

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG

----- ☰ ❁ -----



ĐỒ ÁN TỐT NGHIỆP

Đề tài:

Giảng viên hướng dẫn: PGS. TS Đỗ Hồng Tuấn

Sinh viên thực hiện: Lê Đức Minh Nhật

MSSV: 2013994

TP. Hồ Chí Minh, tháng 5 năm 2024

MỤC LỤC

MỤC LỤC	2
DANH MỤC HÌNH ẢNH	4
MỞ ĐẦU.....	7
CHƯƠNG I: CƠ SỞ LÝ THUYẾT VỀ “BỘ LỌC TỐI THIỂU PHƯƠNG SAI CÓ RÀNG BUỘC TUYẾN TÍNH” VÀ “BỘ TRIỆT CÁC BÚP BÊN TỔNG QUÁT” CHO DÃY AN-TEN MỘT CHIỀU, MỞ RỘNG VÀ CẢI TIẾN “BỘ TRIỆT CÁC BÚP BÊN TỔNG QUÁT” VỚI MẠNG NEURAL CHO DÃY AN-TEN HAI CHIỀU	8
I.1. Bài toán thực tế:	8
I.2. Cơ sở lý thuyết về các giải thuật được đưa ra:	10
I.2.1. Bộ lọc tối thiểu phương sai có ràng buộc tuyến tính (Linearly Constrained Minimum-Variance Filter):	10
I.2.2. Bộ triệt các búp bên tổng quát (Generalized Sidelobes Cancellers):	14
I.2.3. Mở rộng “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều và cải tiến với mạng Neural:.....	18
CHƯƠNG II: MÔ PHỎNG GIẢI THUẬT TRÊN MATLAB.....	26
II.1. Mô phỏng giải thuật “Bộ lọc tối thiểu phương sai có ràng buộc tuyến tính” và “Bộ triệt các búp bên tổng quát” với dãy an-ten một chiều:	26
II.1.1. Tối thiểu phương sai có ràng buộc tuyến tính (LCMV):.....	26
II.1.2. Bộ triệt các búp bên tổng quát (GSC):	32
II.1.3. Nhận xét chung về hai giải thuật:	37
II.2. Mở rộng “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều và cải tiến với mạng Neural:	37
II.2.1. Mở rộng “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều:	37
II.2.2. Cải tiến “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều với mạng Neural:.....	42
II.2.3. Nhận xét chung về giải thuật “Bộ triệt các búp bên tổng quát” được mở rộng và cải tiến bởi mạng Neural:	48
CHƯƠNG III: KẾT QUẢ TỔNG QUÁT CỦA HAI GIẢI THUẬT	49
III.1. Tối thiểu phương sai có ràng buộc tuyến tính (LCMV):	49
III.1.1. Thay đổi số phần tử an-ten M:	49
III.1.2. Giảm khoảng cách góc giữa các tín hiệu mong muốn và can nhiễu:	53
III.1.3. Thay đổi phương sai nhiễu (noise variance):	58

III.2. Bộ triệt các búp bên tổng quát (GSC):	62
III.2.1. Thay đổi số phần tử an-ten M:	62
III.2.2. Giảm khoảng cách góc giữa các tín hiệu mong muốn và can nhiễu:	67
III.2.3. Thay đổi phương sai nhiễu (noise variance):	72
III.3. Tổng quát về hai giải thuật và phần mở rộng, cải tiến:	76
III.3.1. Thực hiện so sánh giữa 2 giải thuật LCMV và GSC dựa trên bài toán đã nêu trên:	76
III.3.2. Thực hiện so sánh “Bộ triệt các búp bên tổng quát” được cải tiến với mạng Neural và truyền thông theo 10 phương sai nhiễu bất kỳ (nằm trong khoảng $0 \div 1$):	79
III.3.3. Kết luận chung về các giải thuật:	81
KẾT LUẬN	83
TÀI LIỆU THAM KHẢO	84
PHỤ LỤC	86

DANH MỤC HÌNH ẢNH

Hình 1. Sơ đồ khói biểu diễn bài toán lọc thống kê.....	9
Hình 2. Bộ lọc FIR.....	11
Hình 3. Sóng tới dãy an-ten tuyến tính M phần tử.....	11
Hình 4. Sơ đồ khói của bộ triệt các búp bên tổng quát GSC.....	16
Hình 5. Đưa bài toán GSC về bài toán bộ lọc tối ưu tiêu chuẩn.....	16
Hình 6. Biểu diễn tín hiệu đến dãy an-ten 2 chiều trong hệ tọa độ Oxyz.....	19
Hình 7. Biểu diễn các phần tử an-ten dãy 2 chiều trên mặt phẳng Oxy.....	19
Hình 8. Các lớp của mô hình mạng Neural Multi-layer Perceptron.....	21
Hình 9. Cách tính trọng số và “biases” của từng lớp trong mạng.....	22
Hình 10. Sơ đồ khói “Bộ triệt các búp bên tổng quát” được cải tiến với mạng Neural.....	24
Hình 11. Tín hiệu gốc cần xử lý.....	27
Hình 12. Tín hiệu trộn tại ngõ vào của bộ lọc.....	28
Hình 13. Tín hiệu sau khi qua xử lý.....	28
Hình 14. Trọng số tối ưu của giải thuật.....	29
Hình 15. Đồ thị tuyến tính của giải thuật LCMV.....	30
Hình 16. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra.....	31
Hình 17. RMSE của giải thuật LCMV.....	31
Hình 18. Tín hiệu gốc muốn thu.....	32
Hình 19. Tín hiệu trộn tại ngõ vào bộ lọc.....	33
Hình 20. Tín hiệu sau khi xử lý tại ngõ ra.....	33
Hình 21. Trọng số tối ưu của giải thuật.....	34
Hình 22. Đồ thị tuyến tính của giải thuật GSC.....	35
Hình 23. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra.....	36
Hình 24. RMSE của giải thuật.....	36
Hình 25. Tín hiệu tại ngõ ra của bộ lọc cho dãy an-ten 2 chiều.....	38
Hình 26. Trọng số tối ưu của bộ lọc cho dãy an-ten 2 chiều.....	38
Hình 27. Đồ thị tuyến tính của “Bộ triệt các búp bên tổng quát” cho dãy an-ten 2 chiều.....	39
Hình 28. Biểu diễn đồ thị tuyến tính dưới dạng các đường đồng mức.....	40
Hình 29. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra.....	41
Hình 30. RMSE của giải thuật.....	41
Hình 31. Tín hiệu gốc cần thu.....	42
Hình 32. Tín hiệu trộn tại ngõ vào của bộ lọc.....	43
Hình 33. Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập huấn luyện.....	43
Hình 34. Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập huấn luyện.....	44
Hình 35. Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập kiểm tra.....	44
Hình 36. Khảo sát hệ số học tối ưu cho mạng Neural.....	45
Hình 37. Khảo sát hệ số học tối ưu cho mạng Neural.....	45
Hình 38. Huấn luyện và kiểm tra mô hình trên tập dữ liệu huấn luyện và kiểm tra đã được xây dựng.....	46
Hình 39. Tín hiệu tại ngõ ra của bộ lọc được cải tiến với mạng Neural.....	47
Hình 40. So sánh giữa tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra của bộ lọc.....	47
Hình 41. Đồ thị tuyến tính của giải thuật với $M = 10$	49
Hình 42. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 10$	50
Hình 43. Đồ thị tuyến tính của giải thuật LCMV với $M = 15$	50

Hình 44. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 15$	51
Hình 45. Đồ thị tuyến tính của giải thuật với $M = 20$	52
Hình 46. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 20$	52
Hình 47. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi M	53
Hình 48. Đồ thị tuyến tính của $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$	54
Hình 49. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$	54
Hình 50. Đồ thị tuyến tính của $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$	55
Hình 51. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$	55
Hình 52. Đồ thị tuyến tính của $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$	56
Hình 53. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$	57
Hình 54. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi khoảng cách góc	57
Hình 55. Đồ thị tuyến tính của giải thuật với $nv = 0.01$	58
Hình 56. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.01$	59
Hình 57. Đồ thị tuyến tính của giải thuật với $nv = 0.1$	59
Hình 58. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.1$	60
Hình 59. Đồ thị tuyến tính của giải thuật với $nv = 1$	61
Hình 60. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 1$	61
Hình 61. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi phương sai nhiễu	62
Hình 62. Đồ thị tuyến tính của giải thuật với $M = 10$	63
Hình 63. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 10$	63
Hình 64. Đồ thị tuyến tính của giải thuật với $M = 15$	64
Hình 65. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 15$	65
Hình 66. Đồ thị tuyến tính của giải thuật với $M = 20$	65
Hình 67. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 20$	66
Hình 68. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi số phần tử an-ten M	67
Hình 69. Đồ thị tuyến tính của giải thuật với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$	68
Hình 70. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$	68
Hình 71. Đồ thị tuyến tính của giải thuật với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$	69
Hình 72. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$	69
Hình 73. Đồ thị tuyến tính của giải thuật với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$	70
Hình 74. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$	71
Hình 75. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi khoảng cách góc giữa các tín hiệu	71
Hình 76. Đồ thị tuyến tính của giải thuật với $nv = 0.01$	72
Hình 77. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.01$	73

Hình 78. Đồ thị tuyến tính của giải thuật với $nv = 0.1$	74
Hình 79. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.1$	74
Hình 80. Đồ thị tuyến tính của giải thuật với $nv = 1$	75
Hình 81. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 1$	75
Hình 82. RMSE giữa tín hiệu gốc và tín hiệu trộn sau khi xử lý theo sự thay đổi phương sai nhiều... Hình 83. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý của GSC và LCMV theo sự thay đổi số phần tử M.....	76
Hình 84. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý của GSC và LCMV theo sự thay đổi phương sai nhiều.....	77
Hình 85. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý của GSC và LCMV theo sự thay đổi khoảng cách gốc giữa các tín hiệu ngõ vào.....	78
Hình 86. RMSE (tín hiệu gốc và tín hiệu tại ngõ ra GSC truyền thống) theo sự thay đổi phương sai nhiều.....	79
Hình 87. RMSE (tín hiệu gốc và tín hiệu tại ngõ ra GSC được cải tiến với mạng Neural) theo sự thay đổi phương sai nhiều.....	80
Hình 88. RMSE (tín hiệu gốc và tín hiệu tại ngõ ra GSC truyền thống, GSC được cải tiến với mạng Neural) theo sự thay đổi phương sai nhiều.....	80
Hình 89. RMSE (ngõ ra mong muốn và ngõ ra qua mạng Neural) theo sự thay đổi phương sai nhiều.	
.....	81

MỞ ĐẦU

Hiện nay, thế giới đã phát triển đến mạng 5G, thế hệ thứ năm của công nghệ di động không dây, cung cấp tốc độ tải lên và tải xuống nhanh hơn, kết nối ổn định hơn và dung lượng được cải thiện so với các mạng trước đây. Sự phát triển của hệ thống truyền thông đi kèm với các yêu cầu về chất lượng tín hiệu ngày một cải thiện. Do đó, vấn đề triệt tiêu các can nhiễu, tối ưu tín hiệu từ hướng tới mong muốn luôn là một trong những ưu tiên cho mọi thế hệ mạng. Kể từ khi tín hiệu được kỹ thuật số hóa, việc xử lý tín hiệu dựa trên thuật toán ngày càng phổ biến và rộng rãi, rất nhiều giải thuật được đưa ra nhằm đáp ứng các yêu cầu về mặt kỹ thuật cũng như yếu tố môi trường. Bên cạnh đó, với sự phát triển của công nghệ Trí tuệ nhân tạo/Học máy (AI/Machine Learning), khả năng tính toán với độ phức tạp cao ngày càng trở nên dễ dàng. Tuy nhiên vẫn rất khó để tìm ra các giải thuật và tính toán tối ưu để tổng quát cho mọi trường hợp.

Nhận thấy tính cấp thiết của vấn đề này, dưới sự hướng dẫn của GVHD, em đã lựa chọn nghiên cứu đề tài « », được chia thành 3 chương như sau :

Chương 1: Cơ sở lý thuyết về “Bộ lọc tối thiểu phuơng sai có ràng buộc tuyến tính” và “Bộ triệt các búp bên tổng quát” cho dãy an-ten một chiều, mở rộng và cải tiến “Bộ triệt các búp bên tổng quát” với mạng Neural cho dãy an-ten hai chiều.

Chương 2: Mô phỏng các giải thuật trên MATLAB.

Chương 3: Kết quả tổng quát của các giải thuật.

CHƯƠNG I: CƠ SỞ LÝ THUYẾT VỀ “BỘ LỌC TỐI THIỂU PHƯƠNG SAI CÓ RÀNG BUỘC TUYẾN TÍNH” VÀ “BỘ TRIỆT CÁC BÚP BÊN TỔNG QUÁT” CHO DÃY AN-TEN MỘT CHIỀU, MỎ RỘNG VÀ CẢI TIẾN “BỘ TRIỆT CÁC BÚP BÊN TỔNG QUÁT” VỚI MẠNG NEURAL CHO DÃY AN-TEN HAI CHIỀU

I.1. Bài toán thực tế:

“Interference cancellation” (triệt can nhiễu) là một kỹ thuật trong hệ thống truyền thông không dây nhằm giảm thiểu hiệu ứng của can nhiễu và cải thiện chất lượng tín hiệu. Trong hệ thống truyền thông không dây, can nhiễu có thể xuất phát từ nhiều nguồn, bao gồm các thiết bị không dây khác hoạt động trong cùng một khu vực và các tín hiệu phản xạ từ môi trường. “Interference cancellation” là một phương pháp xử lý tín hiệu tại bộ thu để loại bỏ hoặc giảm thiểu tác động của can nhiễu. Kỹ thuật này có thể sử dụng một số phương pháp khác nhau, tùy thuộc vào bối cảnh ứng dụng cụ thể cũng như các yêu cầu về mặt kỹ thuật.

Có rất nhiều phương pháp được sử dụng để thực hiện “Interference cancellation” như sử dụng các an-ten đa điểm (MIMO) và kỹ thuật xử lý tín hiệu đa đường (multi-path). Bằng cách sử dụng nhiều an-ten và khai thác tính chất truyền dẫn đa đường, hệ thống có thể phân tách và loại bỏ can nhiễu từ các nguồn khác nhau trong môi trường. Ngoài ra, các phương pháp kết hợp như kỹ thuật mã hóa thông tin (coding) và kỹ thuật đa truy cập (multiple access) cũng có thể được sử dụng để giảm thiểu can nhiễu và tăng cường khả năng chống can nhiễu của hệ thống.

Một trong những phương pháp phổ biến để thực hiện “Interference cancellation” là sử dụng kỹ thuật xử lý tín hiệu số (DSP). Kỹ thuật này sử dụng thông tin về can nhiễu và tín hiệu mong muốn để xác định và loại bỏ can nhiễu từ tín hiệu thu. Các thuật toán xử lý tín hiệu phức tạp có thể được sử dụng để ước lượng và loại bỏ can nhiễu từ tín hiệu thu, đồng thời giữ lại tín hiệu mong muốn.

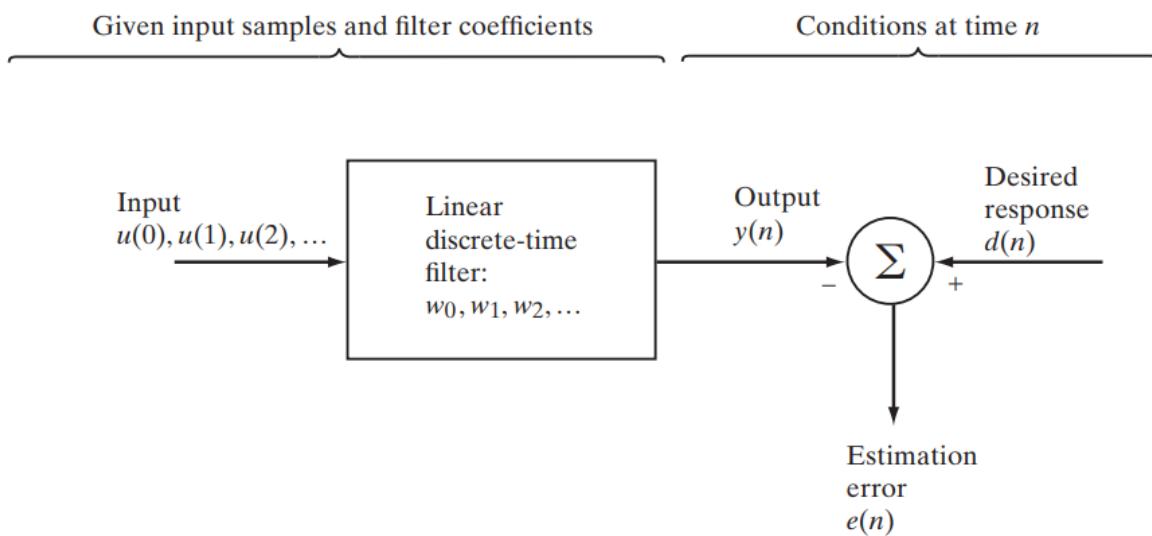
Bài toán về các “bộ lọc thích nghi” là một phần quan trọng của việc xử lý tín hiệu số. Sử dụng “bộ lọc thích nghi” trong môi trường có số liệu thống kê không xác định hoặc môi trường không cố định mang lại giải pháp tối ưu với việc cải thiện đáng kể về hiệu suất so với sử dụng bộ lọc cố định được thiết kế truyền thống. Không những vậy, các bộ lọc thích nghi cung cấp khả năng xử lý tín hiệu mới mà các phương pháp khác không thể thực hiện được. Do đó, các bộ lọc thích nghi đã mang lại nhiều thành công trong các lĩnh vực đa dạng như thông tin liên lạc, điều khiển, radar, sóng siêu âm, địa chấn và kỹ thuật y sinh, cùng nhiều lĩnh vực khác.

Mục tiêu đặt ra là phát triển các giải thuật khác nhau của “bộ lọc thích nghi tuyến tính”. Sự thích nghi được thực hiện bằng cách điều chỉnh các tham số (hệ số) tự do của bộ lọc

phù hợp với dữ liệu ngõ vào, qua đó triệt các ảnh hưởng của can nhiễu và chỉ xuất hiện tín hiệu mong muốn tại ngõ ra của bộ lọc.

Đò án thực hiện nghiên cứu giải thuật ứng dụng cho bộ lọc tối ưu tuyến tính với thời gian rời rạc, hay còn gọi là bộ lọc Wiener. Lý thuyết bộ lọc Wiener được xây dựng cho trường hợp tổng quát của quá trình ngẫu nhiên có giá trị phức với bộ lọc được xác định theo đáp ứng xung của nó. Lý do sử dụng chuỗi thời gian có giá trị phức là trong nhiều tình huống thực tế (ví dụ: thông tin liên lạc, radar, sóng siêu âm, ...), các dữ liệu quan sát được đo ở dạng băng cơ sở (dải tần đại diện cho tín hiệu gốc do nguồn thông tin cung cấp), còn trường hợp chuỗi thời gian có giá trị thực có thể được coi là trường hợp đặc biệt của lý thuyết này. Tuy nhiên, với trọng tâm là nghiên cứu hai giải thuật chính, đò án sẽ công nhận và sử dụng các kết quả có được từ bộ lọc Wiener.

Xét sơ đồ khối của Hình 1 minh họa bộ lọc tuyến tính với thời gian rời rạc. Ngõ vào của bộ lọc gồm chuỗi thời gian $u(0), u(1), u(2), \dots$ và bộ lọc đặc trưng bởi đáp ứng xung được biểu thị bằng chuỗi w_0, w_1, w_2, \dots . Ngõ ra của bộ lọc tại thời điểm rời rạc n là $y(n)$, được sử dụng để cung cấp ước tính về đáp ứng tín hiệu mong muốn $d(n)$. Với ngõ vào bộ lọc và đáp ứng mong muốn tại từng thời điểm rời rạc của các quá trình ngẫu nhiên tương ứng, việc ước tính thường đi kèm với sai số có các đặc tính thống kê riêng của nó. Mục tiêu đặt ra là tối thiểu lỗi ước tính $e(n)$, được định nghĩa là sự khác biệt giữa đáp ứng tín hiệu mong muốn $d(n)$ và ngõ ra bộ lọc $y(n)$. Ngoài ra, để thuận tiện cho quá trình nghiên cứu, giả sử bộ lọc tối ưu tuyến tính là bộ lọc FIR với đặc tính ổn định cho các tính toán thích nghi được thực hiện.



Hình 1. Sơ đồ khái niệm bài toán lọc thống kê.

Tóm lại, bài toán đặt ra như sau: Thiết kế bộ lọc tuyến tính với thời gian rời rạc với ngõ ra $y(n)$ có ước lượng tín hiệu mong muốn $d(n)$, với tập hợp các mẫu ngõ vào $u(0), u(1), u(2), \dots$ sao cho giá trị bình phương trung bình của lỗi ước tính $e(n)$, được định nghĩa là sự khác biệt giữa đáp ứng tín hiệu mong muốn $d(n)$ và ngõ ra thực tế $y(n)$, là tối thiểu.

Hai trong những giải thuật được đưa ra cho bài toán này chính là “Bộ lọc tối thiểu phương sai có ràng buộc tuyến tính” (Linearly Constrained Minimum-Variance Filter) và “Bộ triệt các búp bên tổng quát” (Generalized Sidelobes Cancellers).[1].

I.2. Cơ sở lý thuyết về các giải thuật được đưa ra:

I.2.1. Bộ lọc tối thiểu phương sai có ràng buộc tuyến tính (Linearly Constrained Minimum-Variance Filter):

Mục tiêu của bộ lọc Wiener là giảm thiểu giá trị bình phương trung bình của lỗi ước tính, được định nghĩa là sự khác biệt giữa đáp ứng mong muốn và ngõ ra bộ lọc thực tế. Khi giải quyết vấn đề tối ưu hóa (tối thiểu hóa) này, không có ràng buộc nào được áp đặt cho giải pháp.

Tuy nhiên, trong một số ứng dụng lọc, có thể mong muốn (hoặc thậm chí bắt buộc) thiết kế bộ lọc giảm thiểu tiêu chí bình phương trung bình theo một ràng buộc cụ thể. Ví dụ, yêu cầu có thể là giảm thiểu công suất đầu ra trung bình của bộ lọc tuyến tính trong khi đáp ứng của bộ lọc được đo ở một số tần số quan tâm cụ thể bị hạn chế duy trì không đổi. Trong phần này, xem xét một giải pháp như vậy cho hai trường hợp khác nhau:

Trường hợp 1: Xử lý tín hiệu trong miền thời gian.

Xét bộ lọc FIR như Hình 2 có ngõ ra là các đáp ứng cho ngõ vào $u(n)$, $u(n-1), \dots, u(n-M+1)$ được cho bởi:

$$y(n) = \sum_{k=0}^{M-1} w_k^* u(n-k) \quad (1.1)$$

Với trường hợp đặc biệt các ngõ vào là tín hiệu sin:

$$u(n) = e^{j\omega n} \quad (1.2)$$

Có thể viết lại:

$$y(n) = e^{j\omega n} \sum_{k=0}^{M-1} w_k^* e^{-j\omega k} \quad (1.3)$$

Với ω là tần số góc của tín hiệu sin, được chuẩn hóa theo tần số lấy mẫu; biểu thức tổng là đáp ứng tần số của bộ lọc.

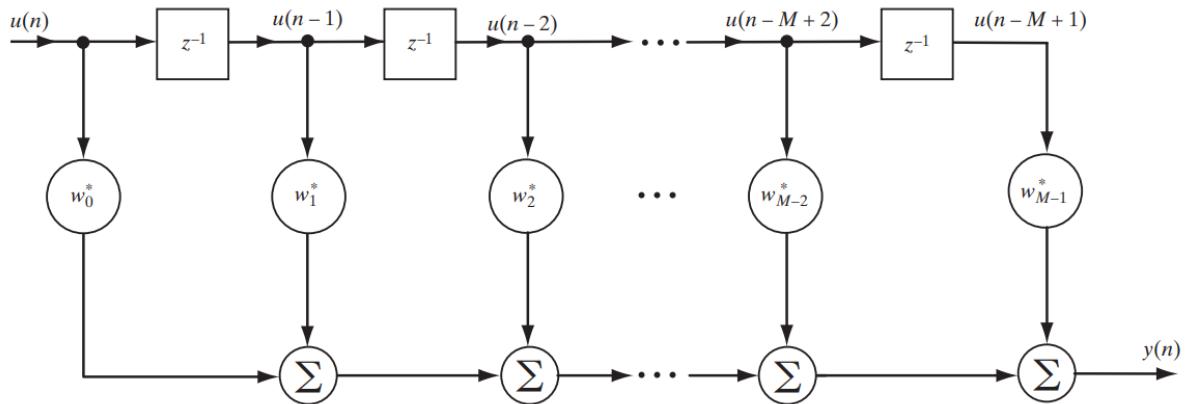
Vấn đề tối ưu hóa có ràng buộc có thể được phát biểu như sau: Tìm tập hợp các hệ số tối ưu của bộ lọc $w_{00}, w_{01}, w_{02}, \dots, w_{0M-1}$ để cực tiểu hóa giá trị bình phương trung bình của ngõ ra bộ lọc $y(n)$, tuân theo ràng buộc tuyến tính:

$$\sum_{k=0}^{M-1} w_k^* e^{-j\omega_0 k} = g \quad (1.4)$$

Trong đó ω_0 là giá trị cho trước của tần số góc chuẩn hóa ω , nằm trong khoảng:

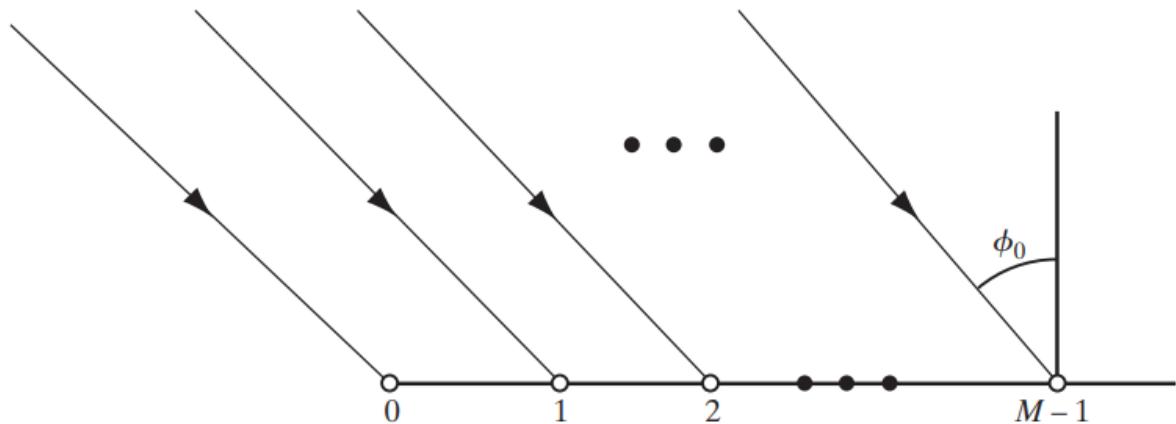
$$-\pi \leq \omega \leq \pi \quad (1.5)$$

Và g là độ lợi có giá trị phức.



Hình 2. Bộ lọc FIR.

Trường hợp 2: Xử lý tín hiệu trong miền không gian.



Hình 3. Sóng tới dãy an-ten tuyển tính M phần tử.

Vấn đề bộ lọc tối ưu hóa có ràng buộc được mô tả bởi các phương trình (1.1) và (1.4) có tính chất thời gian. Có thể xây dựng phiên bản không gian của bài toán bằng cách xem xét “bộ tạo búp sóng (beamformer)” được mô tả trong hình 3, bao gồm một dãy tuyển tính gồm các phần tử an-ten có khoảng cách đều nhau với trọng số có thể điều chỉnh được (không được hiển thị trong hình). Với một nguồn đang hướng đặt ở “trường xa (far field)” phát ra tín hiệu tới dãy, tại thời điểm n , một sóng phản tác động len dãy dọc theo hướng xác định góc ϕ_0 so với phương vuông góc với mảng. Ngoài ra, giả định rằng khoảng cách giữa các phần tử của mảng nhỏ hơn $\lambda/2$, trong đó λ là bước sóng của tín hiệu truyền đi, nhằm thỏa tính chất tương tự không gian của định lý lấy mẫu. Kết quả ngõ ra của bộ tạo búp sóng được cho bởi:

$$y(n) = u_0(n) \sum_{k=0}^{M-1} w_k^* e^{-jk\theta_0} \quad (1.6)$$

Với hướng tới đặc trưng bởi góc điện θ_0 , tương quan với góc tới thực tế φ_0 ; được xác định như sau:

$$\theta_0 = 2\pi \frac{\Delta}{\lambda} \sin \varphi_0 \quad (1.7)$$

$u_0(n)$ là tín hiệu thu được tại an-ten 0 trong hình 3, được xem là tín hiệu tham chiếu; w_k là trọng số của bộ tạo búp sóng. Bây giờ bài toán được phát biểu như sau: Tìm tập hợp các hệ số tối ưu của bộ lọc $w_{o0}, w_{o1}, w_{o2}, \dots, w_{oM-1}$ để cực tiểu hóa giá trị bình phương trung bình của ngõ ra bộ lọc $y(n)$, tuân theo ràng buộc tuyến tính:

$$\sum_{k=0}^{M-1} w_k^* e^{-jk\theta_0} = g \quad (1.8)$$

Trong đó θ_0 là giá trị cho trước của tần số góc điện θ , nằm trong khoảng:

$$-\pi \leq \theta \leq \pi \quad (1.9)$$

Và g là độ lợi có giá trị phức. Bộ tạo búp sóng có băng tần hẹp, nghĩa là đáp ứng của nó chỉ cần được hạn chế ở một tần số duy nhất.

