Transactions on Information Theory **58** (2), 1094-121 (2012).

[28] Dhawan, Sanjeev, “Analogy of promising wireless technologies on different frequencies: Bluetooth, wifi, and wimax”, The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007). IEEE, (2007).

[29] Do T. T., Gan L., Nguyen N. and Tran T. D., “Sparsity adaptive matching pursuit algorithm for practical compressed sensing”, 2008 42nd Asilomar Conference on Signals, Systems and Computers, (2008), pp. 581-587, doi: 10.1109/ACSSC.2008.5074472.

[30] Donoho D. and Elad M., “Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization”, Proc. Natl. Acad. Sci., **100**(5):21978211;2202, (2003).

[31] Donoho D., “Compressed sensing”, IEEE Trans. on Information Theory, **52**, no. 4, pp. 1289–1306, (April 2006).

[32] Donoho D., “For most large underdetermined systems of linear equations, the minimal -norm solution is also the sparsest solution”, Comm. Pure Appl. Math., **59**(6):7978211;829, (2006).

[33] Duarte M., Wakin M., and Baraniuk R., “Fast reconstruction of piecewise smooth signals from random projections”, In Proc. Work. Struc. Parc. Rep. Adap. Signaux (SPARS), Rennes, France, (Nov 2005).

[34] Duarte Carvajalino, Julio Martin, and Guillermo Sapiro, “Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization”, IEEE Transactions on Image Processing, **18**(7):1395–1408, (July 2009).

[35] Du L., Wang R.,Wan W., Yu X. Q., Yu S., “Analysis on greedy reconstruction algorithms based on compressed sensing”, 2012 International Conference on Audio, Language and Image Processing. pp. 783-789. IEEE (jul 2012).

[36] Faragallah O. S. et al., “A Comprehensive Survey Analysis for Present Solutions of Medical Image Fusion and Future Directions”, in IEEE Access, vol. 9, pp. 11358-11371, (2021), doi: 10.1109/ACCESS.2020.3048315.

[37] Fardad, Mohammad, Sayed Masoud Sayedi, and Ehsan Yazdian, “A low-complexity hardware for deterministic compressive sensing reconstruction”, IEEE Transactions on Circuits and Systems I: Regular Papers 65.10 (2018): 3349-3361.

[38] Faria M.L.L.D., Cugnasca C.E., Amazonas J.R.A, “Insights into IoT data and an innovative DWT-based technique to denoise sensor signals”, IEEE Sens. J. (2017), 18, 237–247.

[39] Foucart, Simon, “Sparse recovery algorithms: sufficient conditions in terms of restricted isometry constants”, Approximation Theory XIII: San Antonio 2010. Springer, New York, NY, (2012). 65-77.

[40] Gan, Hongping, et al, “Chaotic compressive sampling matrix: where sensing architecture meets sinusoidal iterator”, Circuits, Systems, and Signal Processing 39.3 (2020): 1581-1602.

[41] Gian Quoc, Anh, et al, “Wireless technology for monitoring site-specific landslide in Vietnam”, International Journal of Electrical and Computer Engineering 8.6 (2018): 4448-4455.

[42] Goresky, Mark, and Andrew M. Klapper, “Fibonacci and Galois representations of feedback-with-carry shift registers”, IEEE Transactions on Information Theory 48.11 (2002): 2826-2836.

[43] Gu, Zhi, et al, “Deterministic compressed sensing matrices from sequences with optimal correlation”, IEEE Access 7 (2019): 16704-16710.

[44] Gurve, Dharmendra, Denis Delisle-Rodriguez, Teodiano Bastos-Filho, and Sridhar Krishnan, “Trends in Compressive Sensing for EEG Signal Processing Applications”, Sensors 20, no. 13: 3703. https://doi.org/10.3390/s20133703 (2020).

[45] Herman M. and Strohmer T., “High-resolution radar via compressed sensing”, IEEE Trans. Sig. Proc. **57** (2009), no. 6, 2275-2284.