So sánh xử lý tín hiệu trong miền thời gian của Hình 2 và không gian của Hình 3, mặc dù giải quyết các tình huống vật lý hoàn toàn khác nhau, nhưng các công thức là tương đương về mặt toán học. Thật vậy, trong cả hai trường hợp, đều cần phải giải quyết bài toán tối ưu hóa có ràng buộc giống nhau. Sử dụng phương pháp nhân tử Lagrange (methods of Lagrange multipliers) để giải quyết bài toán tối ưu này, bắt đầu với xác định giá trị thực của “hàm chi phí (cost function)” J bao gồm 2 phần của vấn đề tối ưu tuyến tính có ràng buộc. Cụ thể:

$$J = \sum_{k=0}^{M-1} \sum_{i=0}^{M-1} w_k^* w_i r(i - k) + Re \left[\lambda * \left(\sum_{k=0}^{M-1} w_k^* e^{-jk\theta_0} - g \right) \right] \quad (1.10)$$

Với $\sum_{k=0}^{M-1} \sum_{i=0}^{M-1} w_k^* w_i r(i - k)$ là công suất ngõ ra và $Re[\lambda(\sum_{k=0}^{M-1} w_k^* e^{-jk\theta_0} - g)]$ là ràng buộc tuyến tính. λ là nhân tử Lagrange phức, và lưu ý rằng không có đáp ứng mong muốn trong biểu thức hàm chi phí J ; đúng hơn, nó bao gồm một ràng buộc tuyến tính phải được thỏa mãn đối với góc điện quy định θ_0 đối với bộ tạo búp sóng, hoặc, tương đương, tần số góc ω_0 trong lọc FIR. Trong mọi trường hợp, việc áp đặt ràng buộc tuyến tính sẽ bảo toàn tín hiệu quan tâm và việc tối thiểu hóa hàm chi phí J sẽ làm giảm nhiễu hoặc can nhiễu.

Giải pháp tối ưu cho bài toán tạo búp sóng (beamforming):

Cần tìm được các giá trị tối ưu của các phần tử trọng số của bộ tạo búp sóng để giảm thiểu hàm chi phí J được xác định trong biểu thức (1.10). Để làm như vậy, xác định vector gradient ∇J và cho nó bằng 0. Do đó có thể tìm được phần tử thứ k của vector gradient ∇J :

$$\nabla_k J = 2 \sum_{i=0}^{M-1} w_i r(i-k) + \lambda * e^{-jk\theta_0} \quad (1.11)$$

Gọi w_{oi} là phần tử thứ i của vector trọng số tối ưu \mathbf{w}_o . Điều kiện cho việc tối ưu bộ tạo búp sóng:

$$\sum_{i=0}^{M-1} w_{oi} r(i-k) = -\frac{\lambda^*}{2} e^{-jk\theta_0}, k = 0, 1, \dots, M-1 \quad (1.12)$$

M phương trình đồng thời này xác định các giá trị tối ưu của trọng số của bộ tạo búp sóng. Nó có dạng hơi giống với dạng của phương trình Wiener–Hopf.

Chuyển biểu thức sang dạng ma trận để dễ dàng tính toán:

$$\mathbf{R}\mathbf{w}_o = -\frac{\lambda^*}{2} \mathbf{s}(\theta_0) \quad (1.13)$$

Với \mathbf{R} là ma trận tương quan $M \times M$ và \mathbf{w}_o là vector trọng số tối ưu $M \times 1$. Vector lái $M \times 1$ được định nghĩa:

$$\mathbf{s}(\theta_0) = [1, e^{-j\theta_0}, \dots, e^{-j(M-1)\theta_0}]^T \quad (1.14)$$

Do đó có thể viết lại biểu thức cho \mathbf{w}_o :

$$\mathbf{w}_o = -\frac{\lambda^*}{2} \mathbf{R}^{-1} \mathbf{s}(\theta_0) \quad (1.15)$$

Trong đó, \mathbf{R}^{-1} ma trận nghịch đảo của ma trận \mathbf{R} , giả sử ma trận này không suy biến (nonsingular). Giả sử này hoàn toàn hợp lý trong thực tế vì thực tế là, đối với bộ tạo búp sóng sóng, tín hiệu nhận được ở đầu ra của mỗi phần tử an-ten của hệ thống có thể bao gồm thành phần nhiễu trắng (nhiễu nhiệt) đại diện cho nhiễu cảm biến.

Giải pháp cho vector trọng số tối ưu được đưa ra trong biểu thức (1.15) chưa tối ưu, vì liên quan đến hệ số nhân Lagrange λ chưa biết (hay chính xác là liên hợp phức của nó). Để loại bỏ λ^* , trước tiên sử dụng ràng buộc tuyến tính của biểu thức (1.8) để viết:

$$\mathbf{w}_o^H \mathbf{s}(\theta_0) = g \quad (1.16)$$

Trong đó H là biểu thị cho ma trận chuyển vị Hermitian (chuyển vị với phức liên hợp). Do đó có thể tìm được λ :

$$\lambda = -\frac{2g}{\mathbf{s}^H(\theta_0) \mathbf{R}^{-1} \mathbf{s}(\theta_0)} \quad (1.17)$$

Với việc sử dụng $\mathbf{R}^{-1} = \mathbf{R}^H$. Dạng bậc 2 của $\mathbf{s}^H(\theta_0) \mathbf{R}^{-1} \mathbf{s}(\theta_0)$ là giá trị thực. Do đó, có thể tìm được vector trọng số tối ưu:

$$\mathbf{w}_o = \frac{g^* \mathbf{R}^{-1} \mathbf{s}(\theta_0)}{\mathbf{s}^H(\theta_0) \mathbf{R}^{-1} \mathbf{s}(\theta_0)} \quad (1.18)$$

Lưu ý rằng bằng cách giảm thiểu công suất đầu ra tuân theo ràng buộc tuyến tính của phương trình (1.16), các tín hiệu tới trên dãy theo các hướng khác với θ_0 có xu hướng bị suy giảm.

Bộ tạo búp sóng được đặc trưng bởi vector trọng số \mathbf{w}_o được gọi là bộ tạo búp sóng có phương sai tối thiểu với ràng buộc tuyến tính (Linearly Constrained Minimum-Variance LCMV). Đối với ngõ vào có giá trị trung bình bằng 0 và do đó ngõ ra có giá trị trung bình bằng 0, “phương sai tối thiểu” và “giá trị bình phương trung bình tối thiểu” thực sự đồng nghĩa với nhau. Ngoài ra, như đã trình bày trên, giải pháp được xác định bởi phương trình (1.18) với ω_0 thay thế cho θ_0 có thể được gọi là bộ lọc LCMV. Mặc dù bộ tạo búp sóng LCMV và bộ lọc LCMV khá khác nhau về mặt vật lý nhưng việc tối ưu của chúng là như nhau về mặt toán học.

I.2.2. Bộ triệt các búp bên tổng quát (Generalized Sidelobes Cancellers):

Mở rộng với bộ tạo búp sóng băng hẹp LCMV được xác định bởi ràng buộc tuyến tính của biểu thức (1.8), lưu ý rằng ràng buộc này thể hiện tích vô hướng:

$$\mathbf{w}^H \mathbf{s}(\theta_0) = g \quad (2.1)$$

Với \mathbf{w} là vector trọng số và $\mathbf{s}(\theta_0)$ là vector lái kích thước $M \times 1$ đọc theo góc điện θ_0 , M là số phần tử an-ten. Có thể tổng quát khái niệm ràng buộc tuyến tính bằng cách đưa ra nhiều ràng buộc tuyến tính:

$$\mathbf{C}^H \mathbf{w} = \mathbf{g} \quad (2.2)$$

Ma trận \mathbf{C} là ma trận ràng buộc tuyến tính, và vector độ lợi \mathbf{g} có số phần tử cố định. Giả sử có L ràng buộc tuyến tính, \mathbf{C} là ma trận kích thước $M \times M$ và \mathbf{g} có kích thước $L \times 1$; mỗi cột trong ma trận \mathbf{C} biểu thị cho mỗi ràng buộc tuyến tính. Ngoài ra, giả sử các ràng buộc này độc lập với nhau, ví dụ:

$$[\mathbf{s}(\theta_0), \mathbf{s}(\theta_1)]^H \mathbf{w} = [1 \ 0]^T \quad (2.3)$$

Bộ tạo búp sóng băng hẹp bị ràng buộc để duy trì tín hiệu mong muốn đến dãy đọc theo góc điện θ_0 , đồng thời, triệt can nhiễu biết trước theo góc điện θ_1 .

Đặt các cột của ma trận \mathbf{C}_a kích thước $M \times (M - L)$ được xác định là phần bù trực giao của các cột của ma trận \mathbf{C} , ta có thể định nghĩa ma trận \mathbf{C}_a :

$$\mathbf{C}^H \mathbf{C}_a = \mathbf{0} \quad (2.4)$$

Hoặc:

$$\mathbf{C}_a^H \mathbf{C} = \mathbf{0} \quad (2.5)$$

Ma trận $\mathbf{0}$ ở 2 biểu thức trên có kích thước $L \times (M - L)$ và $(M - L) \times L$, với thực tế $M > L$.

Xác định ma trận phân vùng $M \times M$:

$$\mathbf{U} = [\mathbf{C} : \mathbf{C}_a] \quad (2.6)$$

có các cột trái rộng trên toàn bộ không gian tín hiệu M chiều. Ma trận nghịch đảo \mathbf{U}^{-1} tồn tại nhờ định thức của ma trận \mathbf{U} khác 0.

Do đó vector trọng số \mathbf{w} được xác định theo ma trận \mathbf{U} :

$$\mathbf{w} = \mathbf{U}\mathbf{q} \quad (2.7)$$

Trong đó vector \mathbf{q} kích thước $M \times M$ được xác định bởi:

$$\mathbf{q} = \mathbf{U}^{-1}\mathbf{w} \quad (2.8)$$

Với \mathbf{q} được phân vùng tương tự biểu thức (2.6):

$$\mathbf{q} = [\mathbf{v} : -\mathbf{w}_a]^T \quad (2.9)$$

Trong đó \mathbf{v} là vector kích thước $L \times 1$ và vector \mathbf{w}_a kích thước $(M - L) \times 1$ là phần của vector trọng số \mathbf{w} không bị ảnh hưởng bởi các ràng buộc. Do đó có thể viết lại biểu thức (2.7):

$$\mathbf{w} = [\mathbf{C} : \mathbf{C}_a][\mathbf{v} : -\mathbf{w}_a]^T = \mathbf{C}\mathbf{v} - \mathbf{C}_a\mathbf{w}_a \quad (2.10)$$

Áp dụng các tuyến tính ràng buộc vào biểu thức (2.2) ta có:

$$\mathbf{C}^H\mathbf{C}\mathbf{v} - \mathbf{C}^H\mathbf{C}_a\mathbf{w}_a = \mathbf{g} \quad (2.11)$$

Với việc $\mathbf{C}^H\mathbf{C}_a = 0$ do đó có thể viết lại:

$$\mathbf{C}^H\mathbf{C}\mathbf{v} = \mathbf{g} \quad (2.12)$$

Do đó có thể tìm được \mathbf{v} :

$$\mathbf{v} = (\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g} \quad (2.13)$$

Biểu thức (2.13) cho thấy các ràng buộc tuyến tính không ảnh hưởng đến \mathbf{w}_a . Tiếp theo, xác định thành phần bộ tạo búp sóng cố định:

$$\mathbf{w}_q = \mathbf{C}\mathbf{v} = \mathbf{C}(\mathbf{C}^H\mathbf{C})^{-1}\mathbf{g} \quad (2.14)$$

Vector này trực giao với các cột của ma trận \mathbf{C}_a nhờ các tính chất được mô tả trong biểu thức (2.5).

Từ định nghĩa này, chúng ta có thể sử dụng phương trình (2.10) để biểu thị vector trọng số tổng quát của bộ tạo búp sóng:

$$\mathbf{w} = \mathbf{w}_q - \mathbf{C}_a\mathbf{w}_a \quad (2.15)$$

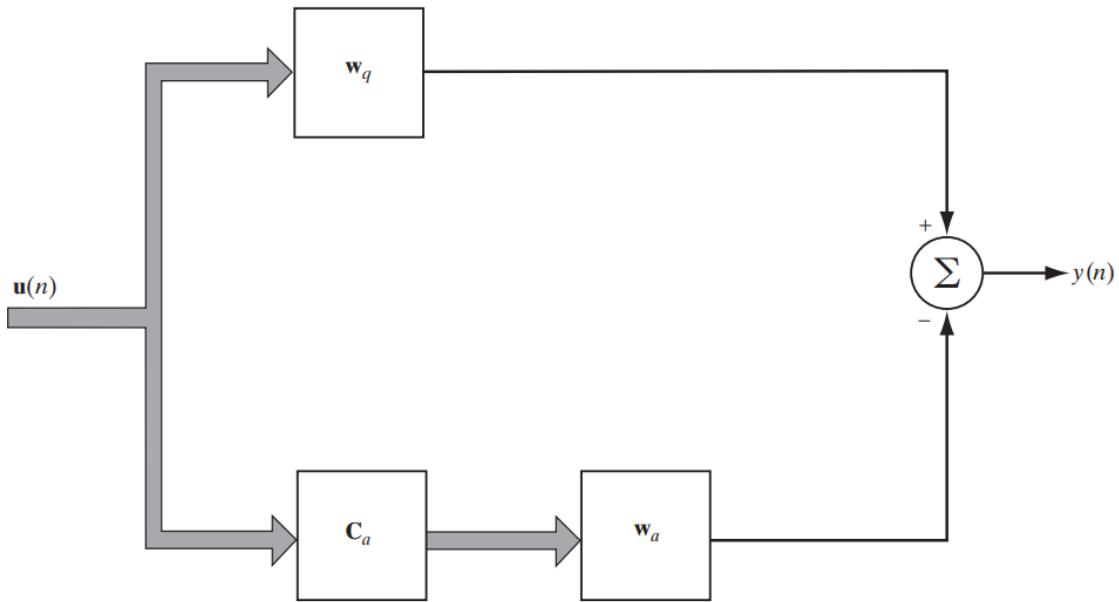
Do đó có thể viết lại biểu thức (2.2):

$$\mathbf{C}^H\mathbf{w}_q - \mathbf{C}^H\mathbf{C}_a\mathbf{w}_a = \mathbf{g} \quad (2.16)$$

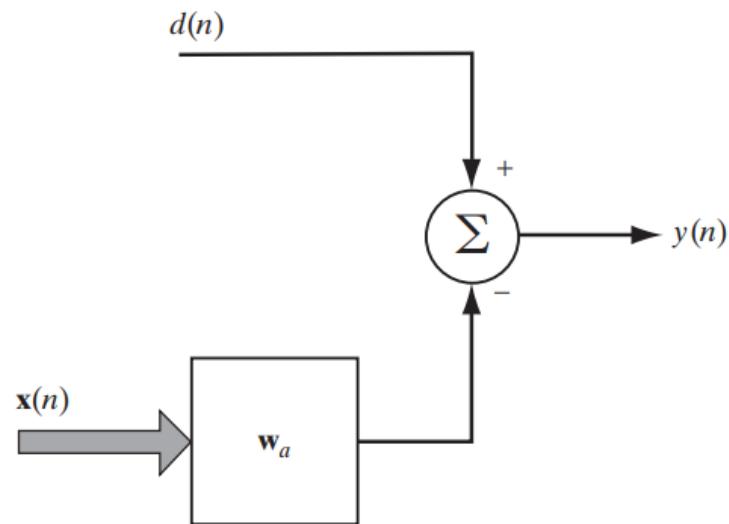
Hay có thể viết lại:

$$\mathbf{C}^H \mathbf{w}_q = \mathbf{g} \quad (2.17)$$

Biểu thức (2.17) cho thấy vector trọng số \mathbf{w}_q là một phần của vector trọng số \mathbf{w} thỏa mãn các ràng buộc. Ngược lại, vector \mathbf{w}_a không bị ảnh hưởng bởi các ràng buộc; do đó nó cung cấp các bậc tự do được xây dựng trong thiết kế của bộ tạo búp sóng. Vì vậy, theo phương trình (2.15), bộ tạo búp sóng có thể được biểu diễn bằng sơ đồ khói như trong Hình 4. Bộ tạo búp sóng tia được mô tả ở đây được gọi là bộ triệt các búp bên tổng quát (Generalized Sidelobes Cancellers - GSC).



Hình 4. Sơ đồ khói của bộ triệt các búp bên tổng quát GSC.



Hình 5. Dựa bài toán GSC về bài toán bộ lọc tối ưu tiêu chuẩn.

Theo biểu thức (2.14), bây giờ có thể thực hiện tối thiểu hóa không ràng buộc giá trị bình phuong trung bình của ngõ ra bộ tạo búp sóng $y(n)$ đối với vector trọng số có thể

điều chỉnh \mathbf{w}_a . Theo phương trình (1.6), ngõ ra của bộ tạo búp sóng được xác định bởi tích vô hướng:

$$y(n) = \mathbf{w}^H \mathbf{u}(n) \quad (2.18)$$

Với vector tín hiệu ngõ vào:

$$\mathbf{u}(n) = u_0(n) [1, e^{-j\theta_0}, \dots, e^{-j(M-1)\theta_0}]^T \quad (2.19)$$

Trong đó góc điện θ_0 được xác định bởi hướng đến của sóng phẳng tới và $u_0(n)$ là tín hiệu điện được thu bởi phần tử an-ten 0 của dãy tuyền tính trong Hình 2.10 tại thời điểm n . Do đó, thay thế phương trình (2.15) vào phương trình (2.18):

$$y(n) = \mathbf{w}_q^H \mathbf{u}(n) - \mathbf{w}_a^H \mathbf{C}_a^H \mathbf{u}(n) \quad (2.20)$$

Do đó nếu xác định:

$$d(n) = \mathbf{w}_q^H \mathbf{u}(n) \quad (2.21)$$

Và:

$$\mathbf{x}(n) = \mathbf{C}_a^H \mathbf{u}(n) \quad (2.22)$$

Có thể viết lại biểu thức dưới dạng tương đồng với bộ lọc Wiener:

$$y(n) = d(n) - \mathbf{w}_a^H \mathbf{x}(n) \quad (2.23)$$

Trong đó, $d(n)$ đóng vai trò như “đáp ứng mong muốn” cho GSC và $\mathbf{x}(n)$ đóng vai trò của vector ngõ vào, như biểu diễn trong Hình 5. Do đó, chúng ta thấy rằng việc sử dụng kết hợp vector \mathbf{w}_q và ma trận \mathbf{C}_a đã chuyển đổi bài toán tối ưu ràng buộc tuyền tính thành bài toán lọc tối ưu tiêu chuẩn. Đặc biệt, bây giờ chúng ta có một vấn đề tối ưu hóa không bị ràng buộc liên quan đến phần \mathbf{w}_a có thể điều chỉnh của vector trọng số, có thể được viết chính thức là

$$\min_{\mathbf{w}_a} E[|y(n)|^2] = \min_{\mathbf{w}_a} (\sigma_d^2 - \mathbf{w}_a^H \mathbf{p}_x - \mathbf{p}_x^H \mathbf{w}_a + \mathbf{w}_a^H \mathbf{R}_x \mathbf{w}_a) \quad (2.24)$$

Với vector tương quan chéo $(M - L) \times 1$:

$$\mathbf{p}_x = E[\mathbf{x}(n) d^*(n)] \quad (2.25)$$

Và ma trận tương quan $(M - L) \times (M - L)$:

$$\mathbf{R}_x = E[\mathbf{x}(n) \mathbf{x}^H(n)] \quad (2.26)$$

Hàm chi phí ở biểu thức là hàm bậc 2 theo vector chưa biết \mathbf{w}_a , mà, như đã nêu trước đây, thể hiện bậc tự do có sẵn trong GSC. Quan trọng nhất, hàm chi phí này có dạng toán học giống hệt như bộ lọc Wiener tiêu chuẩn được xác định. Theo đó, chúng ta có thể dễ dàng sử dụng các kết quả có sẵn của bộ lọc Wiener để đạt được giá trị tối ưu của \mathbf{w}_a :

$$\mathbf{w}_{ao} = \mathbf{R}_x^{-1} \mathbf{p}_x \quad (2.27)$$

Từ biểu thức (2.21) và (2.22) ta có thể biểu diễn lại \mathbf{p}_x :

$$\mathbf{p}_x = E[\mathbf{C}_a^H \mathbf{u}(n) \mathbf{u}^H(n) \mathbf{w}_q] = \mathbf{C}_a^H E[\mathbf{u}(n) \mathbf{u}^H(n)] \mathbf{w}_q = \mathbf{C}_a^H \mathbf{R} \mathbf{w}_q \quad (2.28)$$

Trong đó \mathbf{R} là ma trận tương quan của vector dữ liệu tới $\mathbf{u}(n)$. Tương tự trên ta xác định được ma trận \mathbf{R}_x :

$$\mathbf{R}_x = E[\mathbf{C}_a^H \mathbf{u}(n) \mathbf{u}^H(n) \mathbf{C}_a] = \mathbf{C}_a^H \mathbf{R} \mathbf{C}_a \quad (2.29)$$

Ma trận \mathbf{C}_a có thứ hạng đầy đủ và ma trận tương quan \mathbf{R} xác định dương, do dữ liệu đến luôn chứa một số dạng nhiễu, do đó \mathbf{R}_x không suy biến. Theo đó, chúng ta có thể viết lại nghiệm tối ưu của phương trình (2.27) dưới dạng:

$$\mathbf{w}_{ao} = (\mathbf{C}_a^H \mathbf{R} \mathbf{C}_a)^{-1} \mathbf{C}_a^H \mathbf{R} \mathbf{w}_q \quad (2.30)$$

Xét trường hợp đặc biệt là “môi trường yên tĩnh (quiet environment)”, trong đó tín hiệu thu được chỉ bao gồm nhiễu trắng. Ma trận tương quan \mathbf{R} :

$$\mathbf{R} = \sigma^2 \mathbf{I} \quad (2.31)$$

Với \mathbf{I} là ma trận nhận biết và σ^2 là phương sai nhiễu. Dưới điều kiện này, ta tìm được \mathbf{w}_{ao} như với biểu thức (2.27):

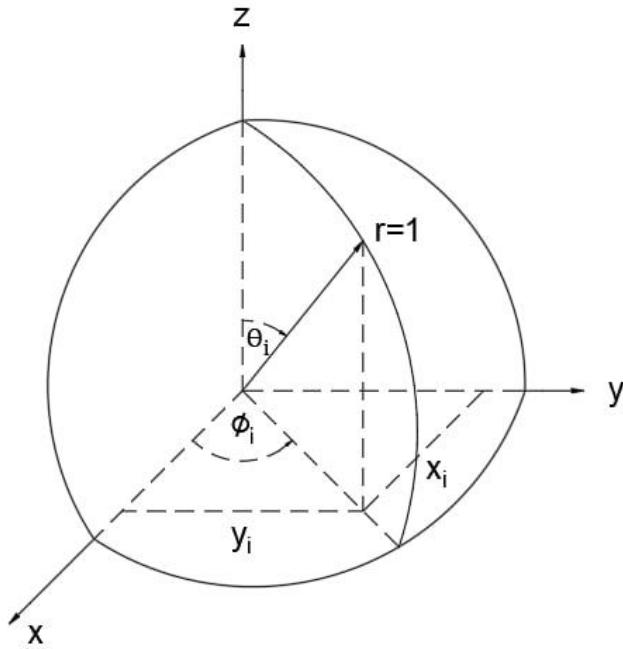
$$\mathbf{w}_{ao} = (\mathbf{C}_a^H \mathbf{C}_a)^{-1} \mathbf{C}_a^H \mathbf{w}_q \quad (2.32)$$

Theo định nghĩa, vector trọng số \mathbf{w}_q trực giao với các cột của ma trận \mathbf{C}_a . Do đó, vector trọng số tối ưu \mathbf{w}_{ao} bằng 0 đối với môi trường yên tĩnh được mô tả bởi biểu thức (2.31). Do đó, với \mathbf{w}_{ao} bằng 0, chúng ta tìm thấy từ biểu thức (2.15) rằng $\mathbf{w} = \mathbf{w}_q$. Vì lý do này mà \mathbf{w}_q thường được gọi là vector trọng số tĩnh, do đó người ta sử dụng chỉ số dưới q để biểu thị nó.

I.2.3. Mở rộng “Bộ triệt các búp bên tổng quát” cho dây an-ten hai chiều và cải tiến với mạng Neural:

I.2.3.1. Mở rộng “Bộ triệt các búp bên tổng quát” cho dây an-ten hai chiều:

Trên thực tế, trong các hệ thống viễn thông, dây an-ten thường được sử dụng là dây 2 chiều, do đó có thể mô phỏng tín hiệu đến dây an-ten theo hệ tọa độ Oxyz như hình.

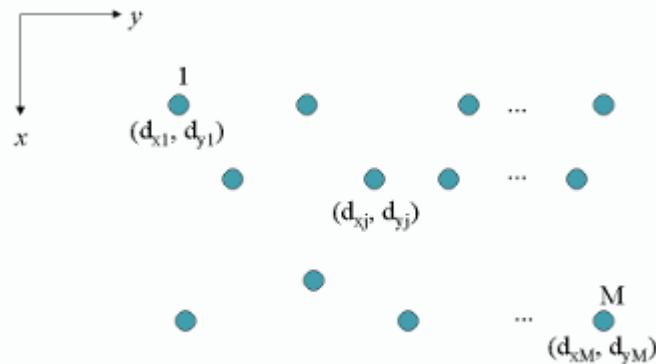


Hình 6. Biểu diễn tín hiệu đến dãy an-ten 2 chiều trong hệ tọa độ Oxyz.

Tín hiệu thứ i có bước sóng λ đến dãy an-ten với hai đặc trưng góc ngǎng θ_i ($-180^\circ \leq \theta_i \leq 180^\circ$) và góc phương vị ϕ_i ($0^\circ \leq \phi_i \leq 90^\circ$), có thể biểu diễn hướng của tín hiệu này trong mặt phẳng Oxy dưới dạng (x_i, y_i) :

$$\begin{cases} x_i = \cos \phi_i \sin \theta_i \\ y_i = \sin \phi_i \sin \theta_i \end{cases} \quad (3.1)$$

Xét dãy an-ten 2 chiều M phần tử được biểu diễn trong mặt phẳng Oxy như hình. Tọa độ của mỗi phần tử an-ten $\mathbf{d}_j = [d_x, d_y]^T$, ($1 \leq j \leq M$). Phần tử đầu tiên là phần tử tham chiếu, do đó $d_{x_1} = d_{y_1} = 0$. Tương tự như dãy an-ten một chiều, các phần tử thuộc cùng một trục Ox hoặc Oy sẽ cách nhau 1 khoảng $\lambda/2$.



Hình 7. Biểu diễn các phần tử an-ten dãy 2 chiều trên mặt phẳng Oxy.

Vector tần số không gian $\Omega_i = [\mu_i, v_i]^T$ của tín hiệu có thể được xác định theo công thức:

$$\begin{cases} \mu_i = -\frac{2\pi}{\lambda} x_i \\ v_i = -\frac{2\pi}{\lambda} y_i \end{cases} \quad (3.2)$$

Do đó, vector lái của dãy an-ten hai chiều ứng với tín hiệu đến được biểu diễn như sau:

$$\mathbf{a}(\mu_i, v_i) = [1, e^{j(d_{x_2}\mu_i + d_{y_2}v_i)}, \dots, e^{j(d_{x_M}\mu_i + d_{y_M}v_i)}]^T = [1, e^{j\mathbf{d}_2^T \Omega_i}, \dots, e^{j\mathbf{d}_M^T \Omega_i}]^T \quad (3.3)$$

I.2.3.2. Lý thuyết về mạng Neural:

I.2.3.2.1. Giới thiệu về mạng Neural:

Mạng neural (Neural Networks) là một hệ thống tính toán được lấy cảm hứng từ cách hoạt động của hệ thống thần kinh sinh học trong não người. Nó được sử dụng trong lĩnh vực trí tuệ nhân tạo (Artificial Intelligence) để học và nhận biết các mô hình và quy luật từ dữ liệu đầu vào.

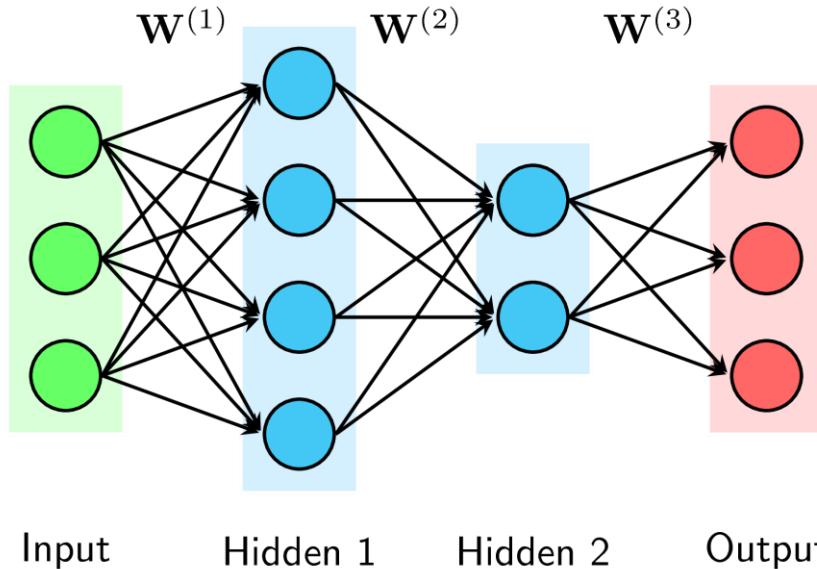
Mạng neural bao gồm một tập hợp các đơn vị tính toán gọi là "nơ-ron nhân tạo" hoặc "nơ-ron nhân tạo" (artificial neurons), được tổ chức thành các lớp. Mỗi nơ-ron nhân tạo nhận đầu vào, thực hiện một phép tính và truyền kết quả cho các nơ-ron khác. Các nơ-ron trong mạng neural thường được kết nối với nhau thông qua các trọng số, và các trọng số này được cập nhật trong quá trình học. Quá trình học này thường được thực hiện thông qua một quá trình gọi là "học có giám sát" (supervised learning), trong đó mạng neural được cung cấp các ví dụ đầu vào đã biết và đầu ra tương ứng. Mạng neural sẽ điều chỉnh các trọng số của nó dựa trên sai số giữa đầu ra dự đoán và đầu ra thực tế, nhằm tối thiểu hóa sai số này.

Có rất nhiều mô hình mạng Neural, nhưng cơ bản nhất và được sử dụng trong nghiên cứu này là Multi - layer Perceptron (MLP).

I.2.3.2.2. Kiến trúc của mạng Neural Multi – layer Perceptron (MLP):

a) Lớp (layers):

Ngoài lớp ngõ vào (input layer) và lớp ngõ ra (output layer), một Multi - layer Perceptron (MLP) có thể có nhiều lớp ẩn (hidden layers) ở giữa. Các lớp ẩn theo thứ tự từ lớp ngõ vào đến lớp ngõ ra được đánh số thứ tự là “Hidden layer 1, Hidden layer 2, ...”. Hình dưới đây là một ví dụ với 2 lớp ẩn.



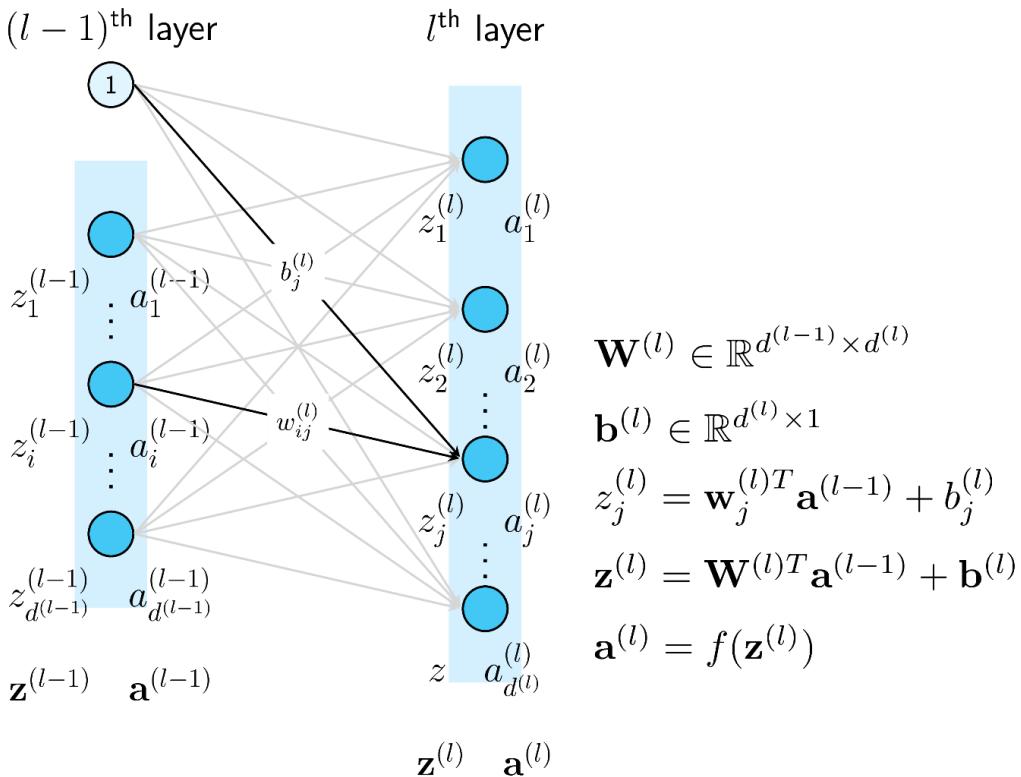
Hình 8. Các lớp của mô hình mạng Neural Multi-layer Perceptron.

b) Điểm đơn vị (Units):

Một “node” hình tròn trong một lớp được gọi là một điểm đơn vị (unit). Điểm ở các lớp ngõ vào, các lớp ẩn, và lớp ngõ ra được lần lượt gọi là điểm ngõ vào, điểm ẩn, và điểm ngõ ra. Đầu vào của các lớp ẩn được ký hiệu bởi z , đầu ra của mỗi uđiểm thường được ký hiệu là a (thể hiện hàm kích hoạt (activation), là giá trị của mỗi unit sau khi áp dụng hàm kích hoạt (activation function) lên z). Đầu ra của điểm thứ i trong lớp thứ l được ký hiệu là $a_i^{(l)}$. Giả sử thêm rằng số điểm trong lớp thứ l (không tính “bias”) là $d^{(l)}$. Vector biểu diễn ngõ ra của layer thứ l được ký hiệu là $\mathbf{a}^{(l)} \in R^{d^{(l)}}$.

c) Trọng số (weights) và “biases”:

Có L ma trận trọng số cho một MLP có L layers. Các ma trận này được ký hiệu là $\mathbf{W}^{(l)} \in R^{d^{(l-1)} \times d^{(l)}}$, $l = 1, 2, \dots, L$ trong đó $\mathbf{W}^{(l)}$ thể hiện các kết nối từ lớp thứ $l - 1$ tới lớp thứ l . Cụ thể hơn, phần tử $w_{ij}^{(l)}$ thể hiện kết nối từ “node” thứ i của lớp thứ $l - 1$ tới “node” thứ j lớp thứ l . Các “biases” của lớp thứ l được ký hiệu là $\mathbf{b}^{(l)} \in R^{d^{(l)}}$. Các trọng số này được ký hiệu như hình. Tập hợp các trọng số và “biases” là \mathbf{W} và \mathbf{b} .



Hình 9. Cách tính trọng số và “biases” của từng lớp trong mạng.

d) Hàm kích hoạt (activation function):

Mỗi ngõ ra của một điểm (trừ các điểm ngõ vào) được tính dựa vào công thức:

$$a_i^{(l)} = f\left(\mathbf{w}_i^{(l)T} \mathbf{a}_i^{(l-1)} + b_i^{(l)}\right) \quad (3.4)$$

Trong đó $f(\cdot)$ là một hàm phi tuyến. Ở dạng vector, biểu thức trên được viết lại:

$$\mathbf{a}_i^{(l)} = f\left(\mathbf{W}_i^{(l)T} \mathbf{a}_i^{(l-1)} + \mathbf{b}_i^{(l)}\right) \quad (3.5)$$

Có rất nhiều hàm kích hoạt được sử dụng trong mạng Neural như hàm “tanh”, hàm “ReLU”, Hàm kích hoạt được sử dụng để tài nghiên cứu là hàm “sigmoid”:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.6)$$

Đạo hàm của hàm “sigmoid”:

$$f'(x) = x(1 - x) \quad (3.7)$$

I.2.3.2.3. Công thức tính toán trọng số và “biases” cho mạng Neural MLP:

Thực hiện tính toán và cập nhật trọng số, “biases” cho mạng Neural MLP theo các bước sau:

Bước	Thực hiện	Biểu thức tính toán
1	Tính toán ngõ vào	$\mathbf{A}(1) = \mathbf{X} \quad (3.8)$

2	Tính toán xuôi (forward)	Với $l = 2, 3, \dots, L$, tính toán: $\mathbf{Z}(l) = \mathbf{W}(l)\mathbf{A}(l-1) + \mathbf{B}(l) \quad (3.9)$ $\mathbf{A}(l) = f(\mathbf{Z}(l)) \quad (3.10)$ $f'(\mathbf{Z}(l)) \quad (3.11)$ $\mathbf{D}(l) = (\mathbf{A}(l) - \mathbf{R}) \odot f'(\mathbf{Z}(l)) \quad (3.11)$
3	Tính toán ngược (backpropagation)	Với $l = L-1, L-2, \dots, 2$, tính toán: $\mathbf{D}(l) = (\mathbf{W}^T(l+1)\mathbf{D}(l+1)) \odot f'(\mathbf{Z}(l)) \quad (3.12)$
4	Cập nhật trọng số và “biases”	Với $l = 2, 3, \dots, L$, tính toán: $\mathbf{W}(l) = \mathbf{W}(l) - \alpha \frac{\mathbf{D}(l)}{n_p} \mathbf{A}^T(l-1) \quad (3.13)$ $\mathbf{b}(l) = \mathbf{b}(l) - \alpha \sum_{k=1}^{n_p} \boldsymbol{\delta}_k(l) \quad (3.14)$ $\mathbf{B}(l) = \text{concatenate}\{\mathbf{b}(l)\}(n_p \text{ lần}) \quad (3.15)$

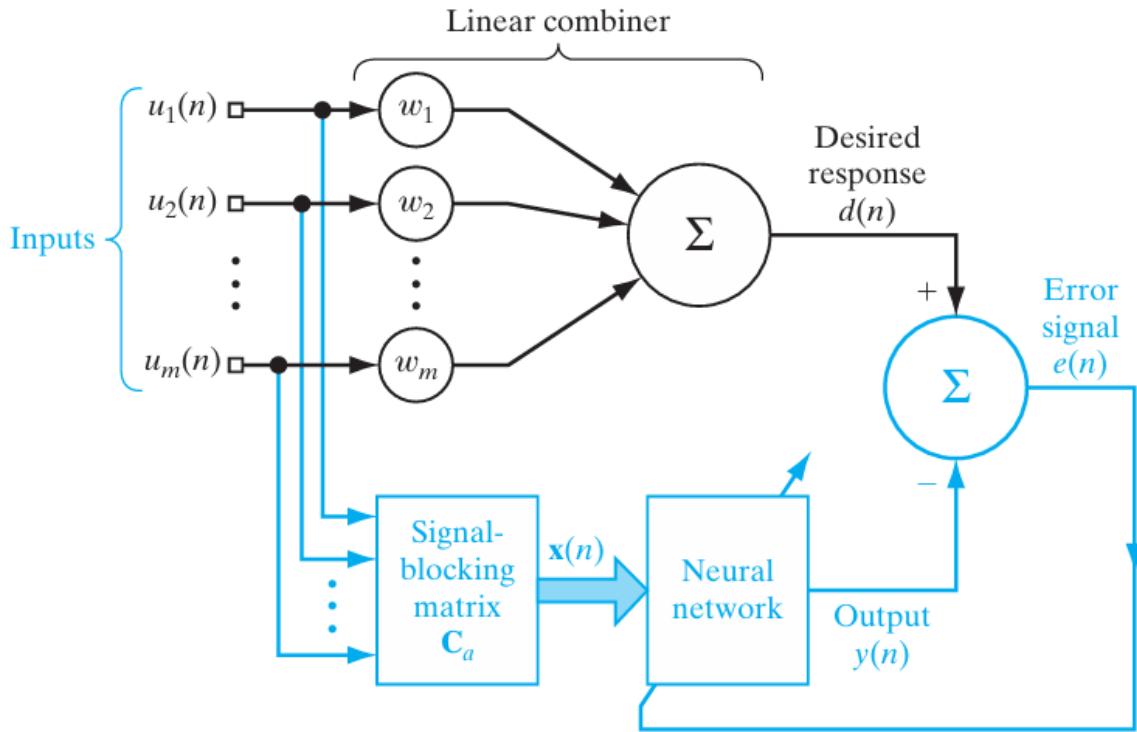
Trong đó:

- Dữ liệu ngõ vào \mathbf{X} có kích thước $n \times n_p$, trong đó n là số chiều (features) của dữ liệu và n_p là số mẫu (patterns) dữ liệu.
- Dữ liệu ngõ ra thực tế \mathbf{R} .
- Ngõ ra của lớp thứ l $\mathbf{Z}(l)$ và hàm kích hoạt $\mathbf{A}(l) = f(\mathbf{Z}(l))$ của lớp đó.
- Ma trận sai số tại lớp thứ l $\mathbf{D}(l)$ với các cột là các vector sai số $\boldsymbol{\delta}_k(l), k = 1, 2, \dots, n_p$.
- Phép $\text{concatenate}\{\mathbf{b}(l)\}$ chính là phép thêm các vector “biases” $\mathbf{b}(l)$ vào ma trận “biases” $\mathbf{B}(l)$.
- Hệ số học của mạng (learning rate) α được chọn từ $\alpha = 0.01, 0.02, \dots, 0.2$ để tiến hành khảo sát.

I.2.3.3. Cải tiến “Bộ triệt các búp bên tổng quát” với mạng Neural:

Đối với “Bộ triệt các búp bên tổng quát” truyền thống, ta thực hiện tính toán vector trọng số tối ưu \mathbf{w}_{ao} theo công thức (2.30) và rút ra được vector trọng số tổng quát theo công thức (2.15). Có thể thấy rõ việc tính toán truyền thống sẽ gặp rất nhiều khó khăn khi mà dữ liệu ngõ vào chúng ta rất lớn, dẫn đến các ma trận, vector trong các công thức tính toán cũng sẽ có kích thước rất lớn. Sự phát triển của công nghệ hiện nay, đặc biệt là Trí tuệ nhân tạo/Học Máy (AI/Machine Learning), đã mang đến nhiều giải pháp tối ưu, cải thiện khả năng tính toán hơn rất nhiều.

Ứng dụng của Học Máy cho “Bộ triệt các búp bên tổng quát” đã được đề xuất với việc sử dụng mạng Neural để tính toán cho phần tử vector trọng số tối ưu \mathbf{w}_{ao} . Cụ thể, thay phần tử \mathbf{w}_{ao} ở “Sơ đồ khói của bộ triệt các búp bên tổng quát GSC” (Hình 4 và 5) bằng mạng Neural như sau:



Hình 10. Sơ đồ khói “Bộ triệt các búp bên tổng quát” được cải tiến với mạng Neural.

Mạng Neural được sử dụng trong ứng dụng này được thiết kế bao gồm:

- Số điểm ngõ vào: tương đồng kích thước của $x(n)$ đã được trai phẳng (flatten), tương ứng với số tính chất/đặc điểm (features) của dữ liệu ngõ vào trong mô hình mạng Neural.
- Số điểm ngõ ra của mạng sẽ tương ứng với kích thước của ngõ ra mong muốn $d(n)$.
- Số điểm lớp ẩn: khảo sát số điểm lớp ẩn từ $10 \div 200$ và tìm ra số điểm lớp ẩn phù hợp. Số điểm lớp ẩn quá lớn/quá nhỏ sẽ dẫn đến hiện tượng “overfitting/underfitting”, sẽ cho hiệu suất kém khi kiểm tra dữ liệu mới.
- Hệ số học: hệ số học của các mạng Neural thường được chọn từ $0.01 \div 0.1$. Tiến hành khảo sát và tìm ra hệ số học phù hợp với mạng Neural này.
- Hàm mất mát (lost function): đo lường sự khác biệt giữa giá trị dự đoán của mạng và giá trị thực tế, cung cấp phản hồi (feedback) cho mạng và điều chỉnh các trọng số (weights) để giảm thiểu hàm mất mát. Thông thường, khi giá trị hàm mất mát nhỏ hơn 10^{-6} thì được xem là đã hội tụ. Từ sơ đồ có thể thấy, hàm mất mát chính là tín hiệu lỗi $e(n)$, là sai lệch của ngõ ra mong muốn và ngõ ra của mạng Neural.
- Tập dữ liệu cho quá trình huấn luyện (training) và kiểm tra (testing): Xây dựng tập dữ liệu với giải thuật GSC truyền thống bằng cách thay đổi phương sai nhiễu từ $0.005 \div 1$. Ngõ vào của tập dữ liệu là các tín hiệu $x(n)$ và ngõ ra là các ngõ ra mong muốn $d(n)$ cho từng trường hợp phương sai nhiễu. Tập dữ liệu được chia thành tập huấn luyện và tập kiểm tra theo tỉ lệ $1/1$.

Với mạng Neural, việc tìm được \mathbf{w}_{ao} trở nên dễ dàng và nhanh chóng, đặc biệt khi dữ liệu cực lớn và phức tạp, qua đó việc tính toán trọng số tổng quát tối ưu \mathbf{w}_{opt} cho ứng dụng cũng nhanh hơn rất nhiều.

CHƯƠNG II: MÔ PHỎNG GIẢI THUẬT TRÊN MATLAB

II.1. Mô phỏng giải thuật “Bộ lọc tối thiểu phuơng sai có ràng buộc tuyến tính” và “Bộ triệt các búp bên tổng quát” với dãy an-ten một chiều:

Bài toán mô phỏng giải thuật:

Xét các tín hiệu đến là 3 tín hiệu có hướng tới tạo với phuơng vuông góc của dãy an-ten thu tín hiệu lần lượt là $\varphi_0, \varphi_1, \varphi_2$ (tín hiệu mong muốn tại góc φ_0). Trong đó, tín hiệu mong muốn thu được có hướng tới tại góc φ_0 , 2 tín hiệu còn lại là can nhiễu. Dãy an-ten thu tín hiệu gồm M phần tử, do đó vector trọng số của dãy cũng có M phần tử tương ứng. Tìm giải thuật tối ưu tín hiệu mong muốn.

Phân tích bài toán:

Tìm vector trọng số $w = [w_0, w_1, w_2, \dots, w_{M-1}]^T$ sao cho phuơng sai ngõ ra $P_o = E[|y(n)|^2]$ nhỏ nhất đồng thời thỏa $w^H s(\theta_0) = g$ với g là độ lợi có giá trị thực cho trước. Trong đó

$$s(\theta_0) = [1, e^{-j\theta_0}, \dots, e^{-j(M-1)\theta_0}]^T \quad (4.1)$$

Là vector lái ứng với góc tới mong muốn; góc điện θ_0 được xác định như sau:

$$\theta_0 = 2\pi \frac{\Delta}{\lambda} \sin \varphi_0 \quad (4.2)$$

Trong đó, Δ là khoảng cách giữa các phần tử an-ten, thông thường chuẩn hóa $\Delta = 0.5\lambda$.

Dựa vào các lý thuyết trình bày ở chương I, thực hiện mô phỏng 2 giải thuật nghiên cứu bằng phần mềm MATLAB. Các thông số cho trước: $\varphi_0 = -60^\circ, \varphi_1 = -50^\circ, \varphi_2 = -40^\circ$. độ lợi g được chuẩn hóa $g = 1$; phuơng sai tín hiệu ngõ vào đều bằng 1, phuơng sai nhiễu 0.1, số phần tử an-ten $M = 20$. Để thuận tiện cho việc quan sát kết quả giải thuật, vẽ đồ thị tuyến tính $|B(\varphi)|^2$ và tính toán Root Mean-square Error là độ sai lệch trung bình của tín hiệu tại ngõ ra và tín hiệu muôn thu, với k là số mẫu tín hiệu:

$$RMSE = \sqrt{\frac{1}{k} \sum_{n=0}^{k-1} |y(n) - d(n)|^2}; \quad (4.3)$$

II.1.1. Tối thiểu phuơng sai có ràng buộc tuyến tính (LCMV):

II.1.1.1. Lời giải để mô phỏng giải thuật:

Từ biểu thức (1.17), với g có giá trị thực, vector lái ứng với hướng tới của tín hiệu mong muôn $s(\theta_0)$ ta có:

$$w_{opt} = \frac{g R^{-1} s(\theta_0)}{s^H(\theta_0) R^{-1} s(\theta_0)} \quad (4.4)$$

Với ma trận tương quan ngõ vào R được xác định:

$$\mathbf{R} = \mathbf{R}_0 + \mathbf{R}_1 + \mathbf{R}_2 \quad (4.5)$$

Trong đó, $\mathbf{R}_0, \mathbf{R}_1, \mathbf{R}_2$ là ma trận tương quan ngõ vào của các tín hiệu tới, được xác định:

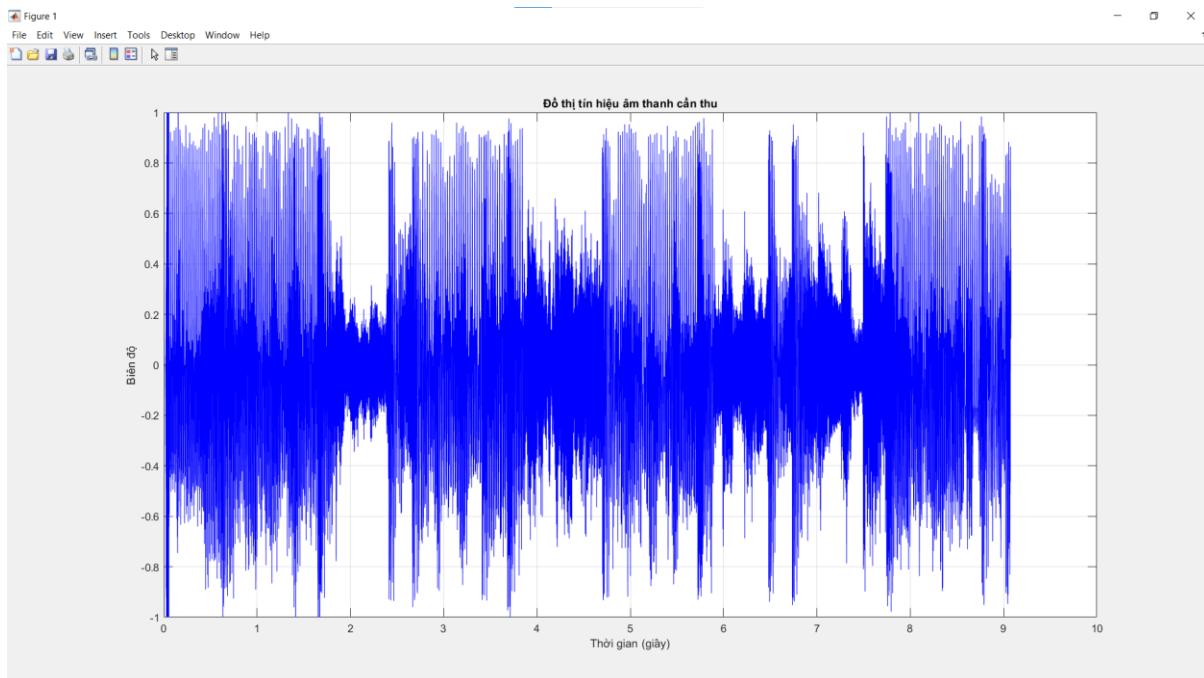
$$\mathbf{R}_i = E[\mathbf{u}_i(n)\mathbf{u}_i^H(n)] = \mathbf{s}^H(\theta_i)P_i\mathbf{s}(\theta_i) \quad (4.6)$$

Với P_i là phương sai ngõ vào (công suất) của tín hiệu ở góc điện thứ i .

II.1.1.2. Thực hiện mô phỏng trên MATLAB:

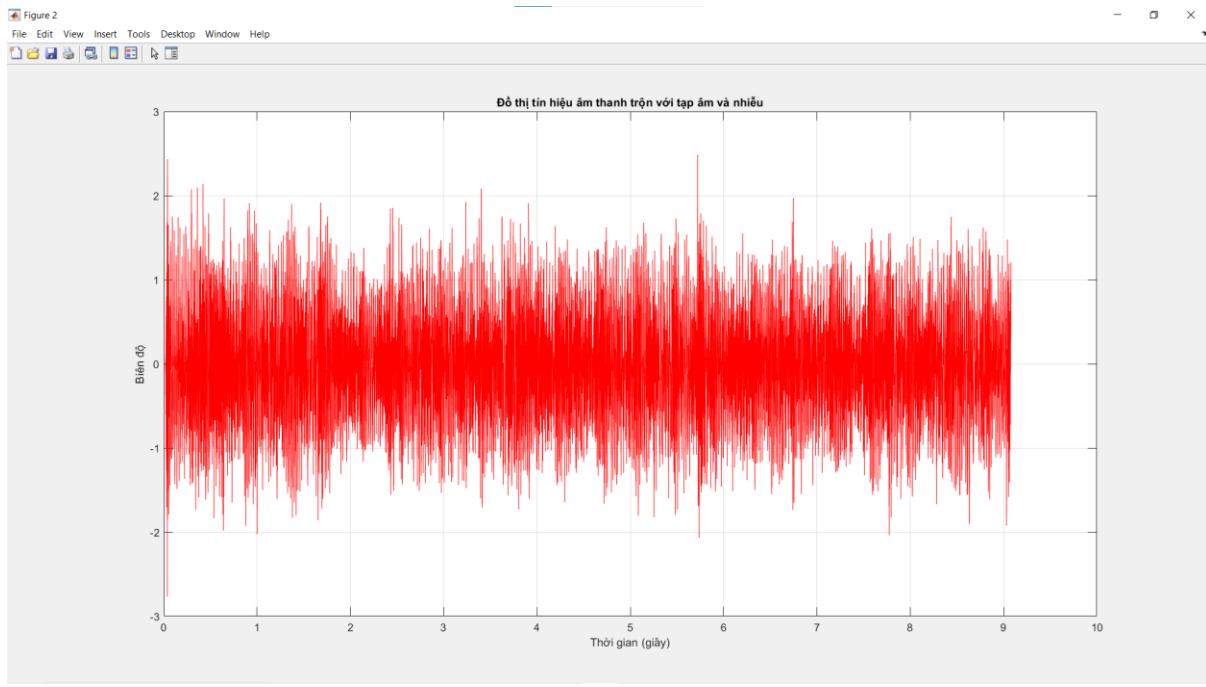
- ❖ Ngõ vào của bộ lọc:

- a) Tín hiệu cần xử lý:



Hình 11. Tín hiệu gốc cần xử lý.

- b) Tín hiệu tại ngõ vào của bộ lọc, gồm tín hiệu cần thu, 2 can nhiễu và nhiễu:

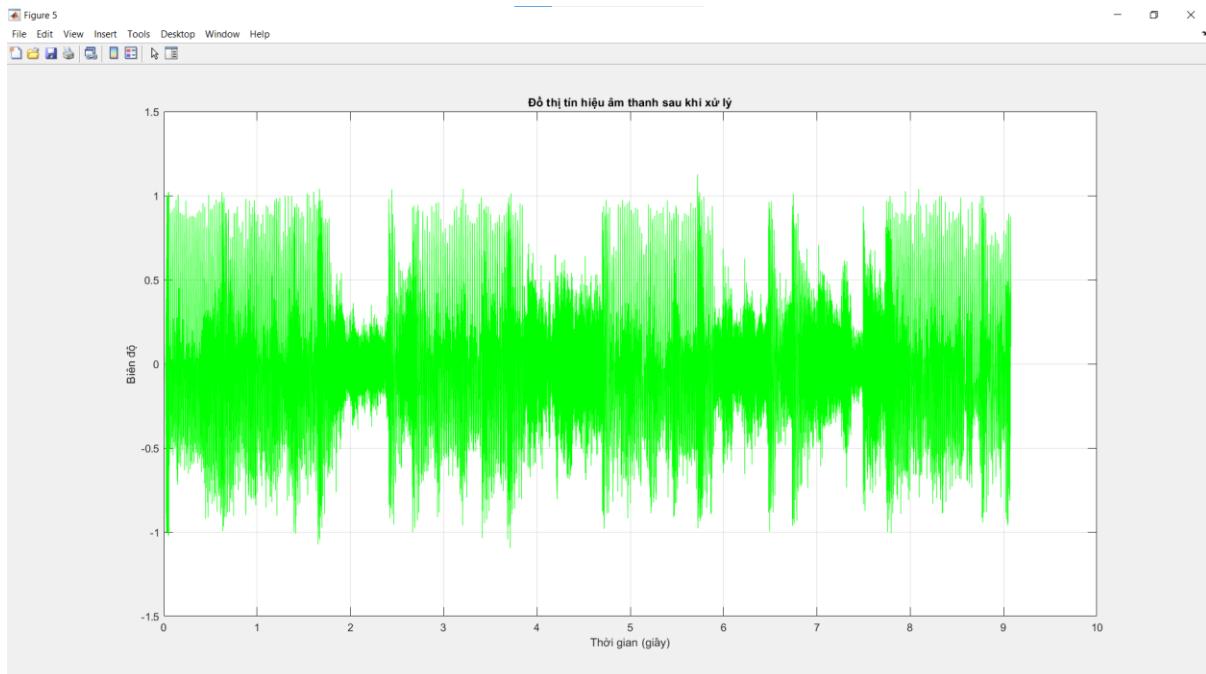


Hình 12. Tín hiệu trộn tại ngõ vào của bộ lọc.

Nhân xét: Tín hiệu trộn gồm tín hiệu mong muốn và 2 can nhiễu kết hợp với nhiễu tại ngõ vào cho kết quả rất hỗn tạp như trên đồ thị.