[46] Hong M., Yu Y., Wang H., Liu F., Crozier S., “Compressed sensing MRI with singular value decomposition-based sparsity basis”, Physics in Medicine and Biology, **56** (2011) 6311–6325, doi: 10.1088/0031-9155/56/19/010.

[47] Howard, Stephen D., A. Robert Calderbank, and Stephen J. Searle, “A fast reconstruction algorithm for deterministic compressive sensing using second order Reed-Muller codes”, 2008 42nd Annual Conference on Information Sciences and Systems. IEEE, (2008).

[48] Jarvis Haupt, Waheed U. Bajwa, Gil Raz and Robert Nowak, “Toeplitz compressed sensing matrices with applications to sparse channel estimation”, IEEE Transactions on Information Theory, vol. **56**, no. 11, pp. 5862-5875, November, (2010).

[49] Jean Bourgain, Stephen Dilworth, Kevin Ford, Sergei Konyagin and Denka Kutzarova, “Explicit constructions of RIP matrices and related problems”, Duke Mathematical Journal, vol. 159, no. 1, pp. 145-185, (2011).

[50] Ker, Justin, et al. “Deep learning applications in medical image analysis”, Ieee Access 6 (2017): 9375-9389.

[51] Khayam, Syed Ali, “The discrete cosine transform (DCT): theory and application”, Michigan State University 114 (2003): 1-31.

[52] Kong L., Zhang D., He Z., Xiang Q., Wan J. and Tao M., “Embracing big data with compressive sensing: a green approach in industrial wireless networks”, in IEEE Communications Magazine, vol. 54, no. 10, pp. 53-59, (October 2016), doi: 10.1109/MCOM.2016.7588229.

[53] Korshunov, Pavel & Ooi, Wei Tsang, “Critical video quality for distributed automated video surveillance”, Proceedings of the 13th ACM International Conference on Multimedia, (MM 2005). 151-160. 10.1145/1101149.1101172.

[54] Lawgaly A., Khelifi F., “Sensor pattern noise estimation based on improved locally adaptive DCT filtering and weighted averaging for source camera identification and verification view document”, IEEE Trans. Inf. Forensics Secur. (2017), 12, 392–404.

[55] Li, Shuxing, et al, “Deterministic construction of compressed sensing matrices via algebraic curves”, IEEE Transactions on Information Theory 58.8 (2012): 5035-5041.

[56] Li, Shuxing, and Gennian Ge. “Deterministic construction of sparse sensing matrices via finite geometry”, IEEE transactions on signal processing 62.11 (2014): 2850-2859.

[57] Li, Lixiang, et al, “Overview of compressed sensing: sensing model, reconstruction algorithm, and its applications”, Applied Sciences 10.17 (2020): 5909.

[58] Linh Trung, Nguyen, et al, “Compressed sensing using chaos filters”, 2008 Australasian Telecommunication Networks and Applications Conference. IEEE, (2008).

[59] Lotfi, Mahsa, and Mathukumalli Vidyasagar, “Compressed sensing using binary matrices of nearly optimal dimensions”, IEEE Transactions on Signal Processing 68 (2020): 3008-3021.

[60] Lu, Weizhi, Tao Dai, and Shu-Tao Xia, “Binary matrices for compressed sensing”, IEEE Transactions on Signal Processing 66.1 (2017): 77-85.

[61] Lustig M., Donoho D., and Pauly J. M., “Sparse MRI: The application of compressed sensing for rapid MR imaging”, Magn. Reson. Med. **58** (2007), no. 6, 1182-1195.

[62] Lustig M., Donoho D.L., Santos J.M., Pauly J.M., “Compressed Sensing MRI”, IEEE Signal Processing Magazine, (2008); 25(2): 72-82, DOI: 10.1109/MSP.2007.914728.

[63] Mallat S.G, Zhang Z, “ Matching pursuits with time-frequency dictionaries ”, IEEE Transactions on Signal Processing **41**(12), 3397-3415 (1993).

[64] Manchanda R. and Sharma K., “A Review of Reconstruction Algorithms in Compressive Sensing”, 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), (2020), pp. 322-325, doi: 10.1109/ICACCM50413.2020.9212838.

[65] Marques, Elaine Crespo, et al, “A review of sparse recovery algorithms”, IEEE access 7 (2018): 1300-1322.

[66] Meyer, François G., Amir Z. Averbuch, and J-O. Stromberg, “Fast adaptive wavelet packet image compression”, IEEE Transactions on image processing 9.5 (2000): 792-800.

[67] MIT Tech. Rev. Magazine, (2007), “Ten emerging technologies 2007”,

[68] Michael Elad, “Optimized projections for compressed sensing”, IEEE Transactions on Signal Processing, **55**(12):5695–5702, (December 2007). http://www.technologyreview.com/specialreports/specialreport.aspx?id=19.

[69] Milne, James S, “Fields and Galois theory”, (2020).

[70] Mousavi, Ahmad, Mehdi Rezaee, and Ramin Ayanzadeh, “A survey on compressive sensing: Classical results and recent advancements”, arXiv preprint arXiv:1908.01014 (2019).

[71] Nam Yul Yu and Guang Gong, “A new binary sequence family with low correlation and large size”, IEEE Transactions on Information Theory, vol. **52**, no. 4, pp. 1624-1636, (April, 2006).

[72] Needell D., Tropp J.A., “ CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”, Applied and Computational Harmonic Analysis, **26** (3), 301-321 (may 2009).

[73] Needell D. and Vershynin R., “Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit”, IEEE J. Select. Top. Signal Processing, 4(2):3108211;316,(2010).

[74] Negahban S., Ravikumar P., Wainwright M. J., and Yu B., “A unied framework for highdimensional analysis of M-estimators with decomposable regularizers”, Stat. Scien., **27**(4):, (2012) 538-557.

[75] Ngoc Minh Nguyen, Nam Nguyen, “An Iterative Greedy Algorithm for parsity-Constrained Optimization”, Tạp chí nghiên cứu KH&CN quân sự, số **40** (2015), tr. 109-117.

[76] Nguyen, Thu LN, and Yoan Shin, “Deterministic sensing matrices in compressive sensing: a survey”, The Scientific World Journal 2013 (2013).

[77] Nguyen, Minh Tuan, and Keith A. Teague, “Compressive sensing based data gathering in clustered wireless sensor networks”, 2014 IEEE International Conference on Distributed Computing in Sensor Systems. IEEE, (2014).

[78] Nguyen, Dinh-Chinh, Tran Duc-Tan, and Duc-Nghia Tran, “Application of compressed sensing in effective power consumption of WSN for landslide scenario”, 2015 Asia Pacific Conference on Multimedia and Broadcasting. IEEE, (2015).

[79] Pati Y., Rezaifar R., and Krishnaprasad P., “Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition”, In Proc. Asilomar Conf. Signals, Systems, and Computers, Pacific Grove, CA, Nov. (1993).

[80] Pedregosa F. and Varoquaux G., and Gramfort A. and Michel V. and Thirion B. and Grisel O. and Blondel M. and Prettenhofer P. and Weiss R. and Dubourg V. and Vanderplas J. and Passos A. and Cournapeau D. and Brucher M. and Perrot M. and Duchesnay E., “Scikit-learn: Machine Learning in Python”, Journal of Machine Learning Research, vol. **12**, 2825-2830, (2011).

[81] Qaisar, Saad, et al, “Compressive sensing: From theory to applications, a survey”, Journal of Communications and networks (15.5 2013): 443-456.

[82] Rani, Meenu, Sanjay B. Dhok, and Raghavendra B. Deshmukh, “A systematic review of compressive sensing: Concepts, implementations and applications”, IEEE Access 6 (2018): 4875-4894.

[83] Rauhut, Holger, “Circulant and Toeplitz matrices in compressed sensing”, arXiv preprint arXiv:0902.4394 (2009).