❖ Kết quả mô phỏng:

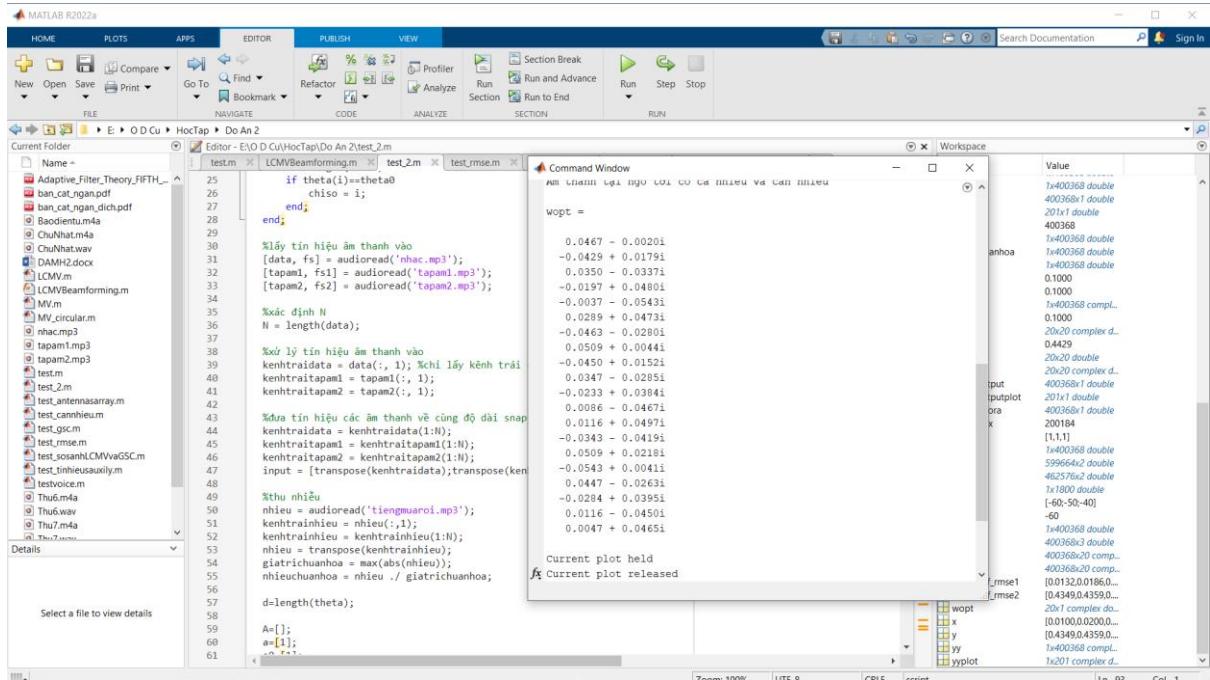
- a) Tín hiệu sau khi đã xử lý tại ngõ ra bộ lọc:



Hình 13. Tín hiệu sau khi qua xử lý.

Nhân xét: Tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

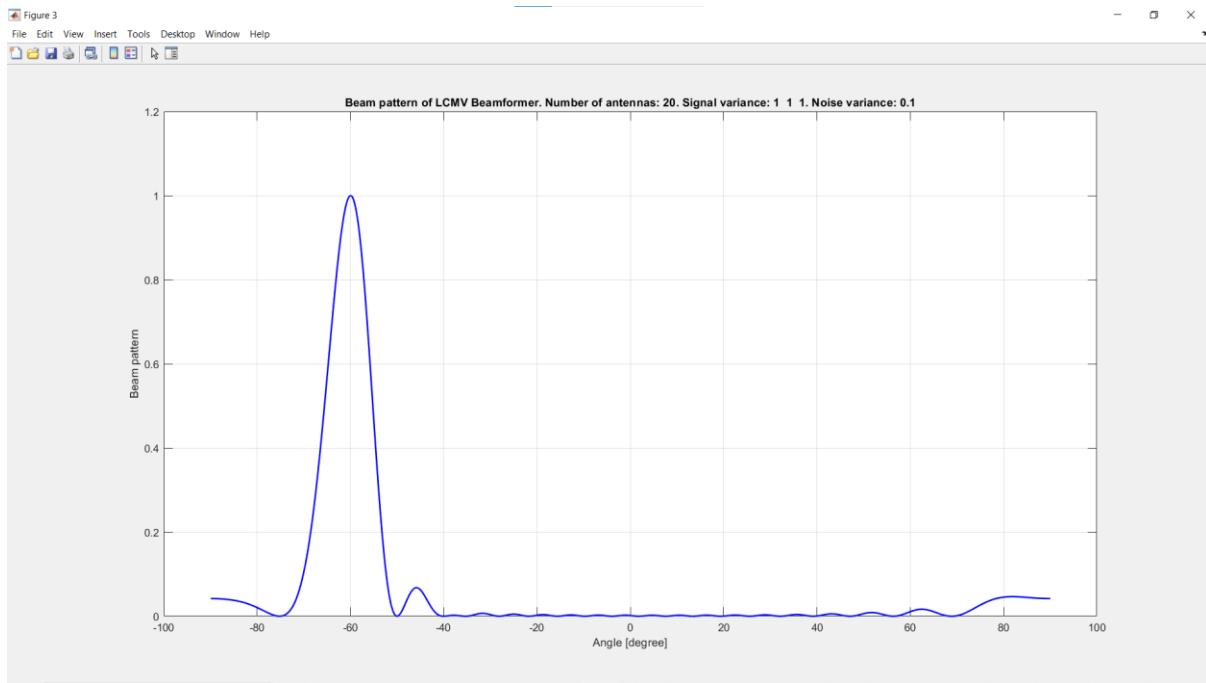
b) Trọng số tối ưu:



Hình 14. Trọng số tối ưu của giải thuật.

$$\begin{aligned}
w_{opt} = & [0.0467 - 0.0020i, \quad -0.0429 + 0.0179i, \\
& 0.0350 - 0.0337i, \quad -0.0197 + 0.0480i, \\
& -0.0037 - 0.0543i, \quad 0.0289 + 0.0473i, \\
& -0.0463 - 0.0280i, \quad 0.0509 + 0.0044i, \\
& -0.0450 + 0.0152i, \quad 0.0347 - 0.0285i, \\
& -0.0233 + 0.0384i, \quad 0.0086 - 0.0467i, \\
& 0.0116 + 0.0497i, \quad -0.0343 - 0.0419i, \\
& 0.0509 + 0.0218i, \quad -0.0543 + 0.0041i, \\
& 0.0447 - 0.0263i, \quad -0.0284 + 0.0395i, \\
& 0.0116 - 0.0450i, \quad 0.0047 + 0.0465i]^T
\end{aligned}$$

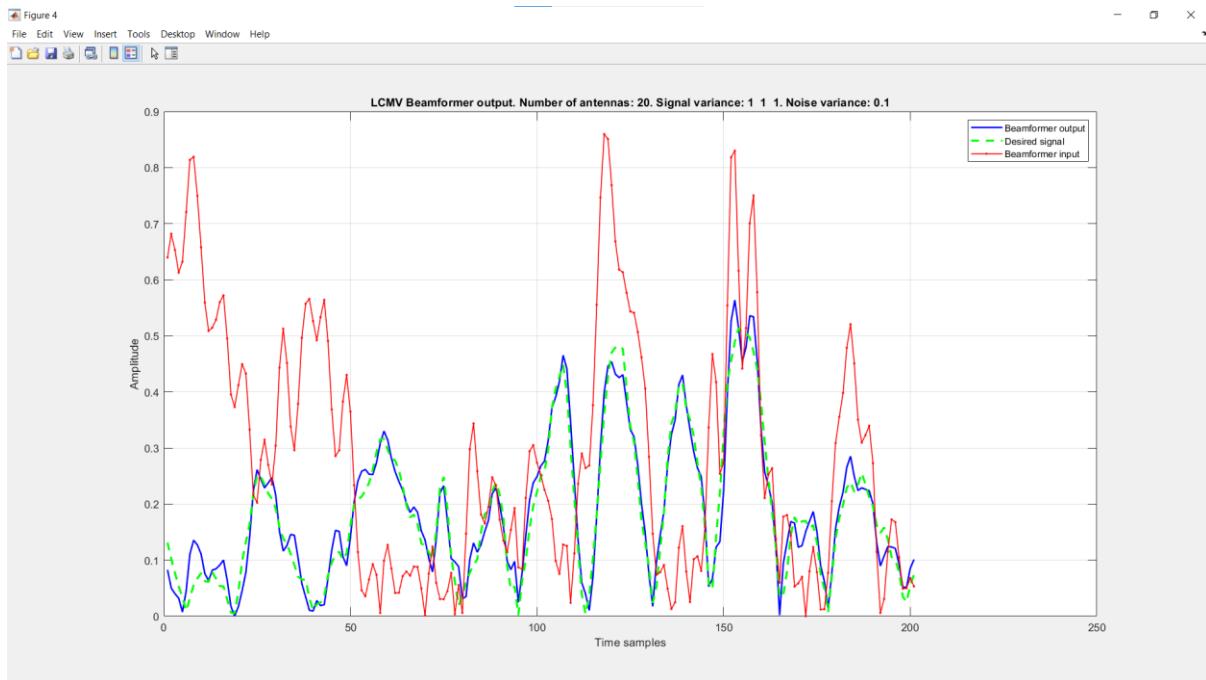
c) Đồ thị tuyển tính:



Hình 15. Đồ thị tần số của giải thuật LCMV.

Nhân xét: Tín hiệu mong muốn tương ứng với góc tới -60° được tối ưu, với hướng búp sóng chính cực đại, đồng thời có nhiều tại góc tới -50° và -40° được triệt tiêu là các điểm null (điểm không), chứng tỏ giải thuật cho khả năng triệt nhiễu cao và tối ưu tín hiệu mong muốn thu được khá tốt.

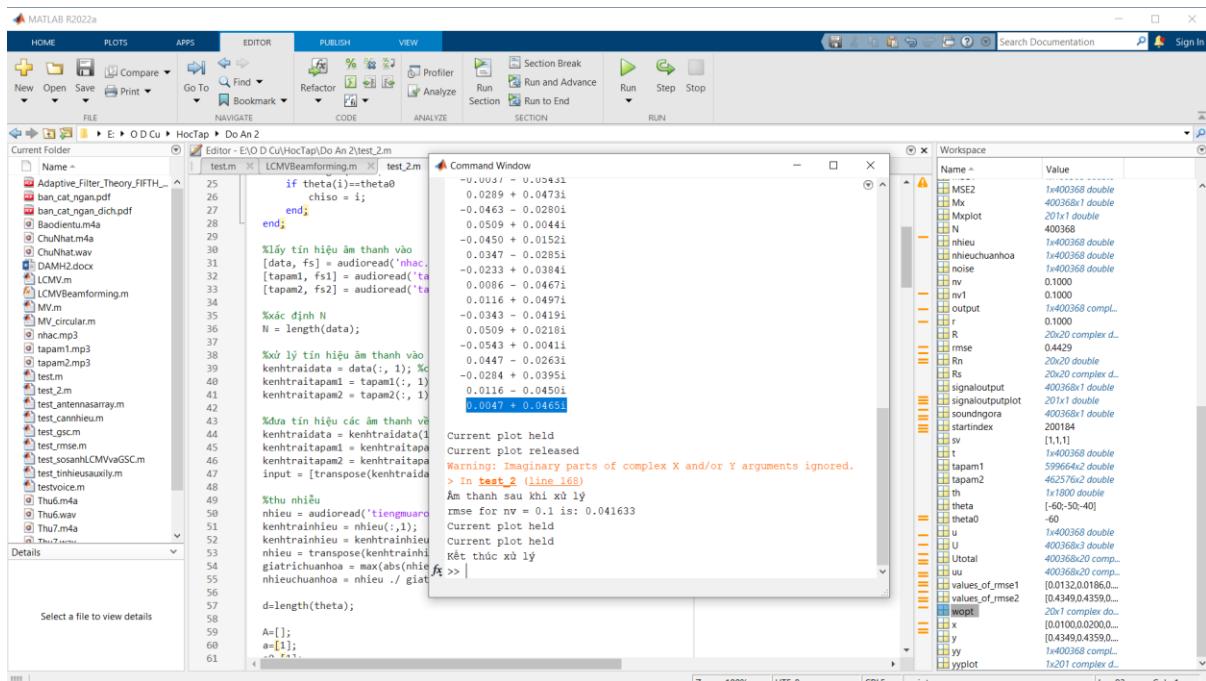
- d) So sánh giữa tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra bộ lọc:



Hình 16. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra.

Nhân xét: Đồ thị biểu diễn tín hiệu tại ngõ vào (màu đỏ), tín hiệu mong muốn (màu xanh) và tín hiệu tại ngõ ra (màu lục). Có thể thấy 2 đường màu xanh và màu lục rất tương đồng nhau với sai số rất nhỏ, cho thấy tín hiệu tại ngõ ra tương đồng với tín hiệu mong muốn.

e) Root Mean-square Error:



Hình 17. RMSE của giải thuật LCMV.

$RMSE = 0.041633$.

Nhân xét: RMSE nhỏ cho thấy tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

II.1.2. Bộ triệt các búp bên tống quát (GSC):

II.1.2.1. Lời giải để mô phỏng giải thuật:

Khác biệt đối với giải thuật LCMV chính là lúc này không chỉ có một mà là nhiều ràng buộc tuyến tính, đặc trưng bởi ma trận ràng buộc tuyến tính \mathbf{C}_a . Tương ứng lúc này, vector độ lợi \mathbf{g} có dạng [1,0,0] với góc của hướng tới mong muốn tại φ_0 . Vector trọng số tối ưu cho giải thuật này được suy ra từ biểu thức (2.30):

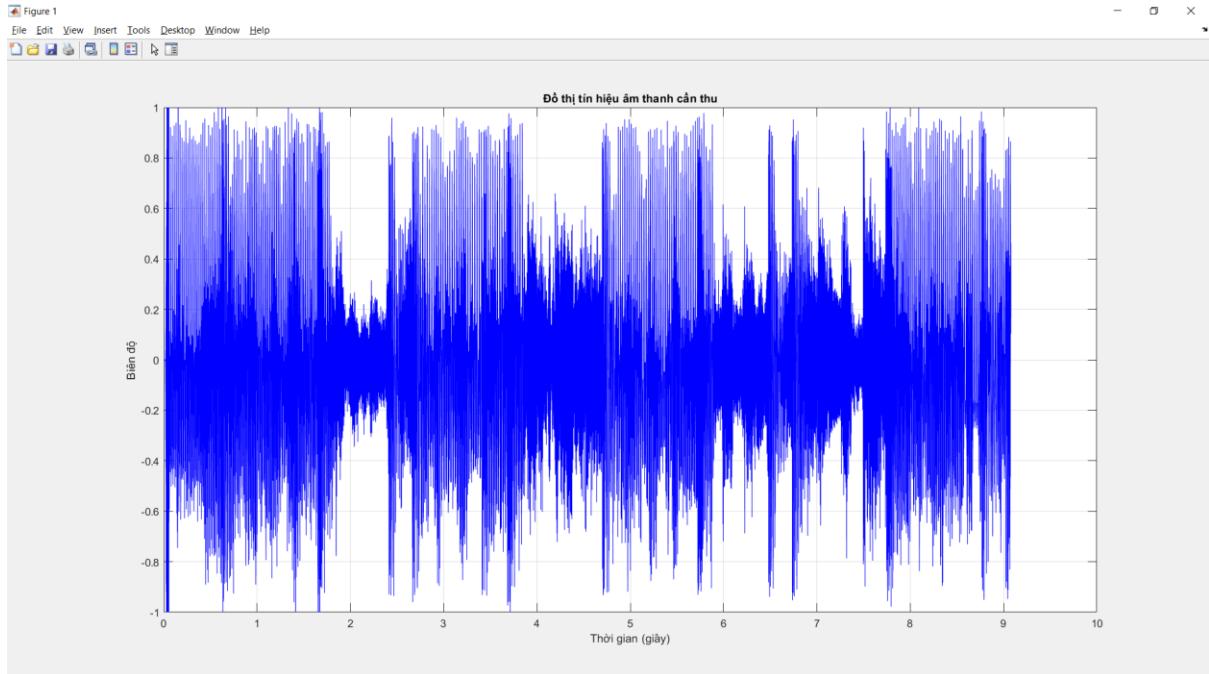
$$\mathbf{w}_{opt} = (\mathbf{C}_a^H \mathbf{R} \mathbf{C}_a)^{-1} \mathbf{C}_a^H \mathbf{R} \mathbf{w}_q \quad (4.7)$$

Trong đó, ma trận tương quan ngõ vào \mathbf{R} được xác định như đối với LCMV; ma trận ràng buộc tuyến tính \mathbf{C}_a và vector trọng số tĩnh \mathbf{w}_q được xác định như lý thuyết chương I.

II.1.2.2. Thực hiện mô phỏng trên MATLAB:

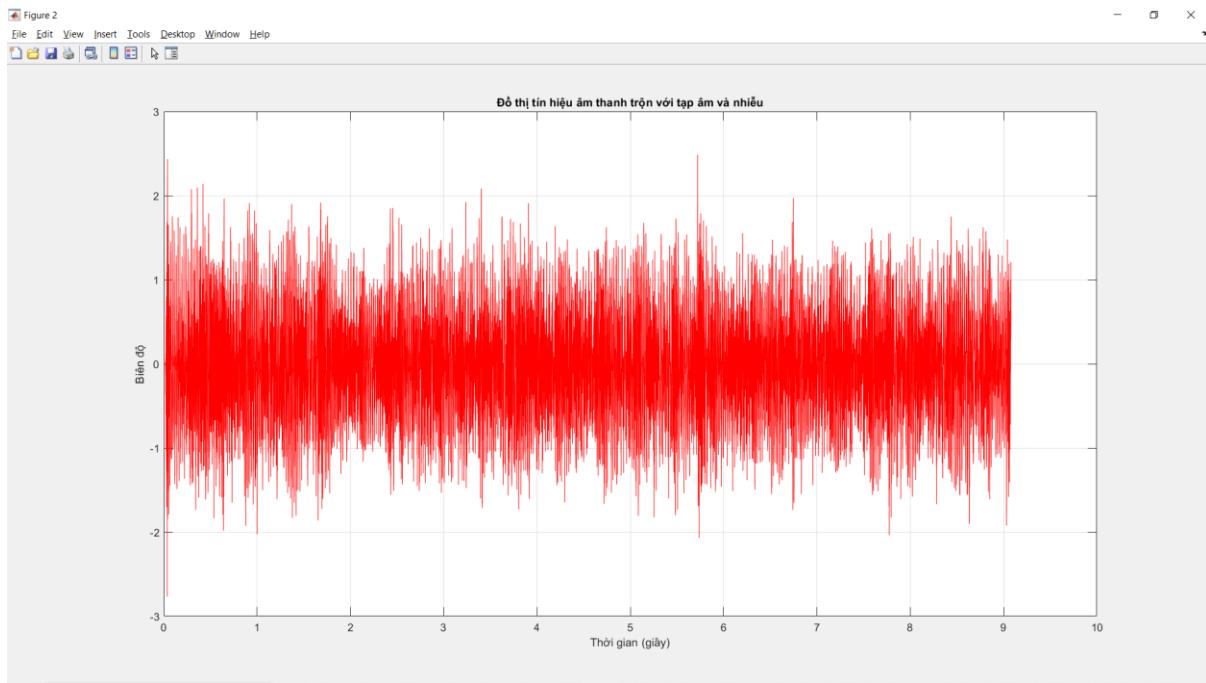
❖ Ngõ vào của bộ lọc:

a) Tín hiệu gốc muốn thu được:



Hình 18. Tín hiệu gốc muốn thu.

b) Tín hiệu tại ngõ vào của bộ lọc, gồm tín hiệu cần thu, 2 can nhiễu và nhiễu:

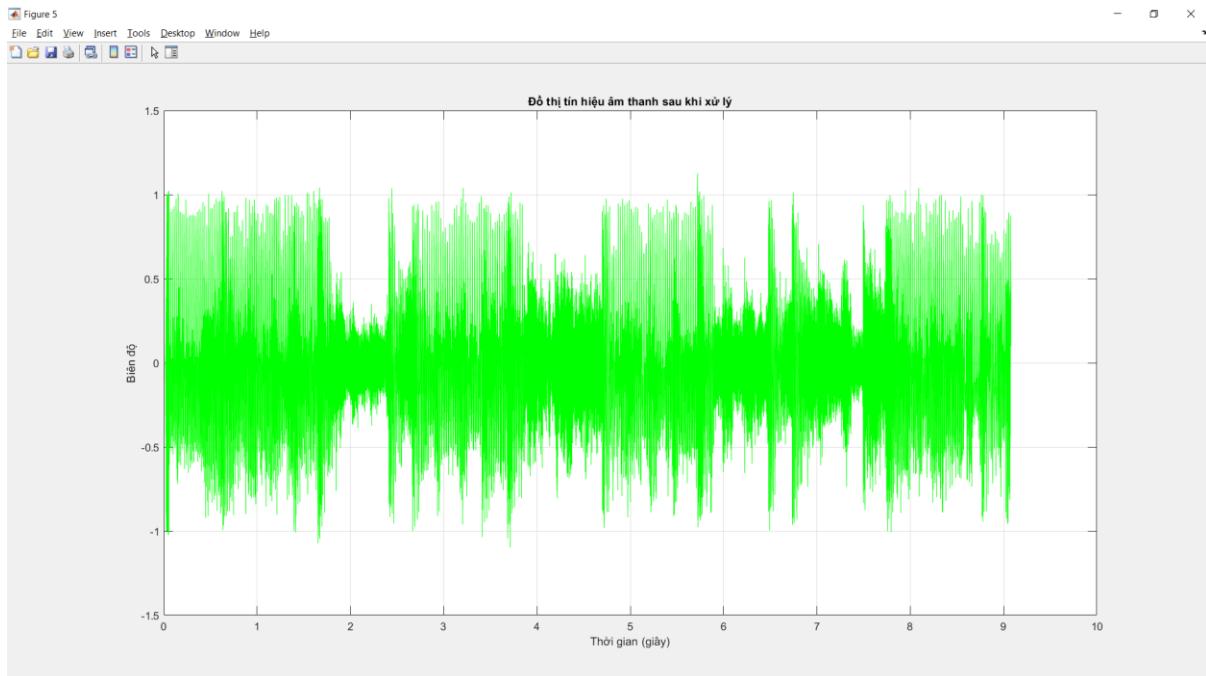


Hình 19. Tín hiệu trộn tại ngõ vào bộ lọc.

Nhận xét: Tương tự với LCMV, tín hiệu trộn gồm tín hiệu mong muốn và 2 can nhiễu kết hợp với nhiễu tại ngõ vào cho kết quả rất hỗn tạp như trên đồ thị.

❖ **Kết quả mô phỏng:**

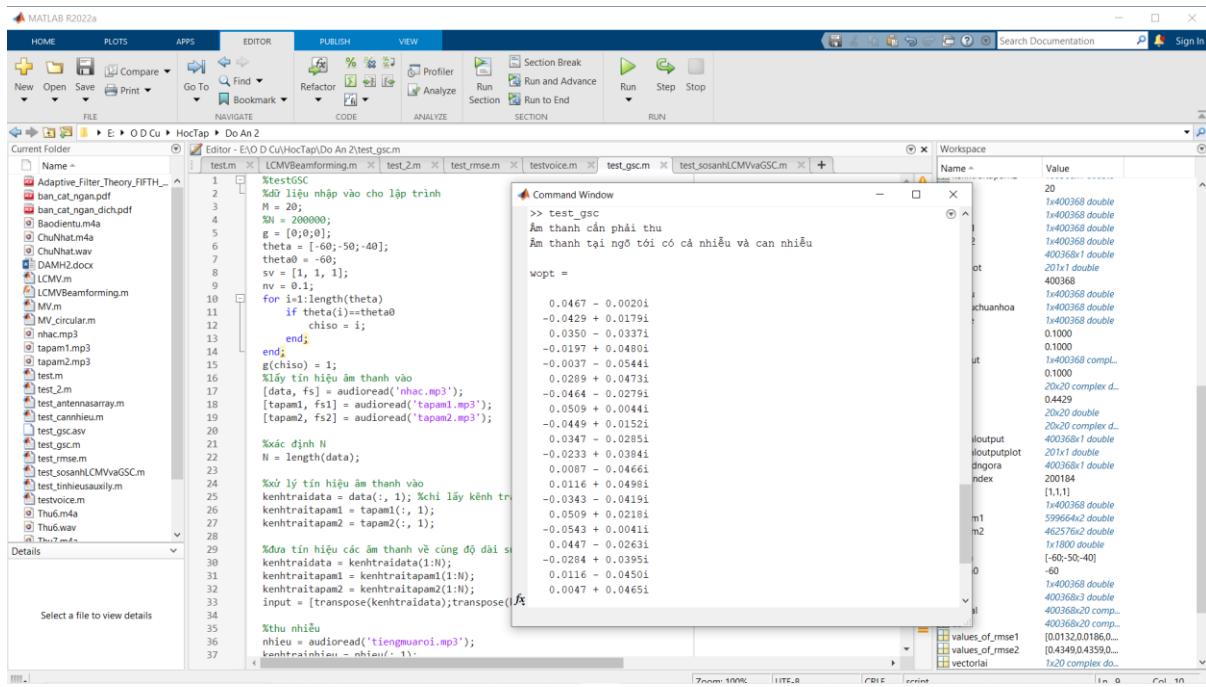
- a) Tín hiệu sau khi đã xử lý tại ngõ ra bộ lọc:



Hình 20. Tín hiệu sau khi xử lý tại ngõ ra.

Nhận xét: Tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

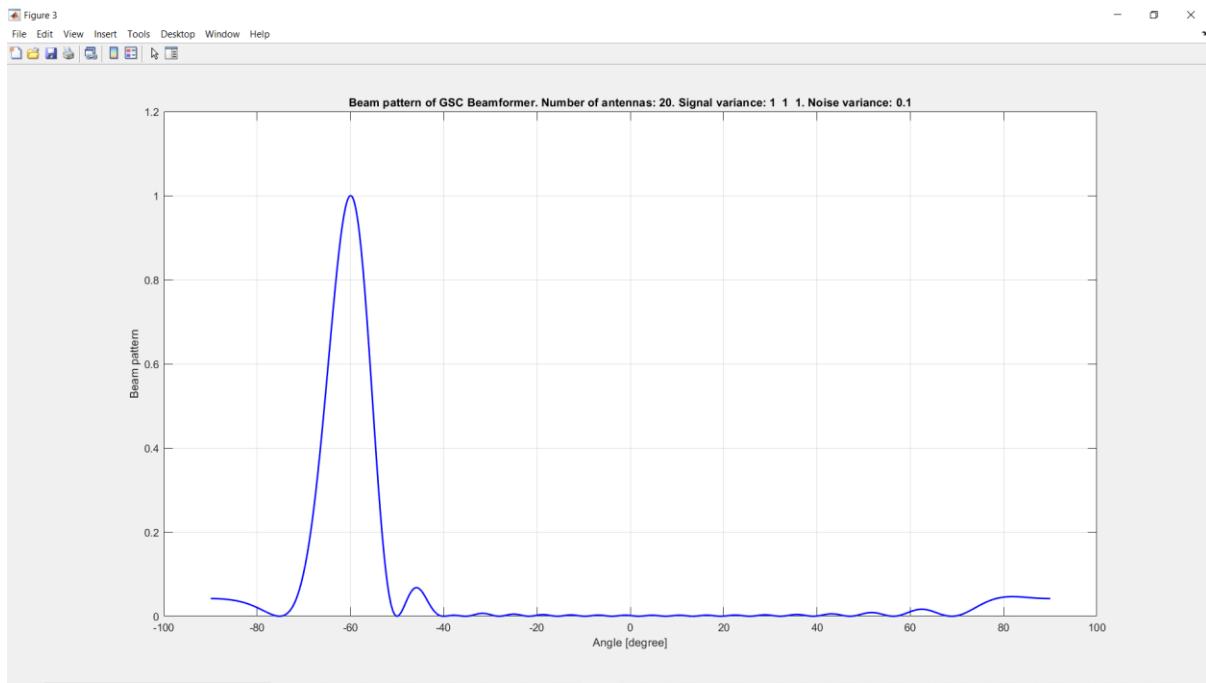
- b) Trọng số tối ưu:



Hình 21. Trọng số tối ưu của giải thuật.

$$\begin{aligned}
 \mathbf{w}_{opt} = & [0.0467 - 0.0020i, \quad -0.0429 + 0.0179i, \\
 & 0.0350 - 0.0337i, \quad -0.0197 + 0.0480i, \\
 & -0.0037 - 0.0544i, \quad 0.0289 + 0.0473i, \\
 & -0.0464 - 0.0279i, \quad 0.0509 + 0.0044i, \\
 & -0.0449 + 0.0152i, \quad 0.0347 - 0.0285i, \\
 & -0.0233 + 0.0384i, \quad 0.0087 - 0.0466i, \\
 & 0.0116 + 0.0498i, \quad -0.0343 - 0.0419i, \\
 & 0.0509 + 0.0218i, \quad -0.0543 + 0.0041i, \\
 & 0.0447 - 0.0263i, \quad -0.0284 + 0.0395i, \\
 & 0.0116 - 0.0450i, \quad 0.0047 + 0.0465i]^T
 \end{aligned}$$

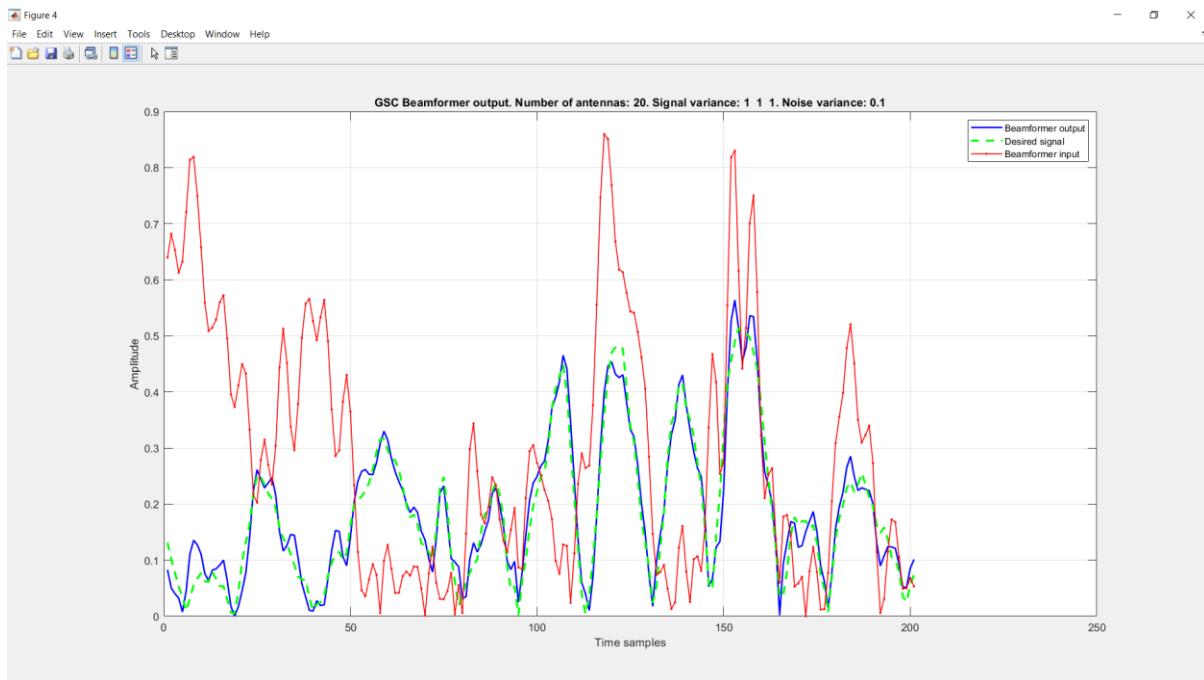
c) Đồ thị tuyển tính:



Hình 22. Đồ thị tần số của giải thuật GSC.

Nhận xét: Tương tự với giải thuật LCMV, tín hiệu mong muốn tương ứng với góc tới -60° được tối ưu, với hướng búp sóng chính cực đại, đồng thời can nhiễu tại góc tới -50° và -40° được triệt tiêu là các điểm null (điểm không), chứng tỏ giải thuật cho khả năng triệt can nhiễu và tối ưu tín hiệu mong muốn thu được khá tốt.

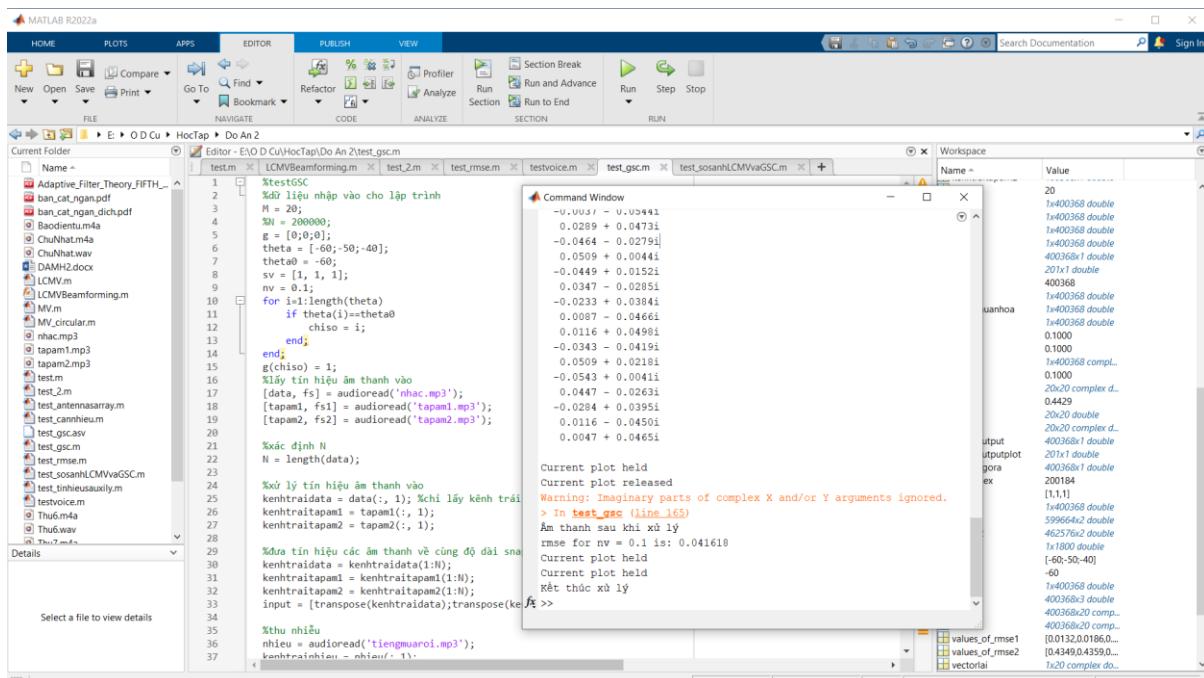
- d) So sánh giữa tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra bộ lọc:



Hình 23. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra.

Nhân xét: Tương tự giải thuật LCMV, đồ thị biểu diễn tín hiệu tại ngõ vào (màu đỏ), tín hiệu mong muốn (màu xanh) và tín hiệu tại ngõ ra (màu lục). Có thể thấy 2 đường màu xanh và màu lục rất tương đồng nhau với sai số rất nhỏ, cho thấy tín hiệu tại ngõ ra tương đồng với tín hiệu mong muốn.

e) Root Mean-square Error:



Hình 24. RMSE của giải thuật.

$RMSE = 0.041618$.

Nhận xét: $RMSE$ nhỏ cho thấy tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

II.1.3. Nhận xét chung về hai giải thuật:

Với các thông số ngõ vào tối ưu ($M = 20, nv = 0.1$), cả hai giải thuật đều rất hiệu quả trong việc xử lý nhiễu và can nhiễu (quan sát đồ thị đồ thị tuyến tính), tín hiệu tại ngõ ra của bộ lọc tương đồng với tín hiệu mong muốn thu được (quan sát đồ thị so sánh tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra; $RMSE$ cả 2 giải thuật đều rất nhỏ).

II.2. Mở rộng “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều và cải tiến với mạng Neural:

Bài toán mô phỏng giải thuật:

Tương tự như với bài toán của GSC cho dãy an-ten một chiều, tuy nhiên, để mô phỏng thể hiện tính mở rộng với dãy an-ten hai chiều, góc phương vị và góc ngang của các tín hiệu đến có sự thay đổi: $(\varphi_i, \theta_i) = (-60^\circ, 30^\circ), (-150^\circ, 50^\circ), (120^\circ, 80^\circ); i = 0, 1, 2$ lần lượt là tín hiệu mong muốn và 2 can nhiễu.

II.2.1. Mở rộng “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều:

II.2.1.1. Lời giải để thực hiện mô phỏng:

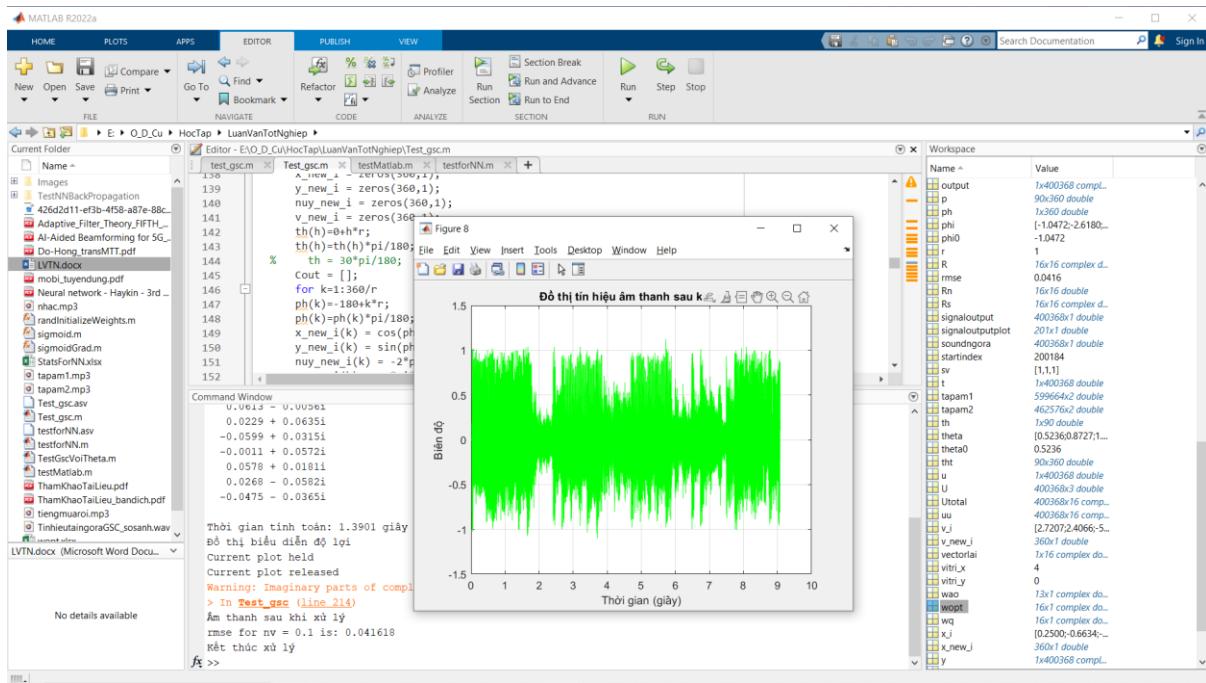
Hướng và tần số không gian của các tín hiệu tới có thể được biểu diễn theo công thức (3.1) và (3.2), đồng thời, tọa độ của các phần tử trong dãy an-ten 2 chiều bây giờ được biểu diễn như hình , ta có thể viết lại công thức vector lái cho mỗi tín hiệu tới như công thức (3.3).

Thực hiện vẽ “Đồ thị tuyến tính” của phần giải thuật và “Biểu diễn đường đồng mức” của “Đồ thị tuyến tính” để dễ dàng quan sát kết quả.

II.2.1.2. Thực hiện mô phỏng trên MATLAB:

❖ Kết quả mô phỏng:

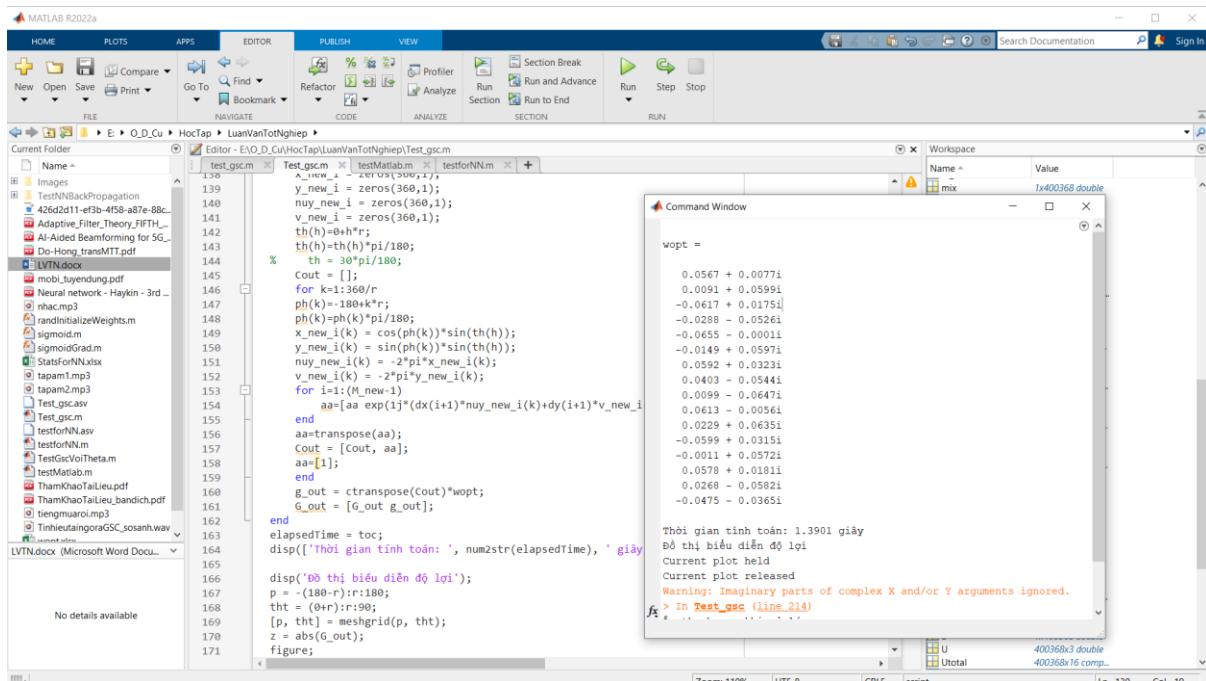
- Tín hiệu sau khi đã xử lý tại ngõ ra bộ lọc:



Hình 25. Tín hiệu tại ngõ ra của bộ lọc cho dãy an-ten 2 chiều.

Nhân xét: Tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

b) Trọng số tối ưu:

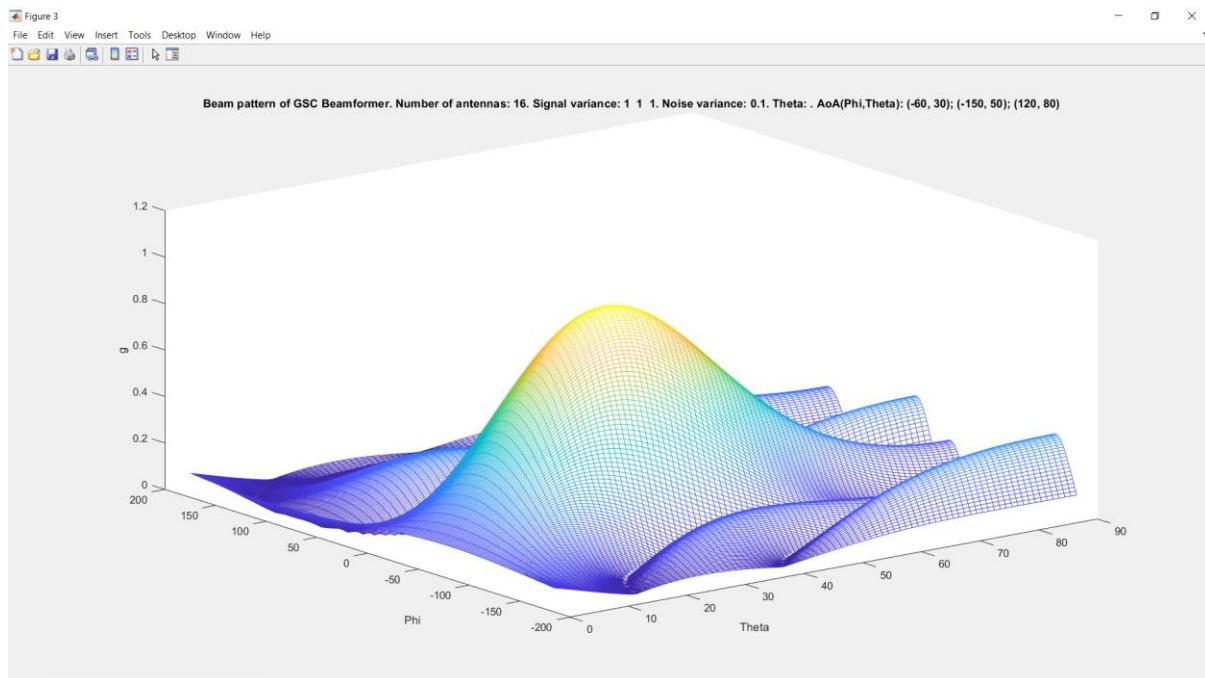


Hình 26. Trọng số tối ưu của bộ lọc cho dãy an-ten 2 chiều.

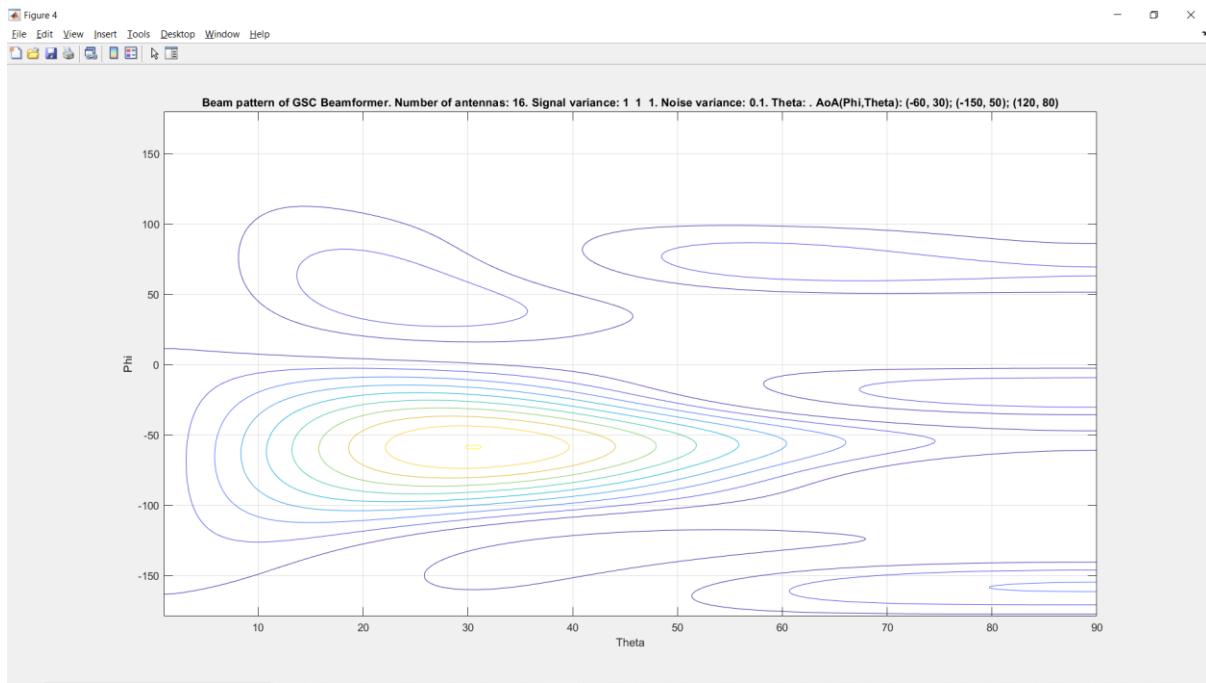
Trọng số tối ưu:

$$\mathbf{w}_{opt} = [0.0567 + 0.0077i, \quad 0.0091 + 0.0599i, \\ -0.0617 + 0.0175i, \quad -0.0288 - 0.0526i, \\ -0.0655 - 0.0001i, \quad -0.0149 + 0.0597i, \\ 0.0592 + 0.0323i, \quad 0.0403 - 0.0544i, \\ 0.0099 - 0.0647i, \quad 0.0613 - 0.0056i, \\ 0.0229 + 0.0635i, \quad -0.0599 + 0.0315i, \\ -0.0011 + 0.0572i, \quad 0.0578 + 0.0181i, \\ 0.0268 - 0.0582i, \quad -0.0475 - 0.0365i]^T$$

c) Đồ thị tuyen tính:



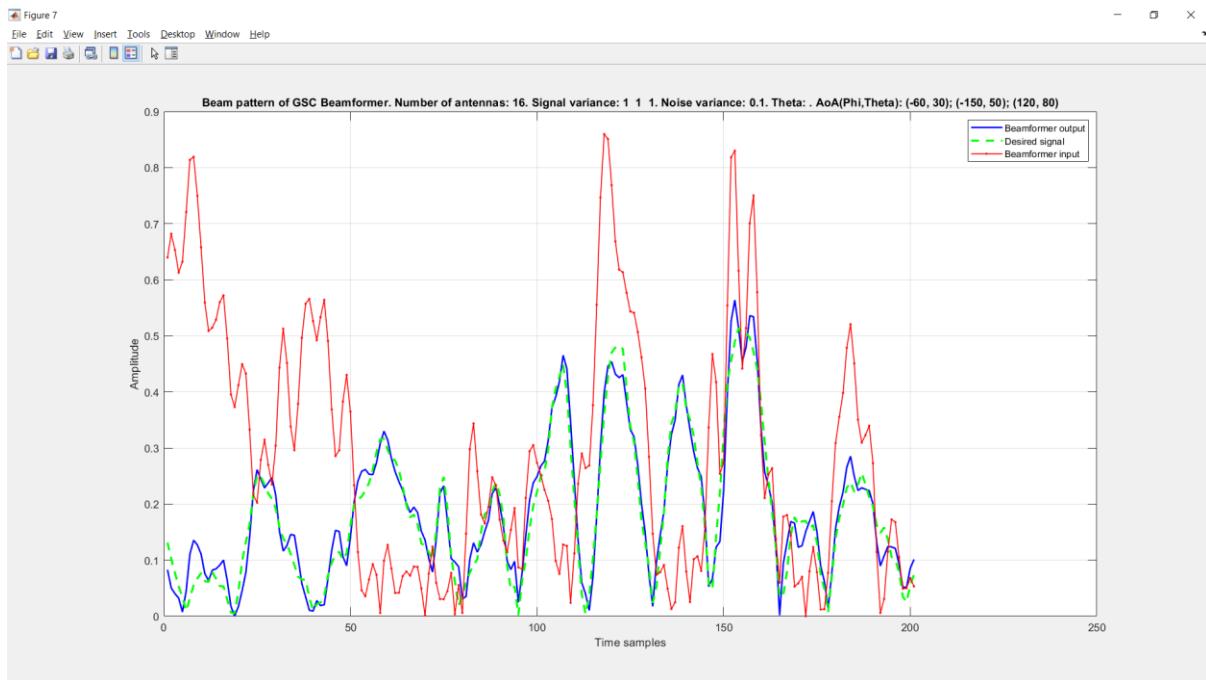
Hình 27. Đồ thị tuyen tính của “Bộ triệt các búp bên tổng quát” cho dãy an-ten 2 chiều.



Hình 28. Biểu diễn đồ thị tần số dưới dạng các đường đồng mức.

Nhân xét: Tương tự với giải thuật trong trường hợp dãy an-ten 1 chiều, tín hiệu mong muốn tương ứng với góc tới $(-60^\circ, 30^\circ)$ được tối ưu, với hướng búp sóng chính cực đại, đồng thời can nhiễu tại góc tới $(-150^\circ, 50^\circ)$ và $(120^\circ, 80^\circ)$ được triệt tiêu là các điểm null (điểm không), chứng tỏ giải thuật cho khả năng triệt can nhiễu và tối ưu tín hiệu mong muốn thu được khá tốt.

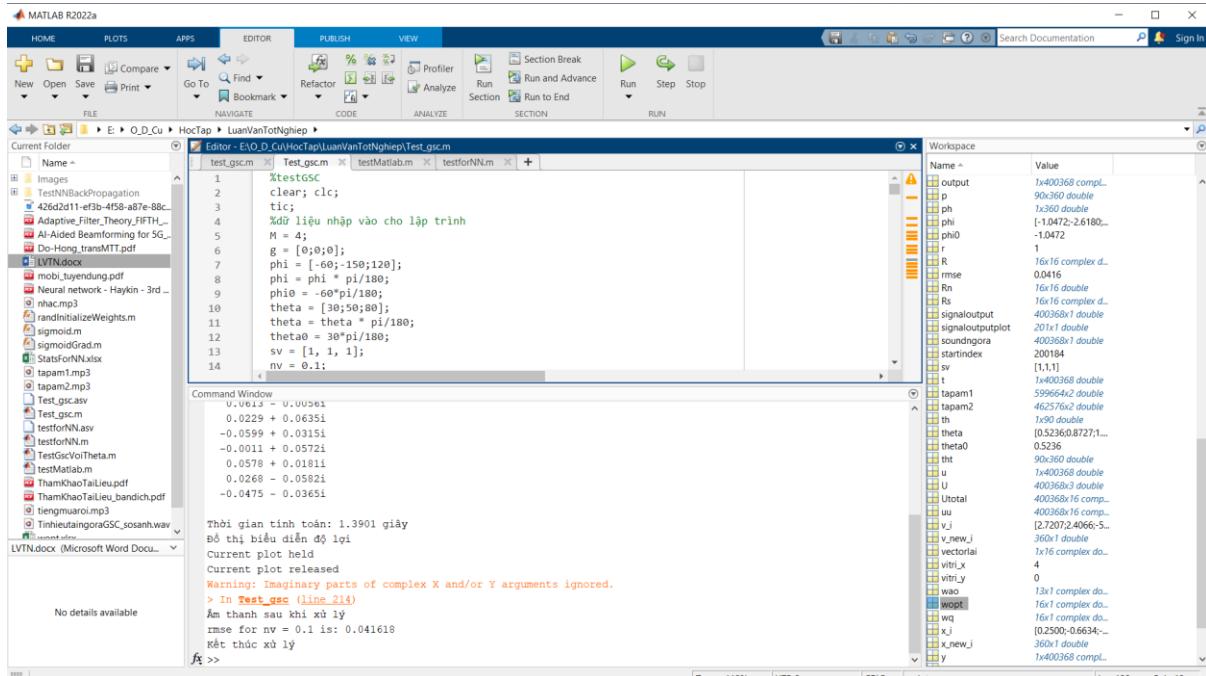
- d) So sánh giữa tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra bộ lọc:



Hình 29. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra.

Nhân xét: Tương tự giải thuật GSC với dãy an-ten 1 chiều, đồ thị biểu diễn tín hiệu tại ngõ vào (màu đỏ), tín hiệu mong muốn (màu xanh) và tín hiệu tại ngõ ra (màu lục). Có thể thấy 2 đường màu xanh và màu lục rất tương đồng nhau với sai số rất nhỏ, cho thấy tín hiệu tại ngõ ra tương đồng với tín hiệu mong muốn.

e) Root Mean-square Error:



Hình 30. RMSE của giải thuật.

$RMSE = 0.041618$.

Nhân xét: RMSE cũng tương đồng với GSC cho dãy an-ten 1 chiều, cho thấy tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

II.2.2. Cải tiến “Bộ triệt các búp bên tổng quát” cho dãy an-ten hai chiều với mạng Neural:

II.2.2.1. Lời giải để thực hiện mô phỏng:

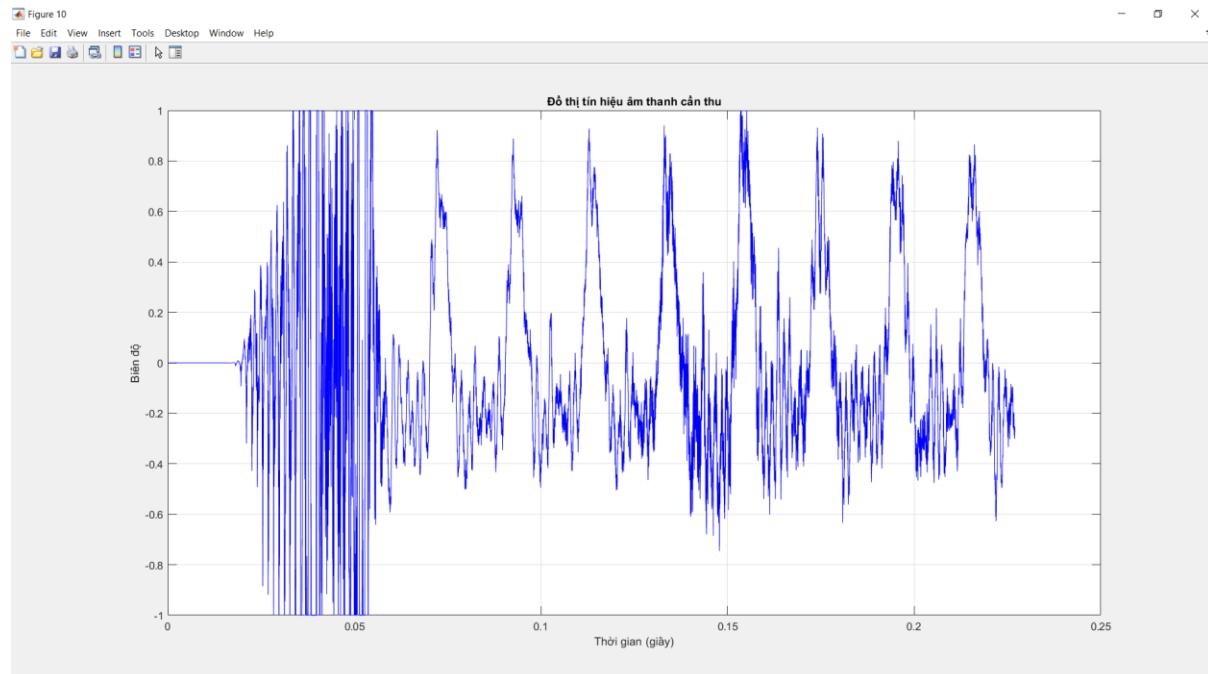
Thực hiện theo sơ đồ khối như hình 10. Với mạng Neural, ta có thể dễ dàng thiết lập số điểm tại ngõ vào và ngõ ra của mạng, tuy nhiên, để chọn được số điểm lớp ẩn và hệ số học (learning rate), cần tiến hành khảo sát rồi chọn ra kết quả tốt nhất để tiến hành huấn luyện, kiểm tra cho mô hình và thực hiện mô phỏng giải thuật như các giải thuật trước.