[84] Applebaum, Lorne, et al, “Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery”, Applied and Computational Harmonic Analysis (26.2 2009): 283-290.

[85] Salahdine F., Kaabouch N., El Ghazi H., “A Bayesian recovery with Toeplitz matrix for compressive spectrum sensing in cognitive radio networks”, International Journal of Communications Systems, 30, (15), pp 1–13, (2017).

[86] Salari, Soheil, et al, “TDOA estimation with compressive sensing measurements and Hadamard matrix”, IEEE Transactions on Aerospace and Electronic Systems 54.6 (2018): 3137-3142.

[87] Sankaranarayanan, Aswin C., and Richard G. Baraniuk, “Compressive sensing”, Computer vision: a reference Guide (2020): 1-5.

[88] Sejdić, Ervin, Irena Orović, and Srdjan Stanković, “Compressive sensing meets time–frequency: An overview of recent advances in time–frequency processing of sparse signals”, Digital signal processing 77 (2018): 22-35.

[89] Sharma S. K., Lagunas E., Chatzinotas S., & Ottersten B. “Application of Compressive Sensing in Cognitive Radio Communications: A Survey”, IEEE Communications Surveys & Tutorials, **18**(3), (2016), 1838–1860, doi:10.1109/comst.2016.2524443.

[90] Shen Y., Li S., “Restricted p–isometry property and its application for nonconvex compressive sensing”, Adv Comput Math. (2012) 37:441–52. doi: 10.1007/s10444-011-9219-y.

[91] Su, Xu, et al, “An improved Toeplitz measurement matrix for compressive sensing”, International Journal of Distributed Sensor Networks 10.6 (2014): 846757.

[92] Su X., Hongpeng Y., Yi C., Yushu X., Xue T., “An improved Toeplitz measurement matrix for compressive sensing”, International Journal of Distributed Sensor Networks, , 10, (6), pp 1-9, (2014).

[93] Sun H., Nallanathan A., Wang C.X., Chen Y., “Wideband spectrum sensing for cognitive radio networks: A survey”, IEEE Wirel. Commun. (2013), 20, 74–81.

[94] Tehrani, Arash Saber, Alexandros G. Dimakis, and Giuseppe Caire, “Optimal deterministic compressed sensing matrices”, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, (2013).

[95] Tong, Fenghua, et al, “Deterministic Constructions of Compressed Sensing Matrices From Unitary Geometry”, IEEE Transactions on Information Theory 67.8 (2021): 5548-5561.

[96] Tropp J.A, Gilbert A.C, “ Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit ”, IEEE Transactions on Information Theory **53**(12),4655-4666 (dec 2007).

[97] Tropp JA et al, “Beyond Nyquist: efficient sampling of sparse bandlimited signals”, IEEE Trans Info Theory Jan. (2010);56(1):520-44.

[98] Tsiligianni, Evaggelia V., Lisimachos P. Kondi, and Aggelos K. Katsaggelos. “Construction of incoherent unit norm tight frames with application to compressed sensing”, IEEE Transactions on Information Theory 60.4 (2014): 2319-2330.

[99] Wakin M. B. et al., “An Architecture for Compressive Imaging”, 2006 International Conference on Image Processing, Atlanta, GA, (2006), pp. 1273-1276, doi: 10.1109/ICIP.2006.312577.

[100] Wei Chen, Miguel R.D. Rodrigues, and Ian J. Wassell, “Projection design for statistical compressive sensing: A tight frame based approach”, IEEE Transactions on signal processing, **61**(8):2016–2029, (April 2013).

[101] Welch L., “Lower bounds on the maximum cross correlation of signals”, IEEE Trans. Inform. Theory, **20**(3):3978211;399, (1974).

[102] Wimalajeewa, Thakshila, and Pramod K. Varshney, “Application of compressive sensing techniques in distributed sensor networks: A survey”, arXiv preprint arXiv:1709.10401 (2017).