II.2.2.2. Thực hiện mô phỏng trên MATLAB:

❖ Ngõ vào của bộ lọc:

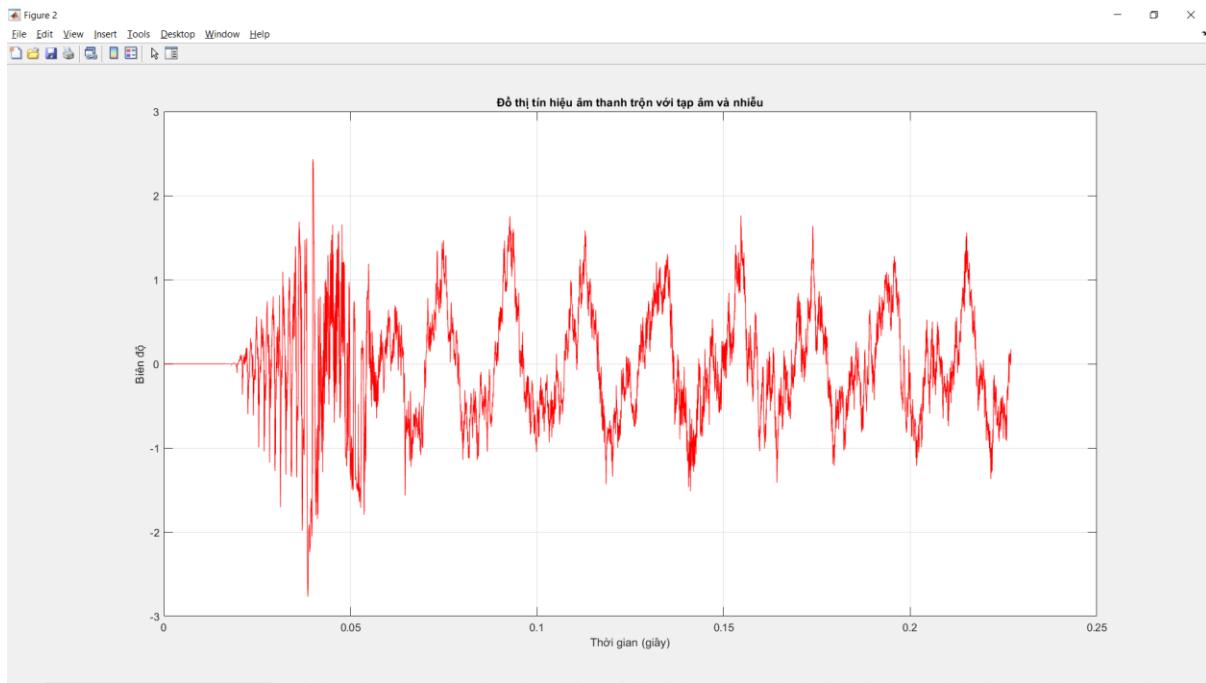
- Tín hiệu gốc cần thu:

Với hạn chế về phần cứng của thiết bị nghiên cứu, tín hiệu được mô phỏng được cắt ngắn từ tín hiệu gốc được mô phỏng ở các phần trước và có dạng:



Hình 31. Tín hiệu gốc cần thu.

- Tín hiệu trộn tại ngõ vào của bộ lọc gồm tín hiệu mong muốn và 2 can nhiễu:

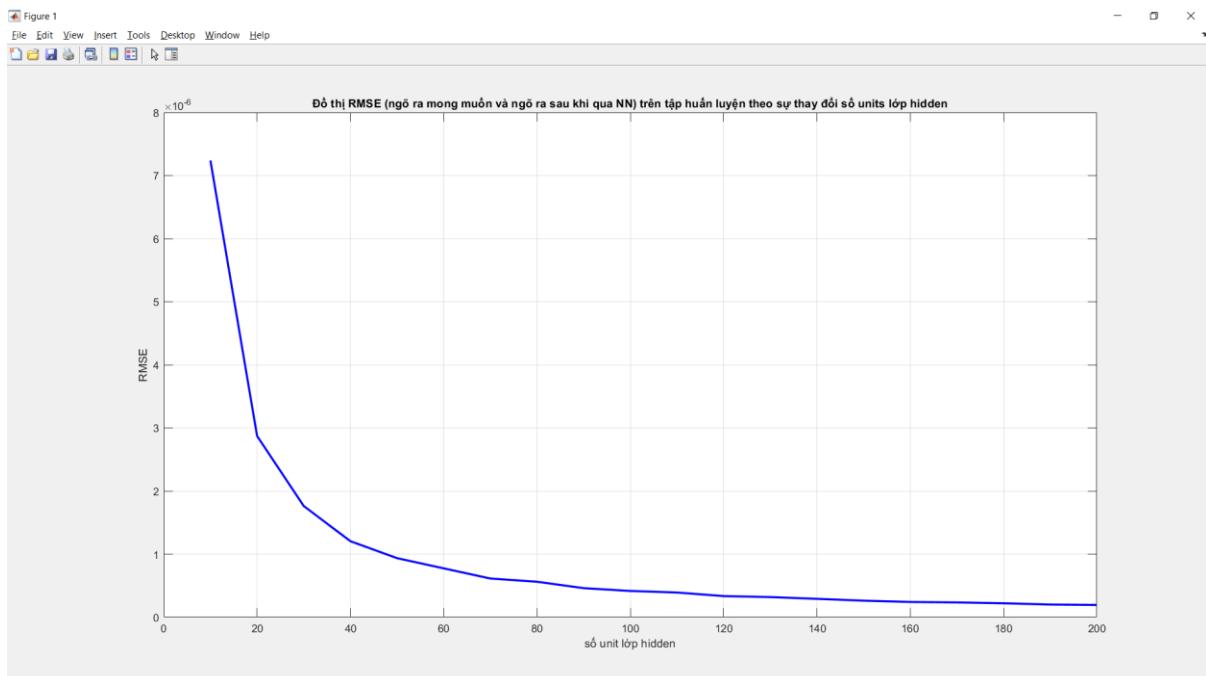


Hình 32. Tín hiệu trộn tại ngõ vào của bộ lọc.

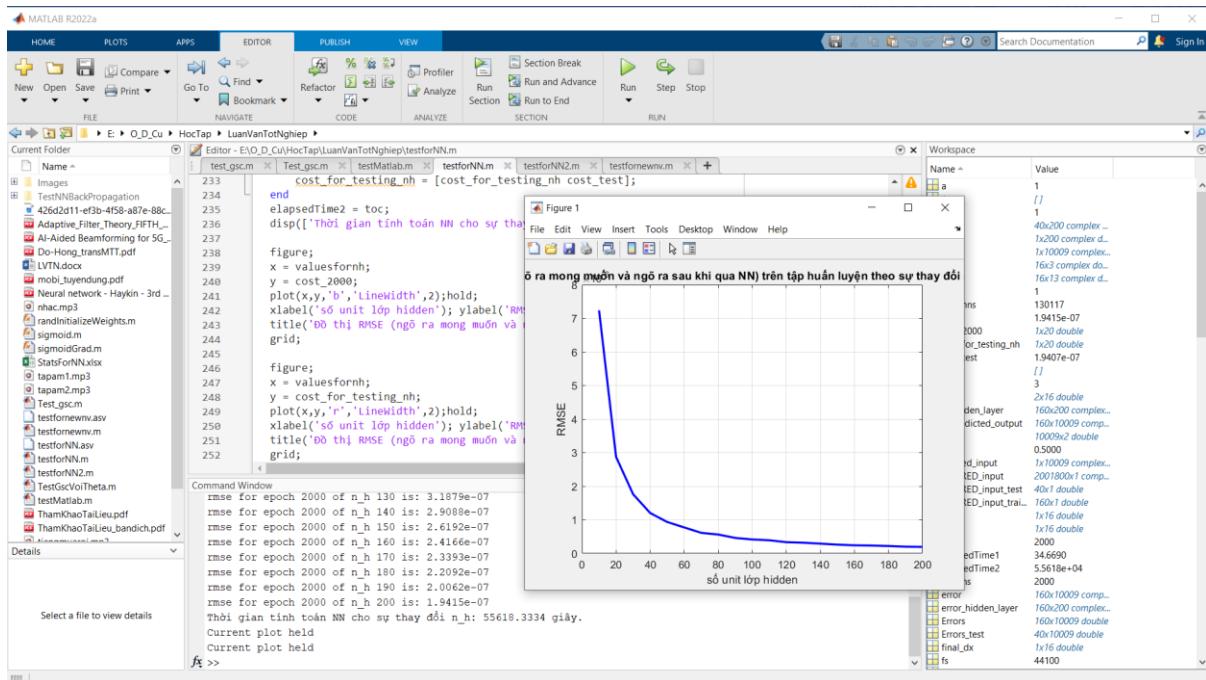
❖ **Kết quả mô phỏng:**

a) Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural:

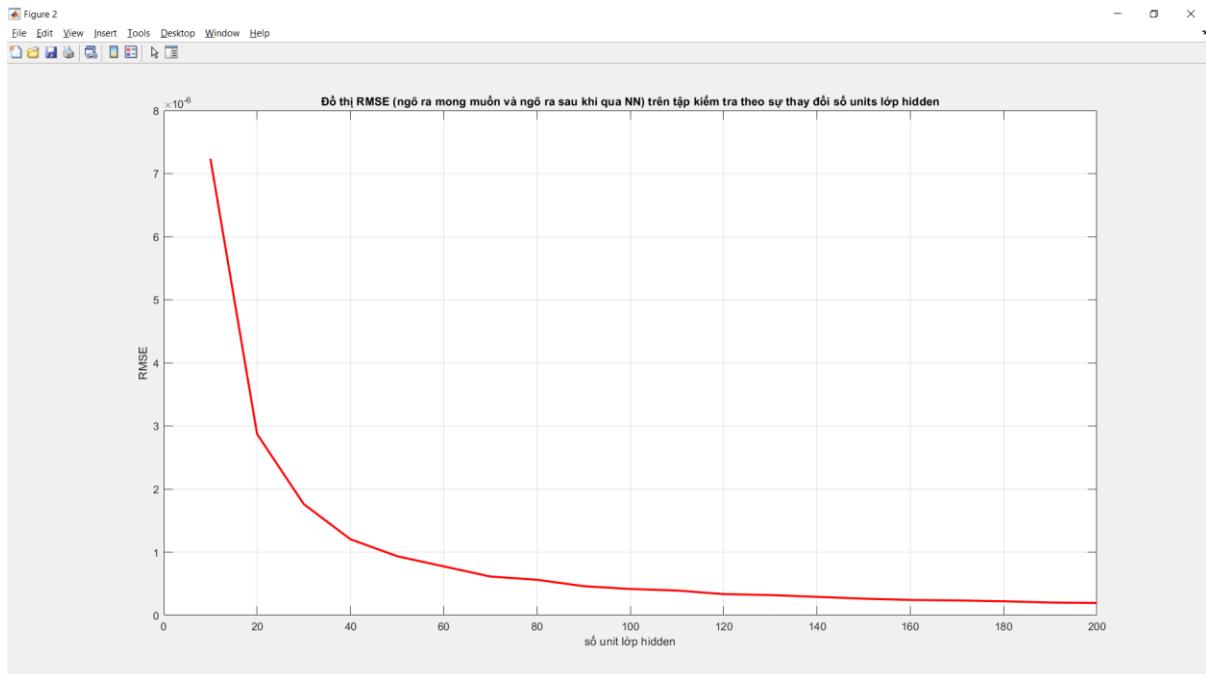
Tiến hành khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập dữ liệu huấn luyện (training) với cùng một hệ số học:



Hình 33. Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập huấn luyện.



Hình 34. Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập huấn luyện.

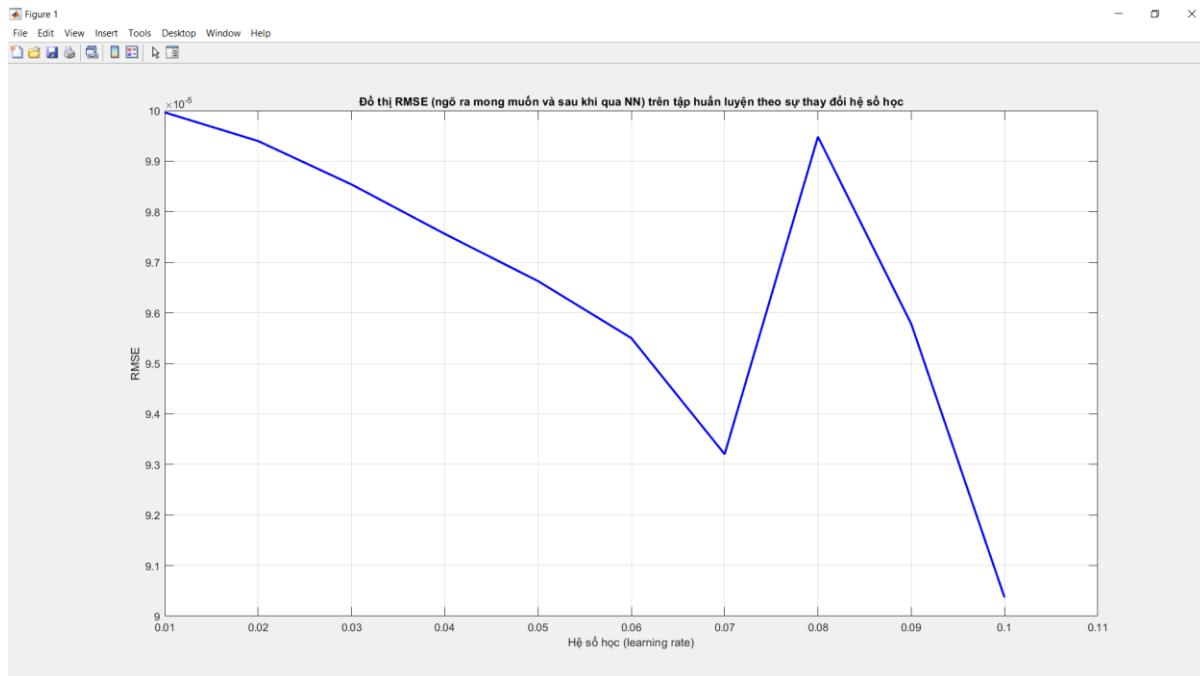


Hình 35. Khảo sát số điểm lớp ẩn tối ưu cho mạng Neural trên tập kiểm tra.

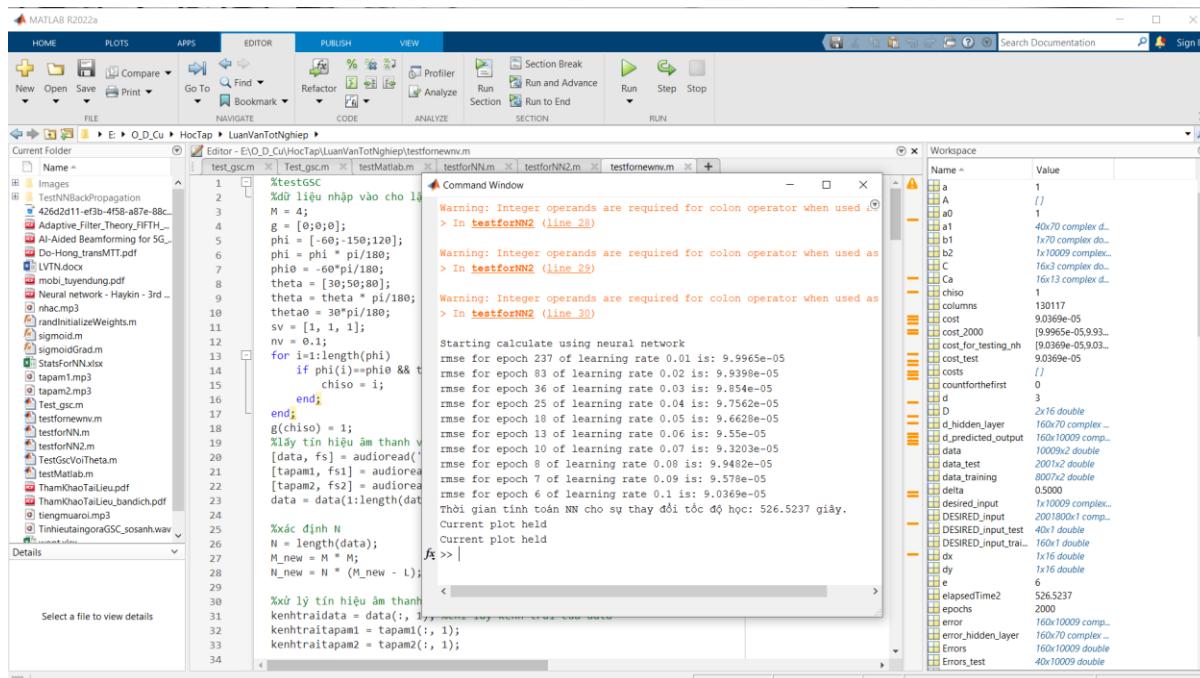
Nhận xét: Để dàng nhận thấy, khi tăng số điểm lớp ẩn, RMSE có xu hướng giảm. Khi số điểm lớp ẩn lớn hơn 50, RMSE gần như là xấp xỉ nhau, rất nhỏ. Thông thường, khi hàm mất mát (cost function), trong ứng dụng này là RMSE giữa ngõ ra mong muốn và ngõ ra sau khi qua mạng Neural, đạt đến giá trị rất nhỏ hơn 10^{-6} được xem là đã hội tụ. Do đó, chọn số điểm lớp ẩn là 40.

b) Khảo sát hệ số học (learning rate) tối ưu cho mạng Neural:

Tiến hành khảo sát hệ số học tối ưu cho mạng Neural trên tập dữ liệu huấn luyện (training) trong khoảng giá trị $0.01 \div 0.1$:



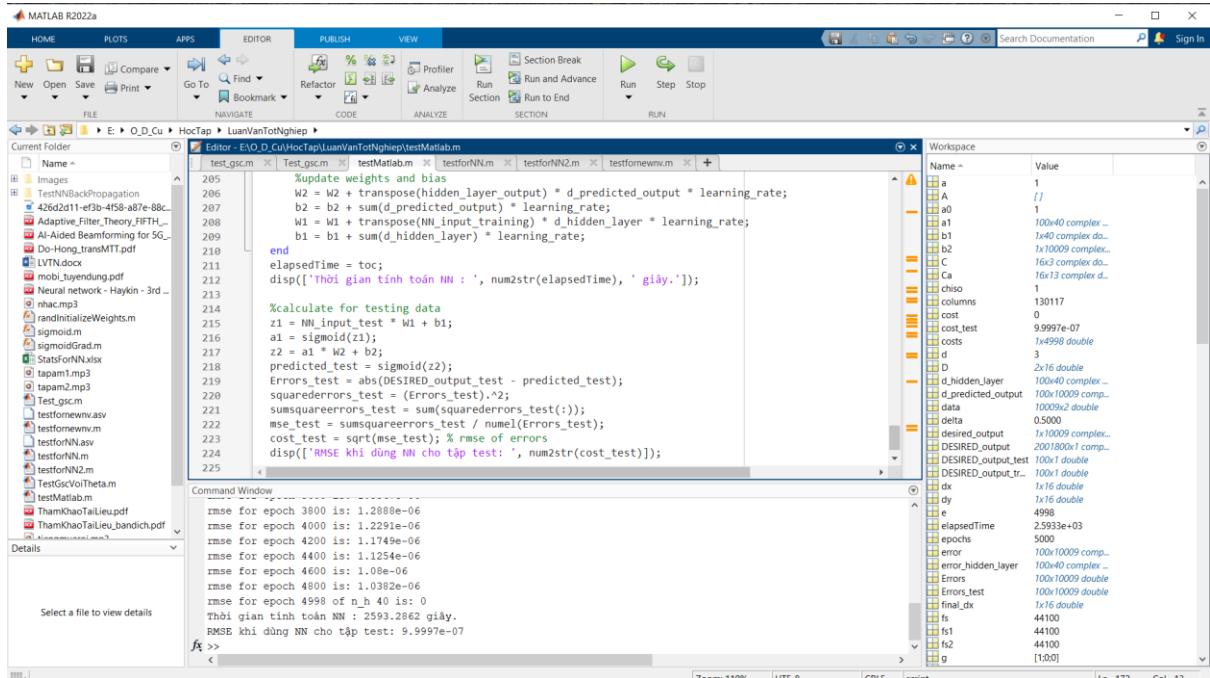
Hình 36. Khảo sát hệ số học tối ưu cho mạng Neural.



Hình 37. Khảo sát hệ số học tối ưu cho mạng Neural.

Nhân xét: Dễ dàng nhận thấy, đối với mô hình mạng Neural này, hệ số học (learning rate) $\alpha = 0.1$ cho kết quả tốt nhất. Đặc biệt, khi tăng hệ số học, tốc độ hội tụ càng nhanh khi số vòng lặp để hồi tu càng giảm.

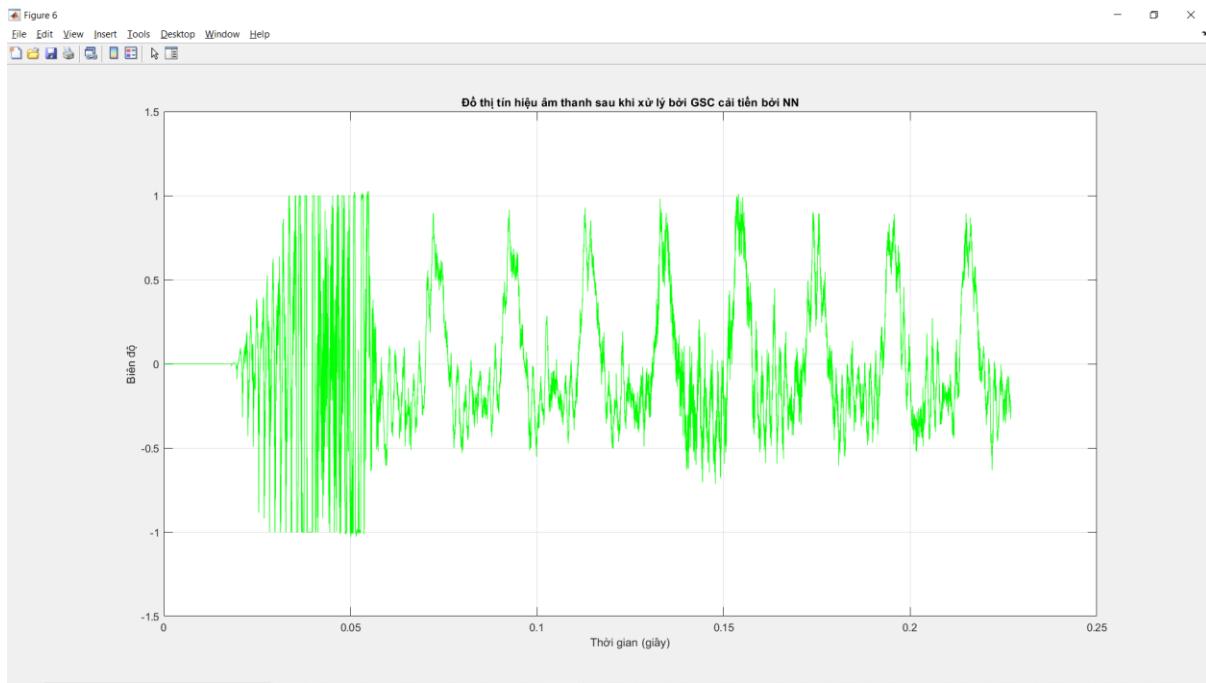
- c) Thực hiện huấn luyện cho mô hình và kiểm tra trên tập dữ liệu huấn luyện và kiểm tra được xây dựng:



Hình 38. Huấn luyện và kiểm tra mô hình trên tập dữ liệu huấn luyện và kiểm tra đã được xây dựng.

Nhân xét: Quá trình huấn luyện kết thúc khi hàm mất mát hội tụ sau 4998 vòng lặp với thời gian hơn 40 phút. Đối với dữ liệu trên tập kiểm tra, hàm mất mát cũng cho giá trị rất nhỏ với RMSE = $9.9997e - 07$ chứng tỏ mô hình mạng Neural được xây dựng khá tốt.

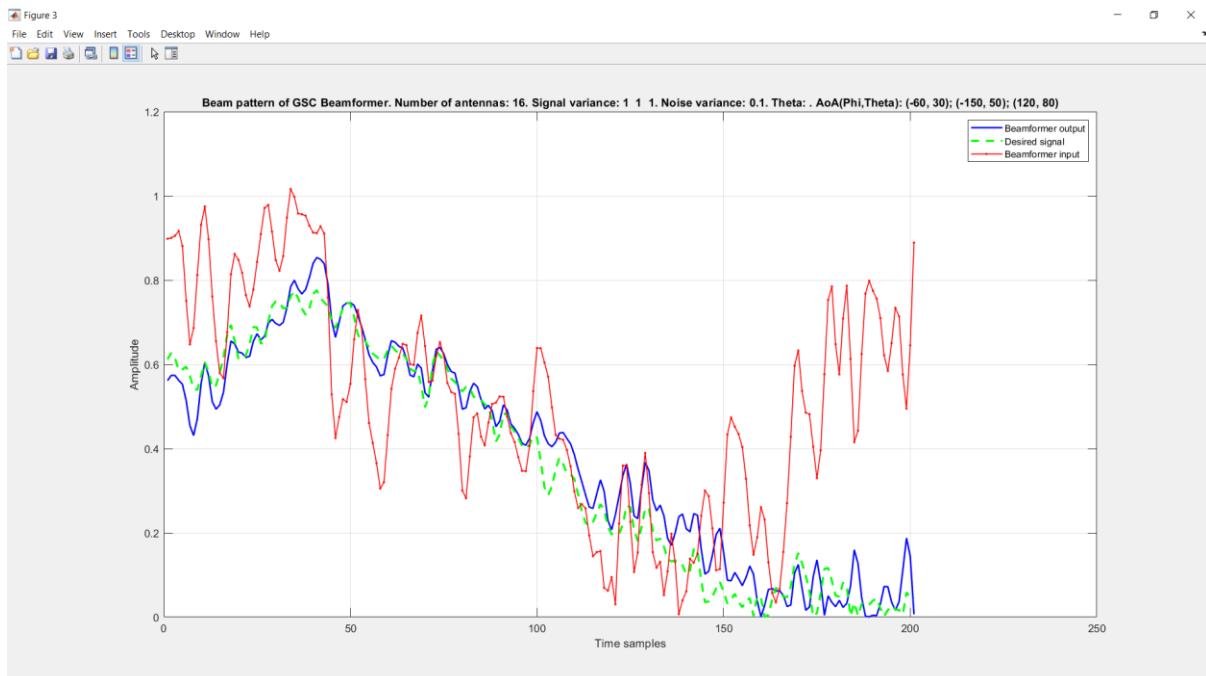
- d) Tín hiệu tại ngõ ra sau khi qua bộ GSC cài tiến với mạng Neural:



Hình 39. Tín hiệu tại ngõ ra của bộ lọc được cải tiến với mạng Neural.

Nhân xét: Tín hiệu tại ngõ ra sau khi qua xử lý tương đồng với tín hiệu mong muốn.

- e) So sánh giữa tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra của bộ lọc:



Hình 40. So sánh giữa tín hiệu tại ngõ vào, tín hiệu mong muốn và tín hiệu tại ngõ ra của bộ lọc.

II.2.3. Nhận xét chung về giải thuật “Bộ triệt các búp bên tổng quát” được mở rộng và cải tiến bởi mạng Neural:

Mở rộng cho dãy an-ten 2 chiều, giải thuật GSC vẫn cho thấy khả năng triệt can nhiễu và tối ưu tín hiệu mong muốn rất tốt như trường hợp dãy an-ten 1 chiều. Việc cải tiến tính toán vector trọng số tối ưu cho giải thuật với mạng Neural cũng cho kết quả rất tốt, với thời gian tính toán rất nhanh và độ chính xác tương tự như giải thuật GSC truyền thống.

CHƯƠNG III: KẾT QUẢ TỔNG QUÁT CỦA HAI GIẢI THUẬT

Cả hai giải thuật đều cho ra kết quả tương đối tốt dù khoảng cách góc của các tín hiệu đến khá là gần nhau. Do đó, để đánh giá tổng quát được từng giải thuật cũng như có sự so sánh chính xác về hiệu quả của 2 giải thuật, dựa trên các thông số của bài toán cũ, đồ án đã nghiên cứu thêm các trường hợp của từng giải thuật: thay đổi số an-ten M, giảm khoảng cách góc giữa các tín hiệu, thay đổi phương sai nhiễu.

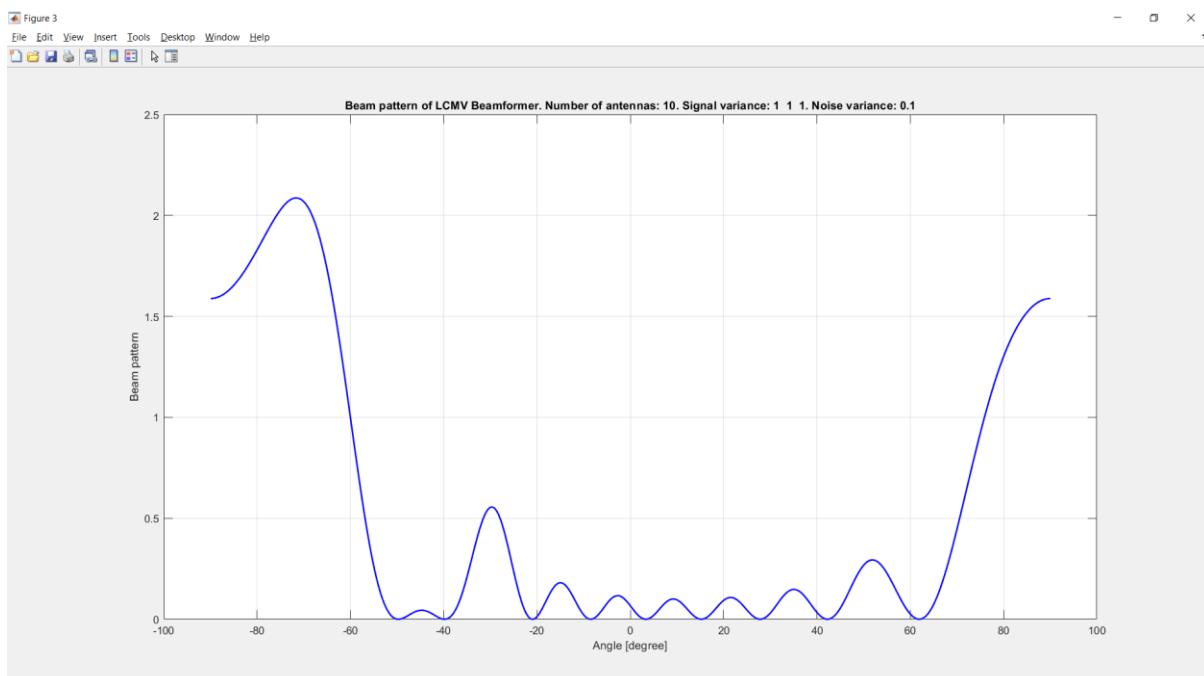
III.1. Tối thiểu phương sai có ràng buộc tuyến tính (LCMV):

III.1.1. Thay đổi số phần tử an-ten M:

Để thuận tiện cho việc quan sát sự hiệu quả của giải thuật, ta cần chú ý vào tham số Root Mean-square Error (RMSE) và quan sát đồ thị tuyến tính cùng đồ thị so sánh tín hiệu mong muốn và ngõ ra của các trường hợp.

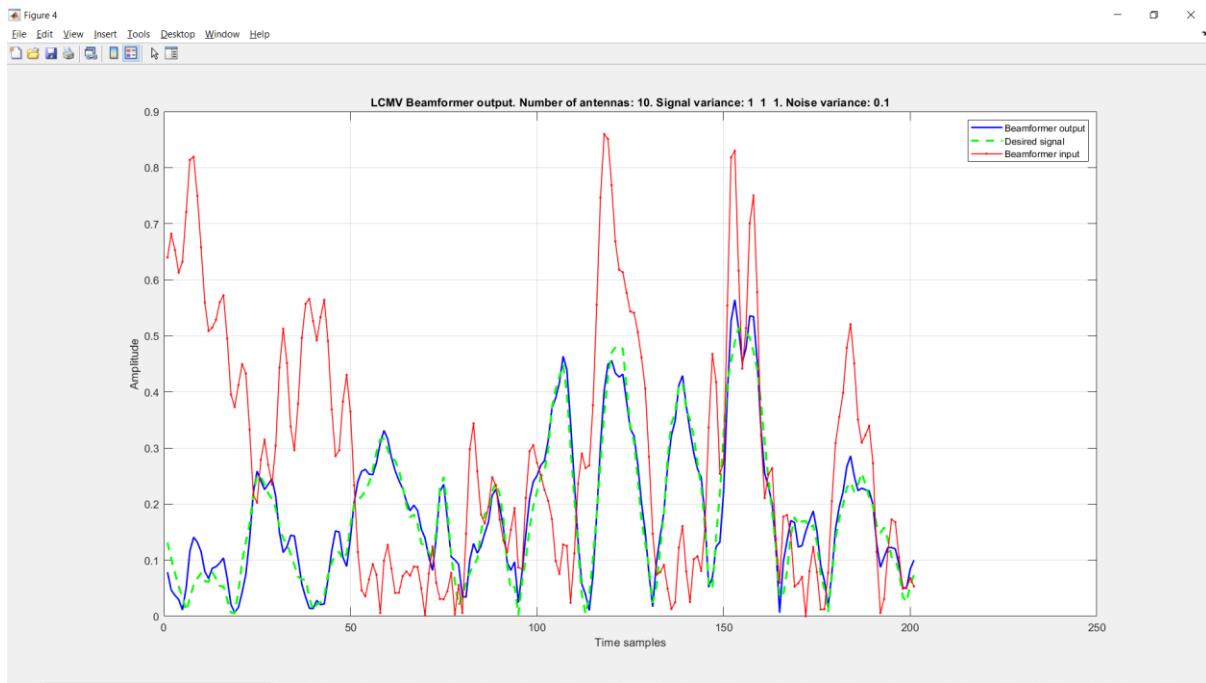
Thay đổi số phần tử $M = 10, 15, 17, 18, 20$ rồi quan sát kết quả:

- ❖ Đồ thị tuyến tính, đồ thị so sánh giữa các tín hiệu và RMSE tương ứng:
a) $M = 10$:



Hình 41. Đồ thị tuyến tính của giải thuật với $M = 10$.

Nhận xét: Với số lượng phần tử an-ten chưa đủ lớn, giải thuật chưa thể tối ưu tín hiệu mong muốn cần thu được tại ngõ ra, búp sóng chính không nằm ở góc của hướng tới tín hiệu mong muốn.

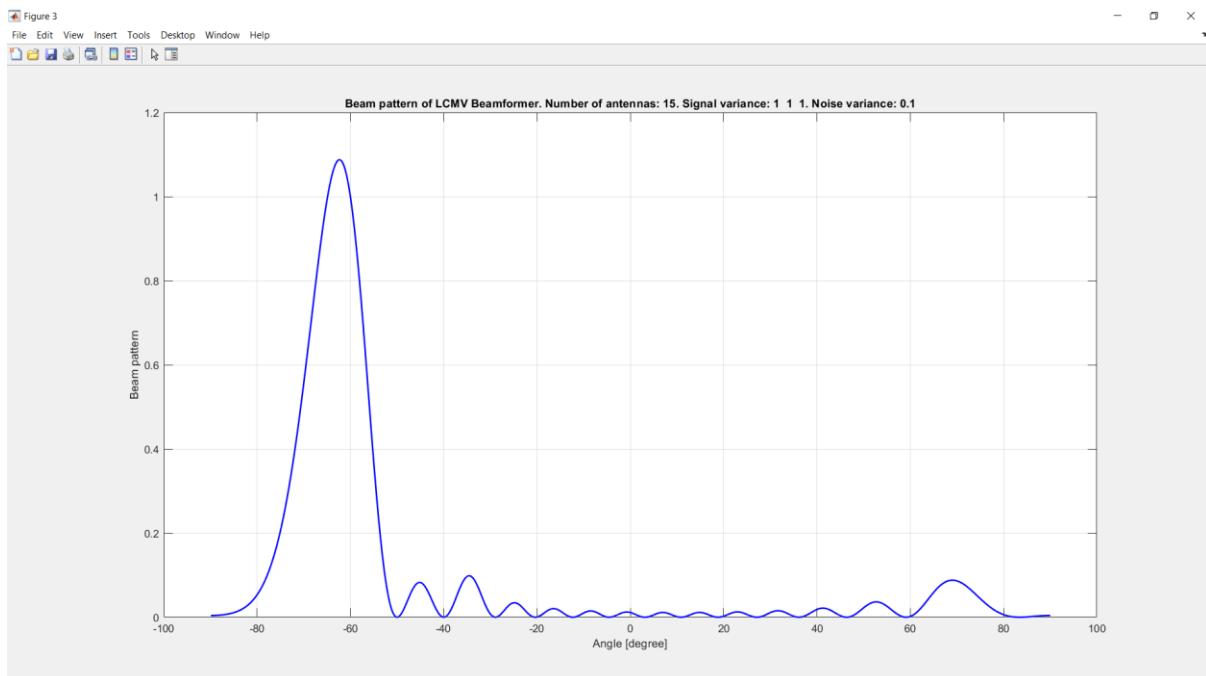


Hình 42. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 10$.

$RMSE = 0.043519$.

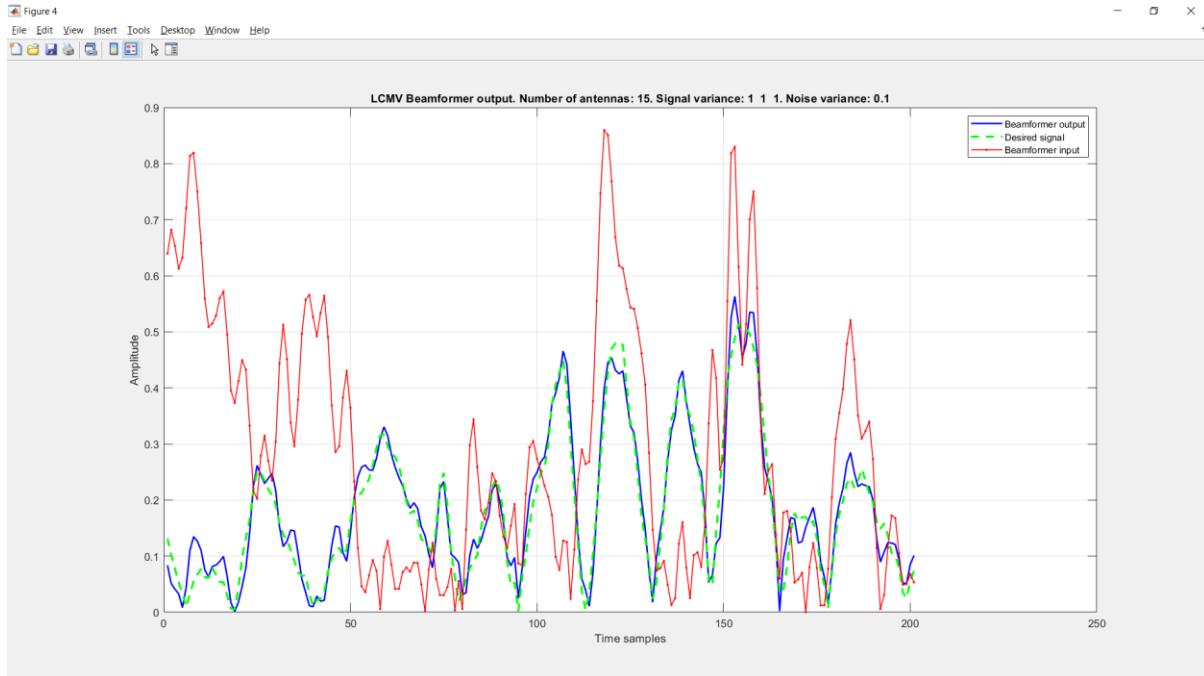
Nhận xét: Đồ thị vẫn cho thấy kết quả là tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra khá tương đồng nhau. Đồng thời, $RMSE$ nhỏ cho thấy tín hiệu tại ngõ ra sau khi qua xử lý cũng tương đồng với tín hiệu mong muốn.

b) $M = 15$:



Hình 43. Đồ thị tuyen tính của giải thuật LCMV với $M = 15$.

Nhân xét: Với số phần tử an-ten $M = 15$ là đủ lớn để giải thuật cho kết quả tối ưu tốt hơn, khi mà góc của búp sóng chính gần với góc hướng tới tín hiệu mong muốn.

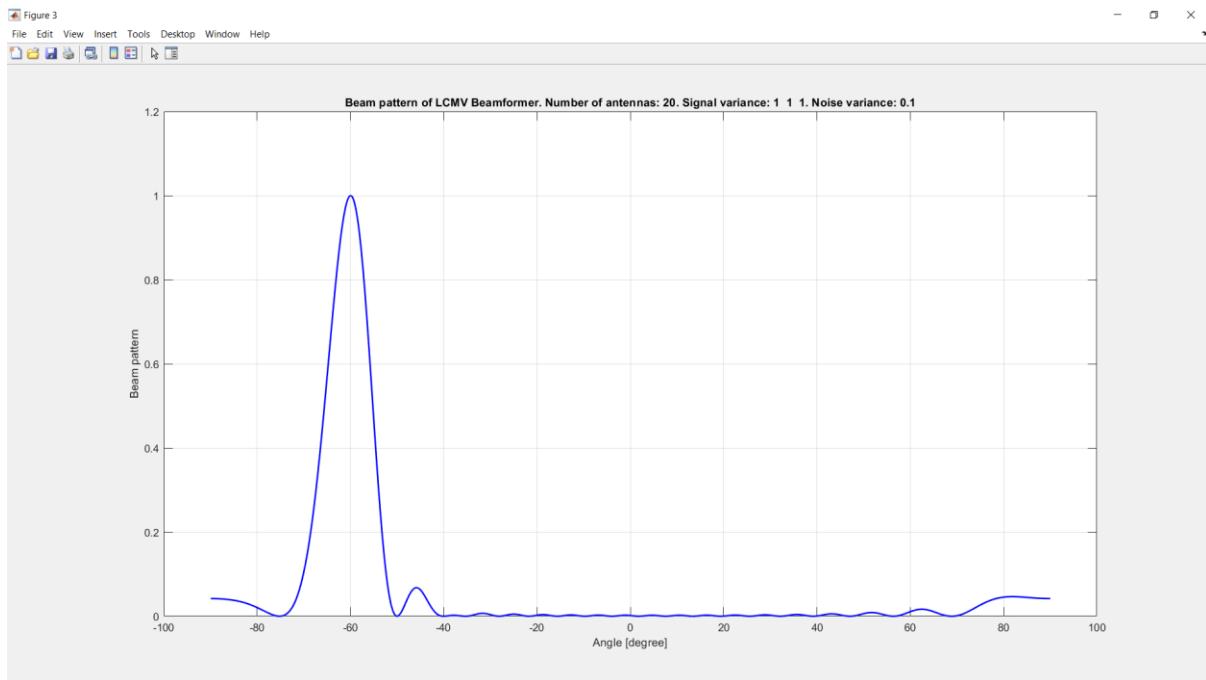


Hình 44. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 15$.

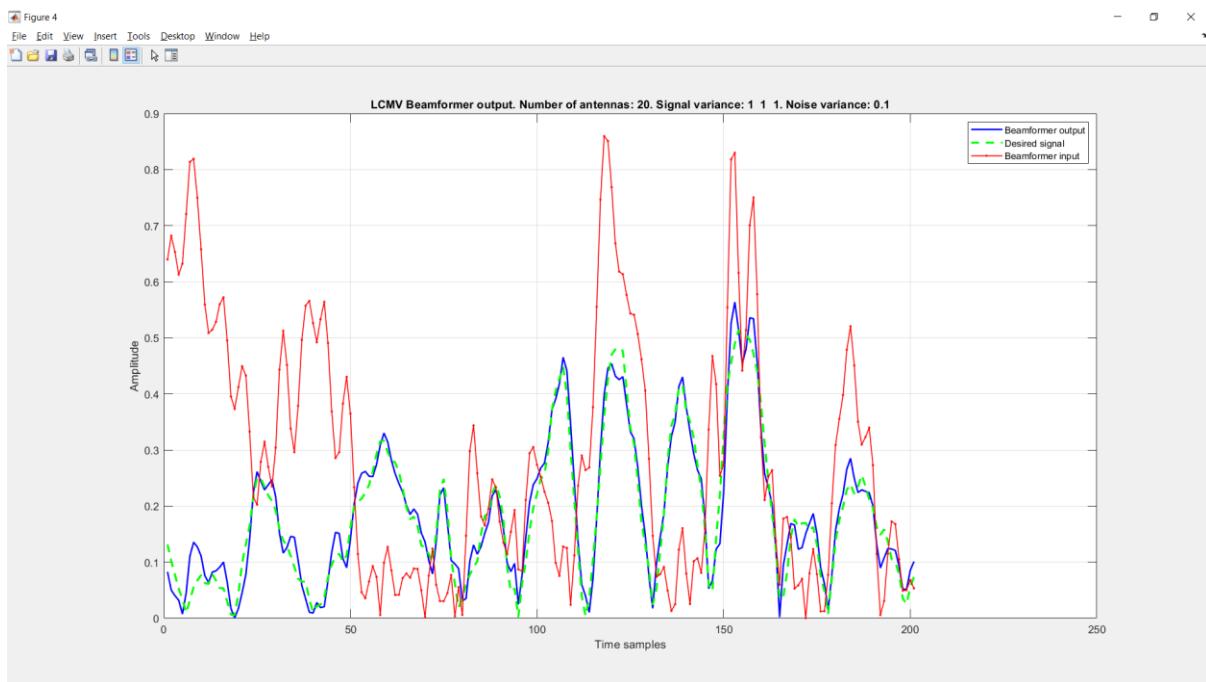
$RMSE = 0.04155$.

Nhân xét: Đồ thị và RMSE cũng cho thấy kết quả là tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra khá tương đồng nhau. RMSE cũng nhỏ hơn khi tăng số phần tử an-ten.

c) $M = 20$:



Hình 45. Đồ thị tuyển tính của giải thuật với $M = 20$.



Hình 46. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 20$.

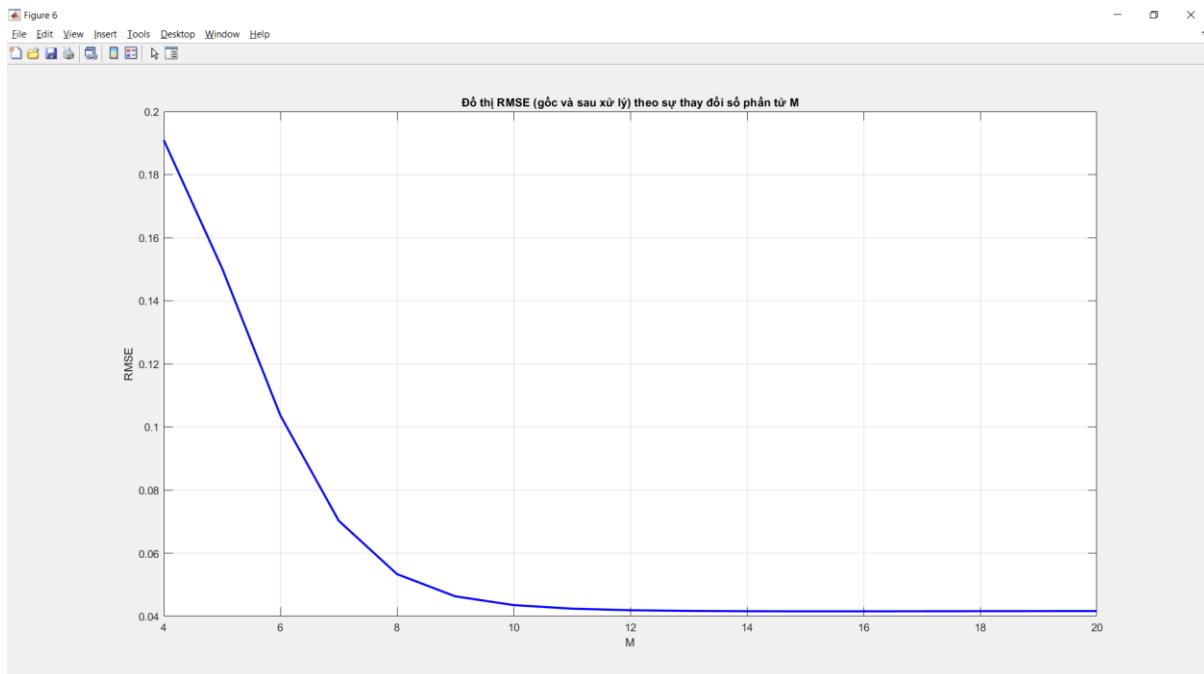
$RMSE = 0.041633$.

Nhận xét: Sau khi khảo sát thêm các giá trị $M = 17, 18$, kết quả tốt nhất là với $M = 20$, tín hiệu tại ngõ ra tương đồng với tín hiệu mong muốn hơn rất nhiều so với các kết quả trước. Búp sóng chính ở chính xác góc của tín hiệu mong muốn, đồng thời tại các góc của hướng tới có nhiều là các điểm null (điểm không), chứng tỏ khả năng triệt can nhiễu và tối ưu tín hiệu mong muốn của giải thuật với $M = 20$ là rất tốt.

d) Nhận xét chung về giải thuật:

Dựa vào sự thay đổi ở góc của búp sóng chính ở đồ thị tuyến tính, có thể thấy với số lượng phần tử M càng tăng thì góc của búp sóng chính càng gần với góc của hướng tới tín hiệu mong muốn hơn, và ở số phần tử $M = 20$ thì giải thuật đạt hiệu quả tốt nhất (đúng với hướng tới mong muốn là ở góc -60°).

- ❖ Sự thay đổi $RMSE$ theo sự thay đổi của số phần tử M giữa tín hiệu gốc và tín hiệu sau khi xử lý:



Hình 47. $RMSE$ giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi M .

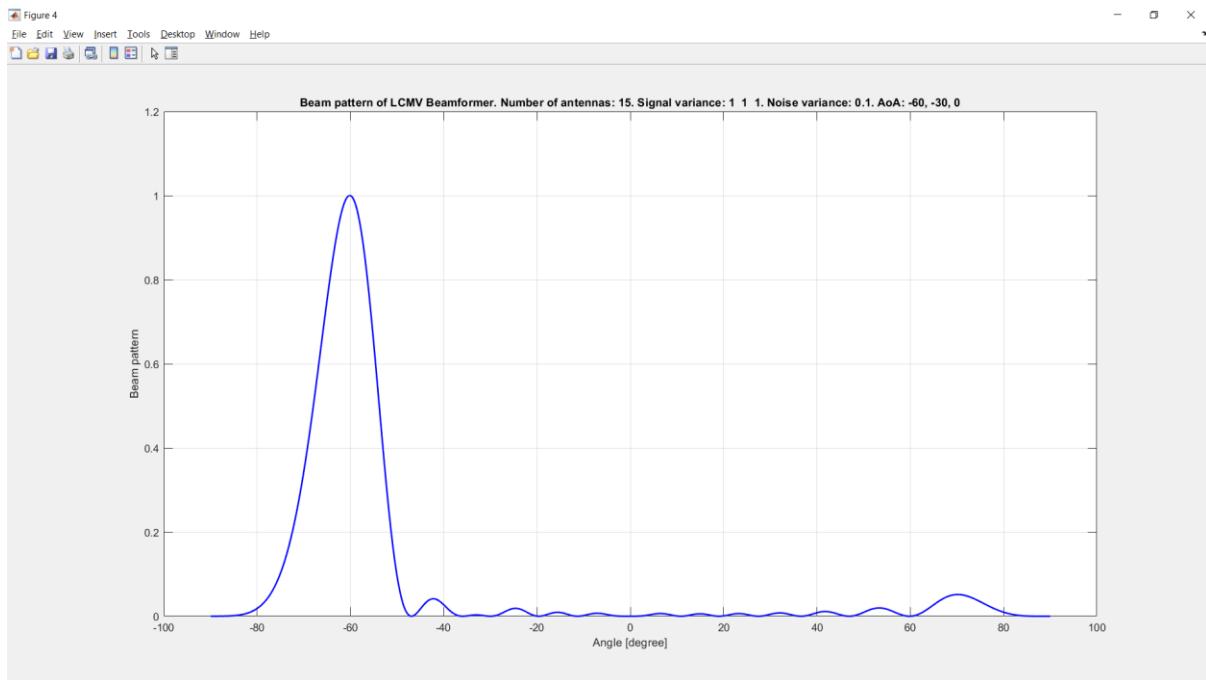
Nhận xét: Càng tăng số phần tử M thì $RMSE$ càng nhỏ dần, nghĩa là tín hiệu tại ngõ ra càng tương đồng với tín hiệu mong muốn. Đặc biệt, khi số phần tử M đủ lớn ($M = 15$) thì $RMSE$ hầu như không đổi, cho thấy giải thuật đạt tối ưu với số phần tử M đủ lớn đó.

III.1.2. Giảm khoảng cách góc giữa các tín hiệu mong muốn và can nhiễu:

Để thuận tiện cho việc quan sát sự hiệu quả của giải thuật, ta cần chú ý vào tham số Root Mean-square Error (RMSE) và quan sát đồ thị tuyến tính cùng đồ thị so sánh tín hiệu mong muốn và ngõ ra của các trường hợp.

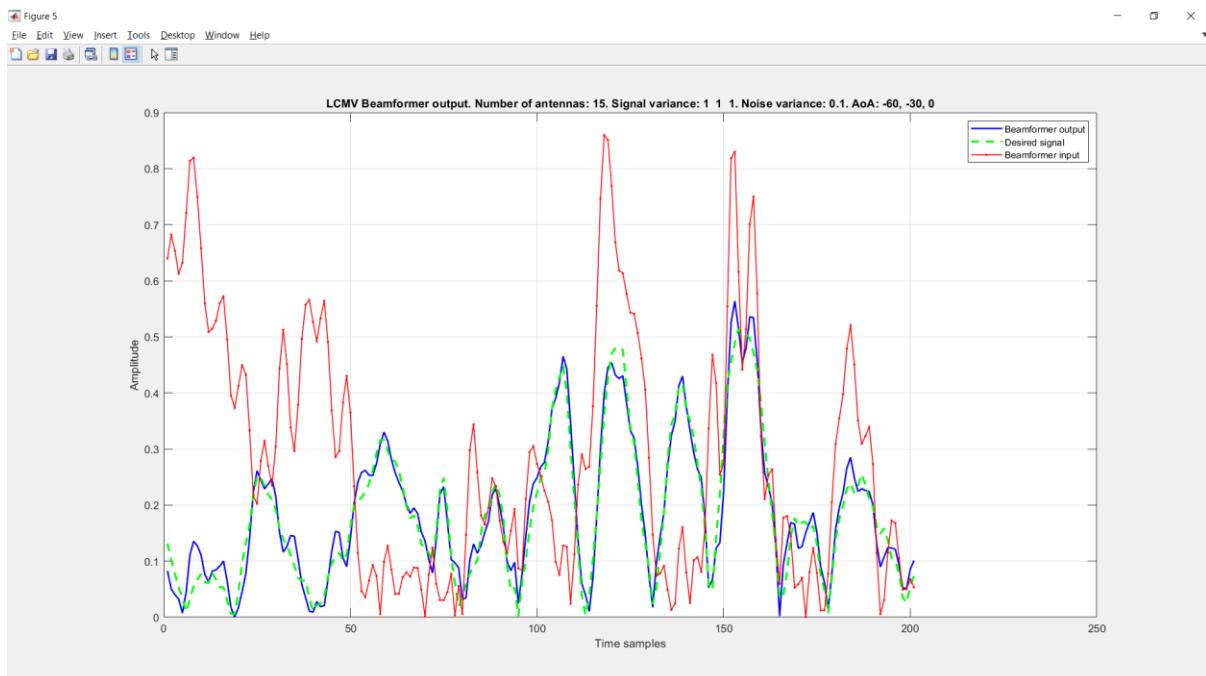
Thay đổi góc của các hướng tới $[\varphi_0, \varphi_1, \varphi_2]$ lần lượt là $[-60^\circ, -30^\circ, 0^\circ]$; $[-60^\circ, -35^\circ, -10^\circ]$; $[-60^\circ, -40^\circ, -20^\circ]$; $[-60^\circ, -45^\circ, -30^\circ]$; $[-60^\circ, -50^\circ, -40^\circ]$ rồi quan sát kết quả (sử dụng số phần tử $M = 15$ để dễ dàng quan sát sự khác biệt):

- ❖ Đồ thị tuyến tính, đồ thị so sánh giữa các tín hiệu và $RMSE$ tương ứng:
- $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$:



Hình 48. Đồ thị tần số tinh của $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$.

Nhận xét: Với khoảng cách góc giữa tín hiệu mong muốn và can nhiễu lớn như vậy, giải thuật có thể xử lý dễ dàng. Tại góc của hướng tới tín hiệu mong muốn là búp sóng chính, đồng thời tại góc của các hướng tới can nhiễu là các điểm null (điểm không), cho thấy giải thuật xử lý tín hiệu tốt với trường hợp này.

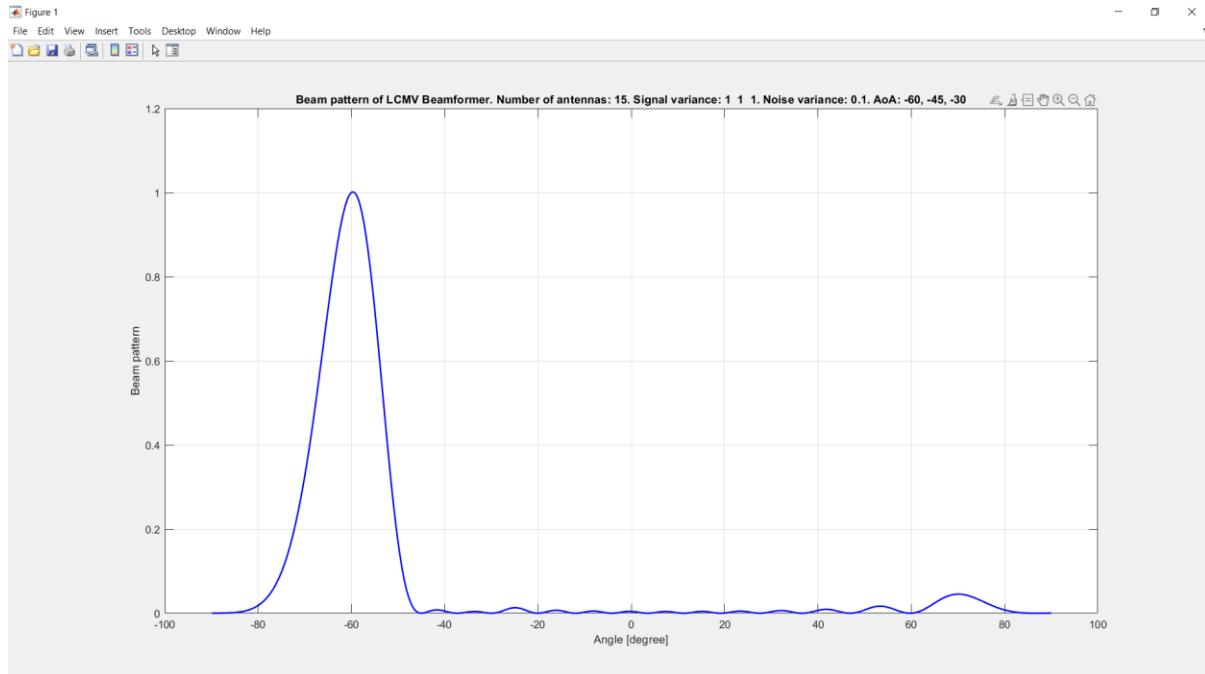


Hình 49. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$.