[103] Xu, Guangwu, and Zhiqiang Xu, “Compressed sensing matrices from Fourier matrices”, IEEE Transactions on Information Theory 61.1 (2014): 469-478.

[104] Yang, Junfeng, and Yin Zhang, “Alternating direction algorithms for ℓ_1-problems in compressive sensing”, SIAM journal on scientific computing 33.1 (2011): 250-278.

[105] Yin W., Wen Z., Li S., “Dynamic compressive spectrum sensing for cognitive radio networks”, In Proceedings of the IEEE Information Sciences and Systems Conference, Baltimore, MD, USA, (23–25 March 2011); pp. 1–6.

[106] Yin, Penghang, and Jack Xin, “Iterative ℓ_1 minimization for non-convex compressed sensing”, arXiv preprint arXiv:1604.07924 (2016).

[107] Zhang, Gesen, et al, “Compressed sensing and reconstruction with bernoulli matrices”, The 2010 IEEE International Conference on Information and Automation. IEEE, (2010).

[108] Zukovic S., Medenica M., Draganic A., Orovic I., Stankovic S., “A Virtual Instrument for Compressive Sensing of Multimedia Signals”, 56th International Symposium ELMAR 2014, Zadar, Croatia, (2014), DOI: 10.1109/ELMAR.2014.6923339.

[109] Zhang, Jun, Guojun Han, and Yi Fang, “Deterministic construction of compressed sensing matrices from protograph LDPC codes”, IEEE Signal Processing Letters 22.11 (2015): 1960-1964.

PHỤ LỤC

Trong luận án này, tác giả đã sử dụng ngôn ngữ lập trình Python để mô phỏng tạo dãy M, tính hàm vết, thứ tự lồng ghép, ghép dãy phi tuyến, trộn dãy để xem xét đặc tính ngẫu nhiên, giá trị tương quan giữa các dãy để tạo ma trận lấy mẫu trong Chương 2, và lập trình các thuật toán khôi phục trong Chương 4. Ngôn ngữ Python hiện nay là ngôn ngữ phổ biến nhất trên toàn thế giới và tích hợp rất nhiều công cụ tiện ích cho đa số các lĩnh vực nghiên cứu như viễn thông, trí tuệ nhân tạo, thống kê, xử lý tín hiệu số...

Phụ lục 1: Chương trình tạo chuỗi PN

```
import numpy as np
from scipy.stats import pearsonr
def vet_cal(a,g):
  l = len(a)
  n = len(g)
  m = l-n
  y = np.zeros(n, dtype=int)
  x = np.zeros(n, dtype=int)
  b = np.zeros(n, dtype=int)
  z = a[0:n]
  for i in range(m):
    if z[0] == 0:
          for j in range(n):
            y[j] = z[j] ^ b[j]
        else:
          for j in range(n):
                y[j] = z[j] ^ g[j]
        z = np.concatenate((y[1:n], a[n+i]), axis=None)
  if z[0] == 0:
    for j in range(n):
      x[j] = z[j] ^ b[j]
  else:
    for j in range(n):
    x[j] = z[j] ^ g[j]
```

```
  return x
############################################################
def LFSR(s,t,M_length,sub_degree):
  n = len(s)
  m = len(t)
  c = np.zeros((M_length, sub_degree))
  c[0, :] = s
  b = np.zeros(m, dtype=int)
  for k in range(1, pow(2, n) - 1):
    b[0] = int(s[t[0]-1]) ^ int(s[t[1]-1])
    if m > 2:
      for i in range(1, m - 2):
        b[i] = s[t[i + 1]] ^ b[i-1]

    for j in range(1, n):
      s[n - j] = s[n - j - 1]
    s[0] = b[m - 2]
    c[k, :] = s
  seq = c[:, n-1]
  return seq
############################################################
def phase(s_initial, d, M_length, sub_dgree):
  s = s_initial
  d_length = len(d)
  t = []
  for i_d in range(1, d_length-1):
    if d[i_d-1] == 1:
      t.append(d_length - i_d)
  f = np.zeros(d_length, dtype=int)

  for i_d in range(1, d_length+1):
    f[d_length - i_d] = d[i_d-1]

  n = len(s)
  at = np.zeros((M_length, d_length-1))
  at[0, :] = s
  m = len(t)
  b = np.zeros(m, dtype=int)

  for k in range(1, pow(2, n) - 1):
    b[0] = s[t[0]-1] ^ s[t[1]-1]
    if m > 2:
      for i in range(1, m-2):
        b[i] = s[t[i+1]] ^ b[i-1]
```

```
    for j in range(1, n):
      s[n - j] = s[n - j - 1]
      s[0] = b[m-2]
      at[k, :] = s

  ss = np.zeros((M_length, M_length))
  for i in range(1, M_length+1):
    s1 = at[i-1, :]
    seq = LFSR(s1, t, M_length,sub_dgree)
    ss[i-1, :] = seq
  d = ss.transpose()

  k = d[0:sub_dgree, :]
  e = np.zeros((sub_dgree, sub_dgree))

  for i in range(1, sub_dgree+1):
  for j in range(1, sub_dgree+1):
    if i == j:
      e[i-1, j-1] = 1

  for i in range(1, sub_dgree):
    j = i+1
    l = sub_dgree + 2-j
    e[j:sub_dgree, i-1] = f[2:l]

  g = np.zeros((sub_dgree, M_length))
  for i in range(1, M_length+1):
    g[:, i-1] = e.dot(k[:, i-1])

  h = np.zeros((sub_dgree, M_length))
  for i in range(1, sub_dgree+1):
    for j in range(1, M_length+1):
      h[i-1, j-1] = g[i-1, j-1] % 2
  p = h.transpose()
  return ss
######################################################################
def PNLG(ss,ip,m,T):
  s = ss.transpose()
  P = np.zeros((m, T))
  for i in range(1, m + 2):
    P[:, i] = s[:, ip[i-1]]
  U = P.transpose()
  PN = np.reshape(U, m*T, order='F')
  print(PN)
  np.savetxt("foo.csv", PN, delimiter=",")
```

```
  return PN
##############################################################################
def Itp_cal(GF, gm):
  n = len(GF) - 1
  m = len(gm) - 1
  M_length = pow(2, m) - 1
  T = (pow(2, n) - 1) / (M_length)
  alpha = np.zeros((M_length - 1) * len(GF)).reshape(M_length - 1, len(GF))
  for i in range(1, M_length):
    a = np.array(np.concatenate((1, np.zeros(int(T) * (i), dtype=int)), axis=None))
    alpha[i - 1, :] = vet_cal(a, GF)  # bo hang dau tien trong ma tran

  Tralpha = np.zeros((M_length + 1) * len(GF)).reshape(M_length + 1, len(GF))
  for i in range(1, int(T)):
    b = np.array(np.concatenate((1, np.zeros(int(T - 1) * i, dtype=int)), axis=None))
    b[int(T - 1) * i - i] = 1
    Tralpha[i - 1, :] = vet_cal(b, GF)

  Itp = np.zeros(int(T - 1), dtype=int)
  for i in range(1, int(T)):
    for j in range(1, M_length):
      a = np.array_equal(alpha[j - 1, :], Tralpha[i - 1, :])
      if (a):
        Itp[i - 1] = j
  return Itp
##############################################################################
GF = np.array([1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 1])
gm = np.array([1, 1, 0, 0, 0, 0, 1])
s_initial = np.array([0, 0, 0, 0, 0, 1])
n = len(GF) - 1
sub_dgree = len(gm) - 1
M_length = pow(2, sub_dgree) - 1
T = (pow(2, n) - 1) / (M_length)

Itp = Itp_cal(GF, gm)
ss = phase(s_initial, gm, M_length, sub_dgree)
PN1 = PNLG(ss, Itp, M_length, int(T))
PN2 = np.roll(PN1, 0)
corr, _ = pearsonr(PN1, PN2)
print('Pearsons correlation: %.3f' % corr)
```

Phụ lục 2: Chương trình tạo ma trận lấy mẫu

```
def Gauss_matrix(M, N):
        A = []
        for i in range(N):
                col = np.random.randn(M)
                col /= norm2(col)
                A.append(col)
        A = np.array(A).T
        return A

def Bernoulli_matrix(M, N):
        B = []
        for i in range (M * N):
                A = np.random.binomial(size = M * N, n = 1, p = 0.5)
                B.append(A)
        B = np.array(B)

def BPNSM_matrix(M, N):
        A1 = []
        for i in range (len(PN1)):
                for t in range (len(PN1)):
                        if PN1[t] == 0:
                                PN1[t] = -1
                PN_1 = np.roll(PN1, i)
                A1.append(PN_1)
        A2 = []
        for i in range (len(PN2)):
                for t in range (len(PN2)):
                        if PN2[t] == 0:
                                PN2[t] = -1
                PN_2 = np.roll(PN2, i)
                A2.append(PN_2)
        A = A1 + A2
        return A
```

Phụ lục 3: Thuật toán khôi phục

Thuật toán khôi phục Matching Pursuit

```
def MP( y: np.ndarray,
                D: np.ndarray,
                eps_min: np.float64 = 1e-3,
                iter_max: int = 1000) -> np.ndarray:
        x_pre = np.zeros(D.shape[0], dtype=np.float64)
```

```
        r = y.astype(np.float64)
        for _ in range(iter_max):
                dot = np.dot(D, r)
                i = np.argmax(np.abs(dot))
                x_pre[i] = dot[i]
                r -= x_pre[i] * D[i, :]
                if np.linalg.norm(r) < eps_min:
                        break
        return x_pre
```

Thuật toán khôi phục DRMP

```
def DRMP(D, y, N, stop=np.infty, r_thresh=0.01):
# Input: D: Ma tran lay mau
# Vector y:
        r = y
        x_pre = np.zeros(N)
        Lambdas = []
        t = 0
        # Thiet lap dieu kien dung
        while norm2(r) > r_thresh and i < stop:
        # Gan:
                u = D.T.dot(r)
        # Buoc nhan dinh
                Lambda = np.argmax(abs(u))
                Lambdas.append(Lambda)
        # Tat ca cac phan tu duoc chon tu he co so
                basis = D[:, Lambdas]
        # Buoc uoc luong
                x_pre[Lambdas] = np.linalg.inv(np. \
                dot(basis.T, basis)).dot(basis.T).dot(y)
        # Buoc loai bo:
                r = y - D.dot(x_pre)
                t += 1
        return x_pre.T, Lambdas
```

Phụ lục 4: Thực nghiệm và đánh giá

```
def run_signalRF_noise(self):
```

```
def run_image_noise(self):
        img = plt.imread('images/lena_256.jpg')/255
        img = rgb2gray(img)
        img = resize(img, self.imsize)
        # Chuyen tin hieu 2-D thanh 1-D
        dct = self.dct32x32(img)
        zigzag_vector = self.dct2zigzag(dct)
        zigzag_shape = zigzag_vector.shape
        # DRMP with dct zigzag vectors
        N = 16
        for M in self.M:
                pre_zigzag_vector = np.zeros(zigzag_shape)
                D = define_A(M, N)
                for i in range(zigzag_shape[0]):
                        x = zigzag_vector[i].T
                        n = np.random.normal(0, self.sigma, M).T  # Add noise
                        y = D.dot(x) + n
                        x_pre, Lambdas = DRMP(D, y, N, r_thresh=norm2(n))
                        pre_zigzag_vector[i] = x_pre.T
                pre_dct = self.zigzag2dct(pre_zigzag_vector, self.imsize)
                pre_img = self.idct32x32(pre_dct)
                norm_e =Normalized_Error(img, pre_img)
                self.norm_error.append(norm_e)
                psnr = PSNR(img, pre_img)
                mse = MSE(img, pre_img)
```