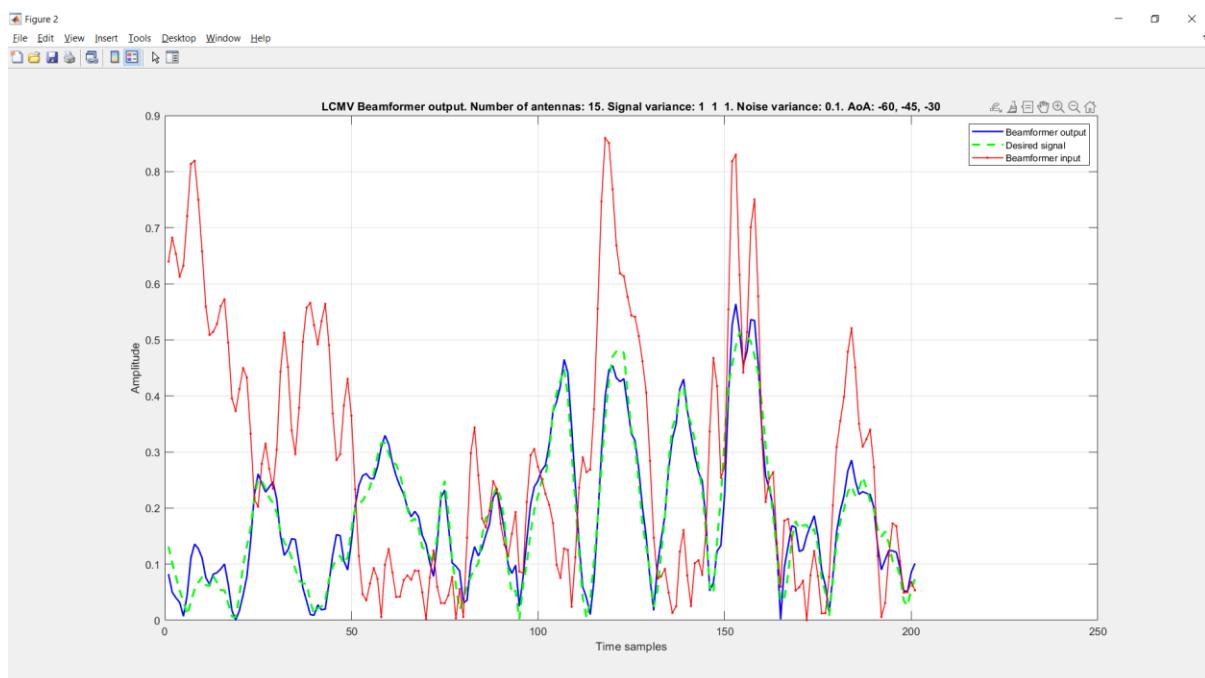
$RMSE = 0.041633$.

Nhân xét: Với khoảng cách góc giữa hướng tới của tín hiệu mong muốn và *can nhiễu* ở trường hợp này, tín hiệu tại *ngõ ra* (màu lục) rất tương đồng với tín hiệu mong muốn (màu xanh) và khác biệt rất nhiều với tín hiệu trộn tại *ngõ vào* (màu đỏ). RMSE cũng rất nhỏ cho thấy sự tương đồng giữa tín hiệu tại *ngõ ra* sau khi qua xử lý và tín hiệu mong muốn.

b) $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$:



Hình 50. Đồ thị tần số của $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$.

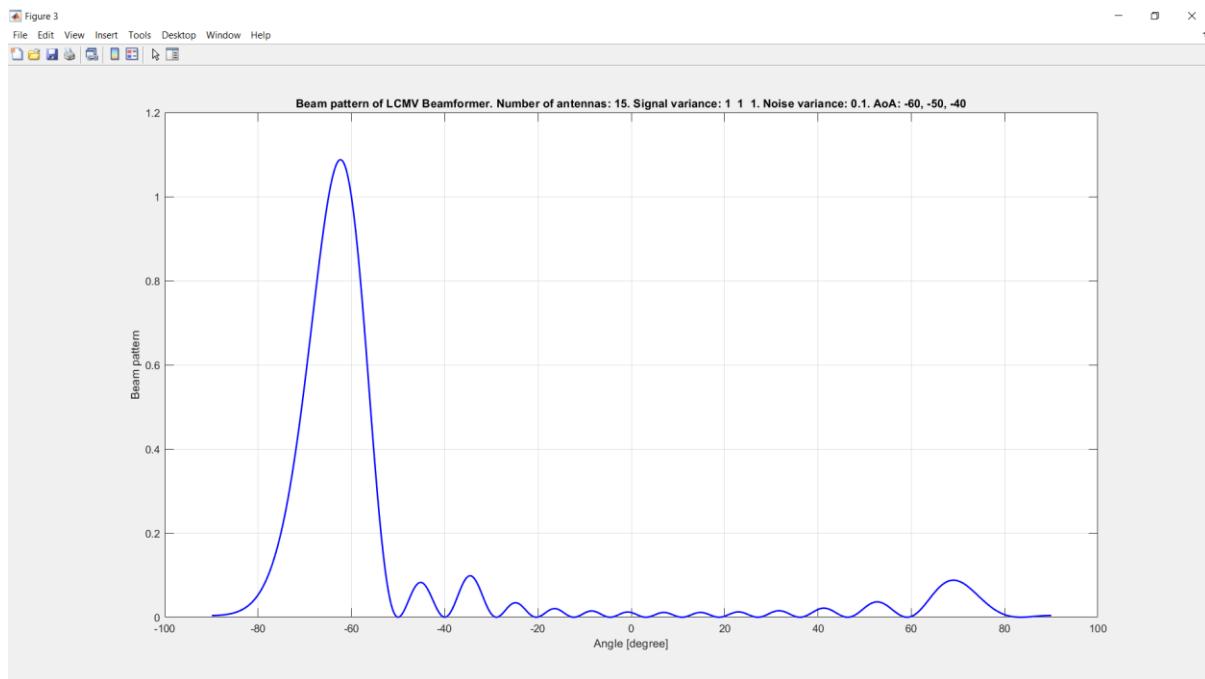


Hình 51. Đồ thị so sánh tín hiệu trộn tại *ngõ vào*, tín hiệu mong muốn và tín hiệu sau khi xử lý tại *ngõ ra* với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$.

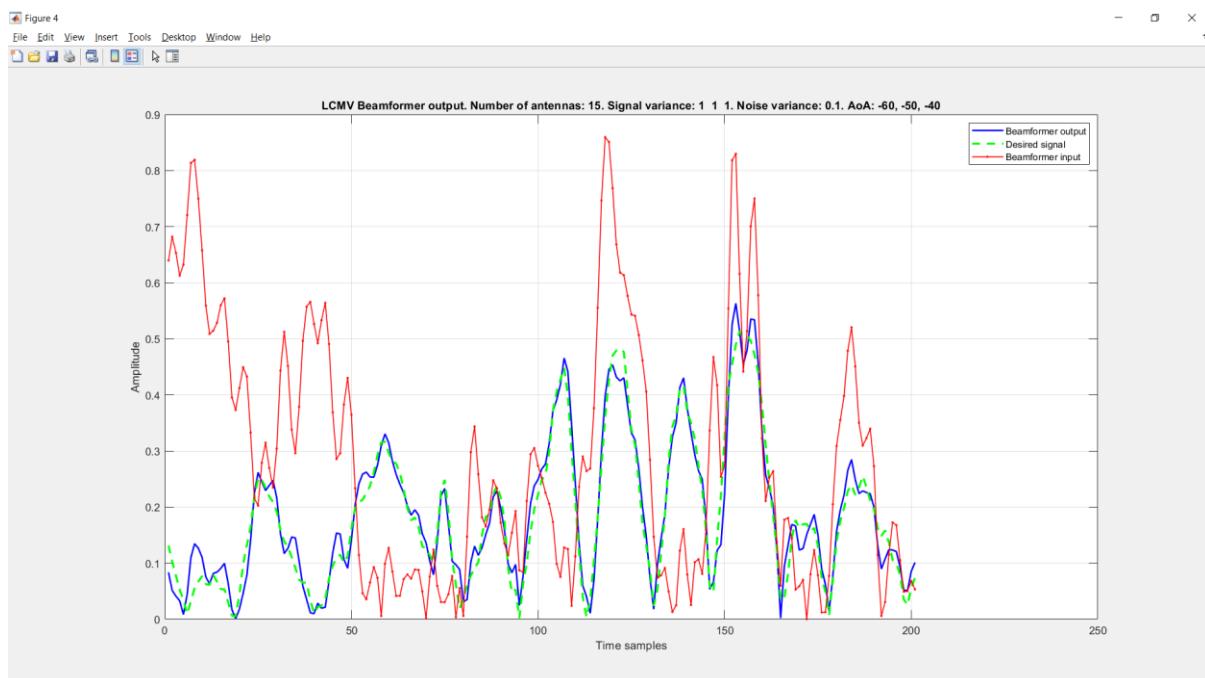
$RMSE = 0.04166$.

Nhân xét: Qua các trường hợp khảo sát với khoảng cách góc giữa tín hiệu mong muốn và can nhiễu nhỏ dần, búp sóng chính càng lệch khỏi góc của hướng tới mong muốn, và các điểm null (điểm không) cũng ở các vị trí góc khác góc của hướng can nhiễu, thể hiện rõ ràng ở trường hợp này.

c) $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$:



Hình 52. Đồ thị tần số của $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$.



Hình 53. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$.

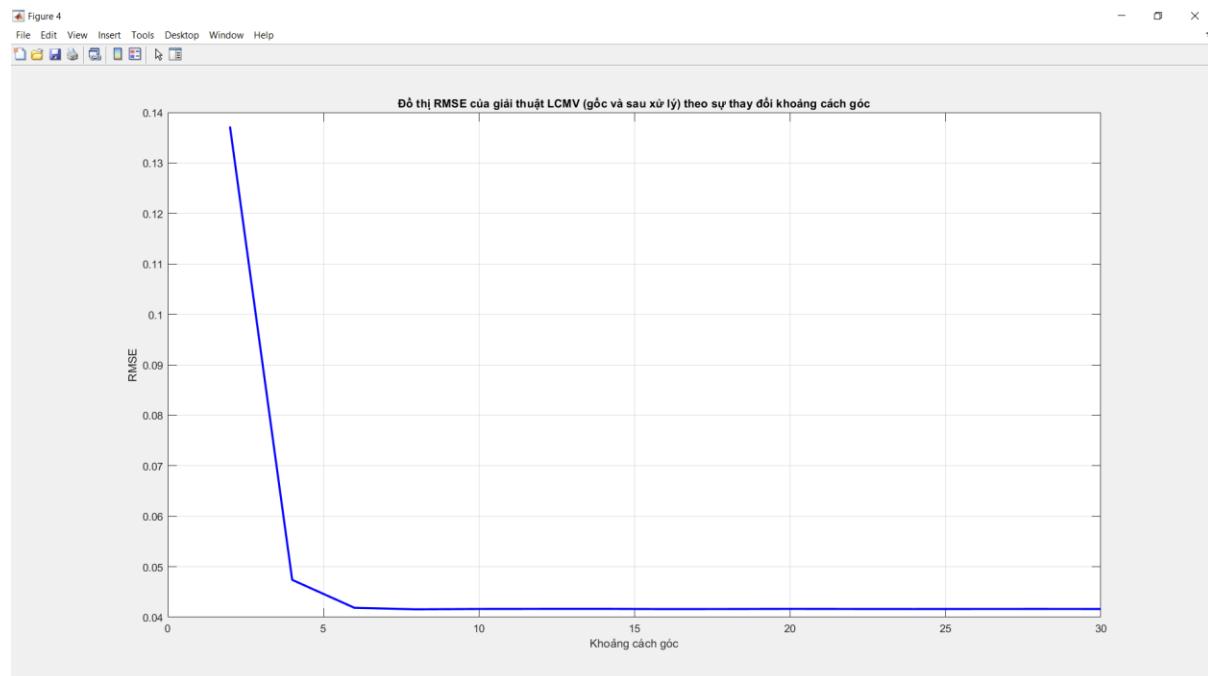
RMSE = 0.04155.

Nhân xét: Thể hiện rõ ràng việc giảm khoảng cách góc giữa tín hiệu mong muốn và can nhiễu ảnh hưởng đến khả năng xử lý tín hiệu của giải thuật. Búp sóng chính bây giờ nằm lệch so với góc của hướng tới mong muốn một khoảng tương đối lớn.

d) Nhận xét chung:

Dựa vào sự thay đổi ở góc của búp sóng chính ở đồ thị tuyến tính và búp sóng tại các góc của hướng tới can nhiễu, có thể thấy khoảng cách góc giữa các tín hiệu càng gần thì việc thu tín hiệu mong muốn và triệt can nhiễu càng khó hơn (búp sóng chính không nằm ở góc của hướng tới tín hiệu mong muốn, vẫn có búp sóng tại góc hướng tới can nhiễu).

❖ Sự thay đổi RMSE giữa tín hiệu gốc và tín hiệu sau khi qua xử lý theo sự thay đổi của khoảng cách góc giữa hướng tới tín hiệu mong muốn và can nhiễu:



Hình 54. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi khoảng cách góc.

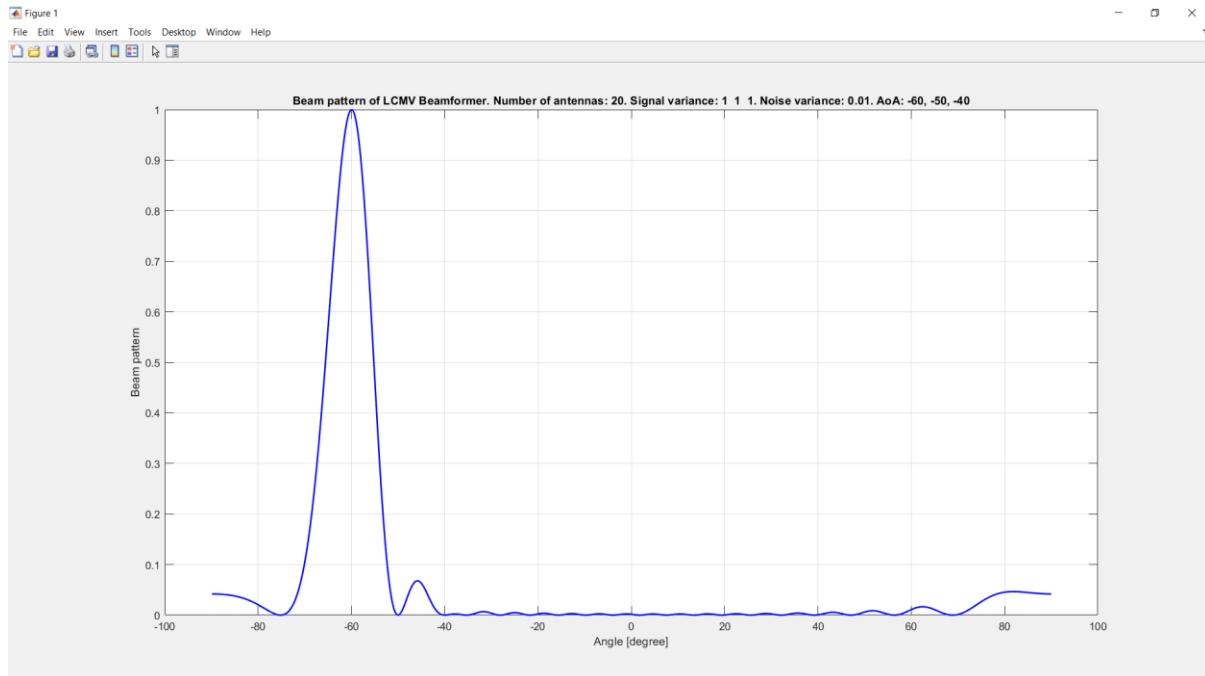
Nhân xét: Khi bắt đầu giảm khoảng cách góc giữa tín hiệu mong muốn và can nhiễu, RMSE cũng tăng dần lên, thể hiện sự sai lệch giữa tín hiệu tại ngõ ra và tín hiệu mong muốn. Đặc biệt, với khoảng cách góc đủ lớn thì sự sai lệch không thay đổi nhiều, tuy nhiên, khi can nhiễu và tín hiệu mong muốn càng gần nhau (dưới 6°), RMSE tăng lên rất nhanh. Điều này chứng tỏ khi hướng đến của các tín hiệu càng gần nhau thì việc xử lý tín hiệu càng phức tạp hơn, khó tối ưu tín hiệu mong muốn thu được.

III.1.3. Thay đổi phương sai nhiễu (noise variance):

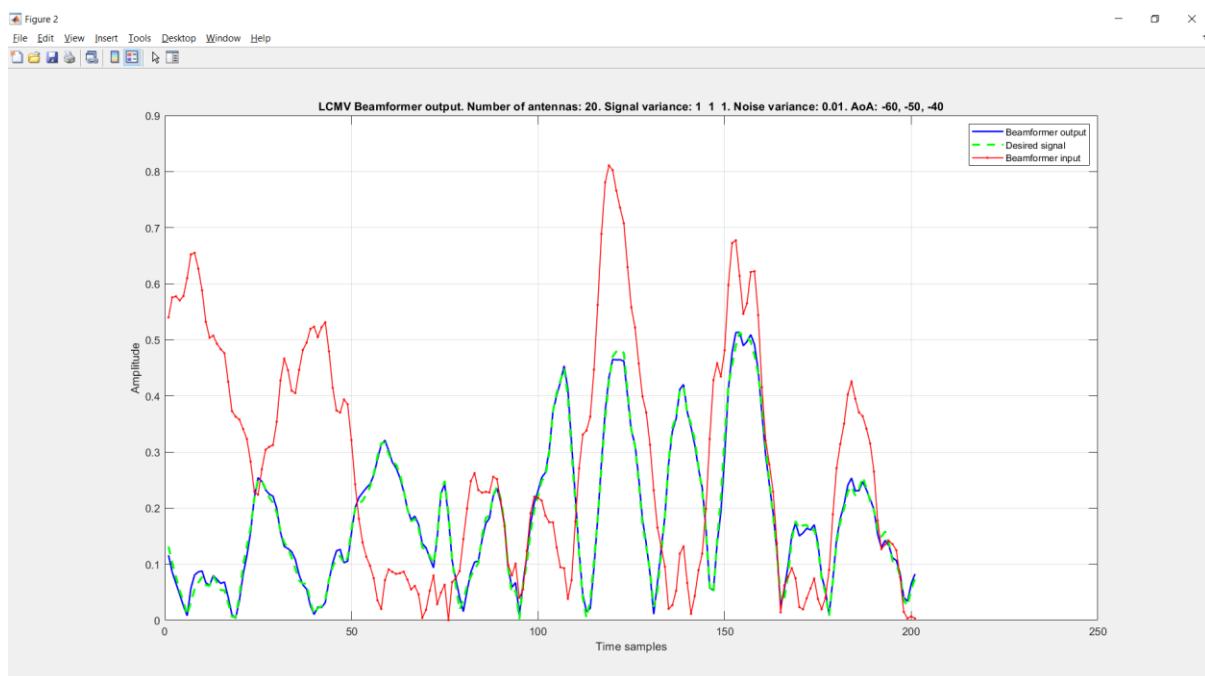
Để thuận tiện cho việc quan sát sự hiệu quả của giải thuật, ta cần chú ý vào tham số Root Mean-square Error (*RMSE*) và quan sát đồ thị tuyến tính cùng đồ thị so sánh tín hiệu mong muốn và ngõ ra của các trường hợp.

Thay đổi phương sai nhiễu $nv = 0.01, 0.1, 1$ rồi quan sát kết quả:

- ❖ Đồ thị tuyến tính, đồ thị so sánh giữa các tín hiệu và *RMSE* tương ứng:
- a) $nv = 0.01$:



Hình 55. Đồ thị tuyến tính của giải thuật với $nv = 0.01$.

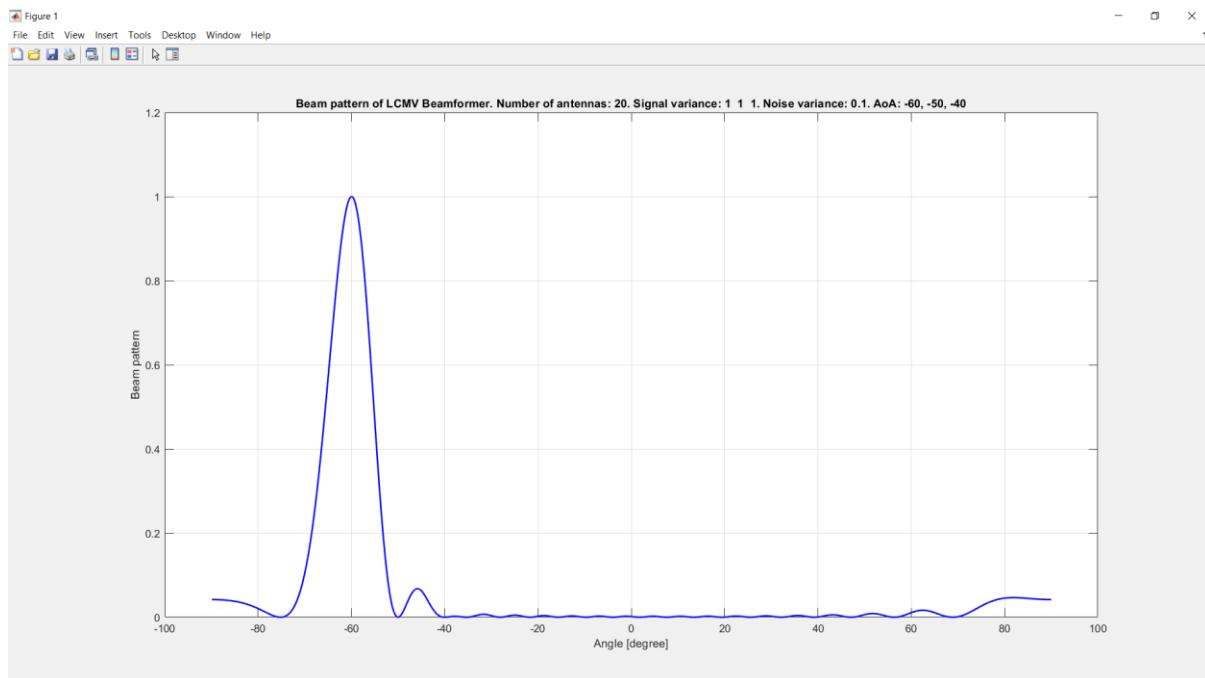


Hình 56. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.01$.

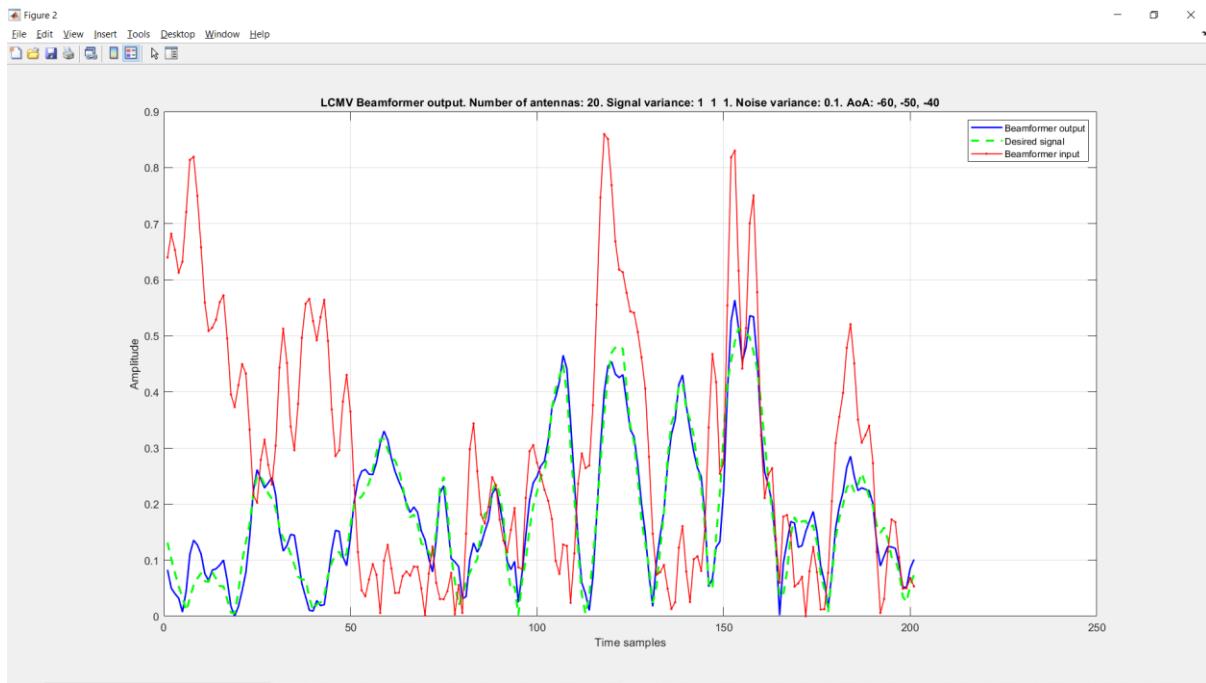
$RMSE = 0.013161$.

Nhận xét: Với phuơng sai nhiễu nhỏ, khả năng xử lý của bộ lọc khá tốt, bằng chứng là búp sóng chính ở góc của hướng tới tín hiệu mong muốn và các điểm null (điểm không) (điểm không) ở góc của hướng tới can nhiễu. RMSE cũng rất nhỏ cho thấy sự tương đồng lớn giữa tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục).

b) $nv = 0.1$:



Hình 57. Đồ thị tuyền tính của giải thuật với $nv = 0.1$.

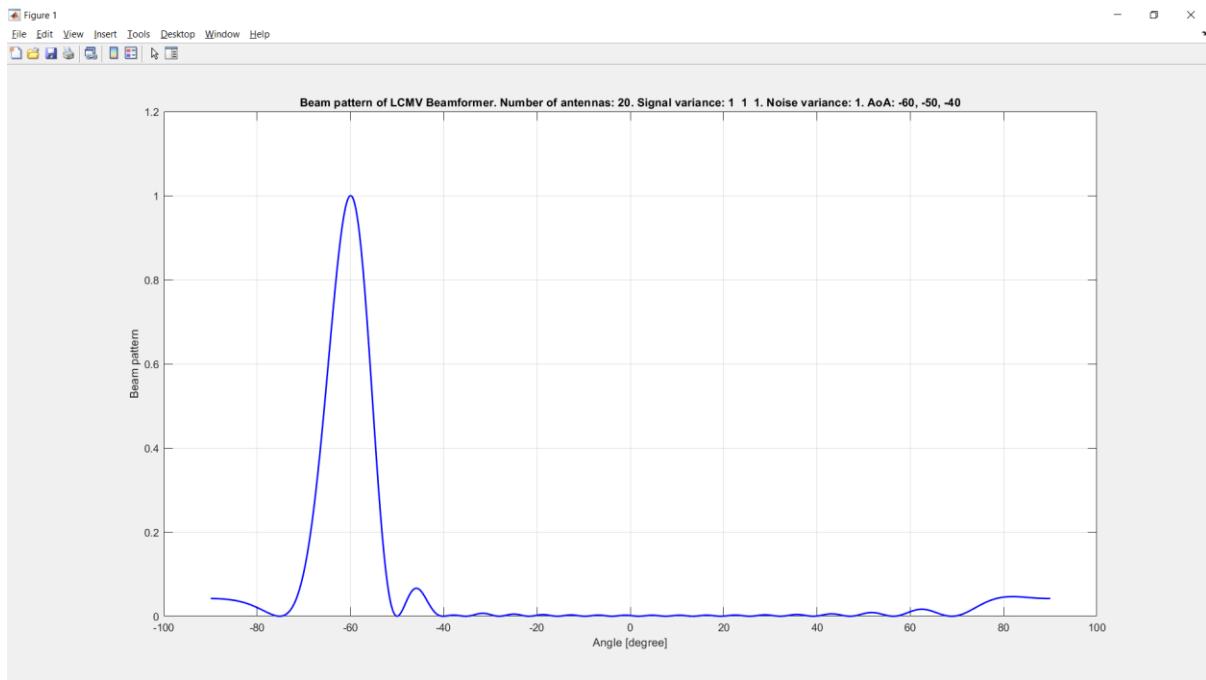


Hình 58. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.1$.

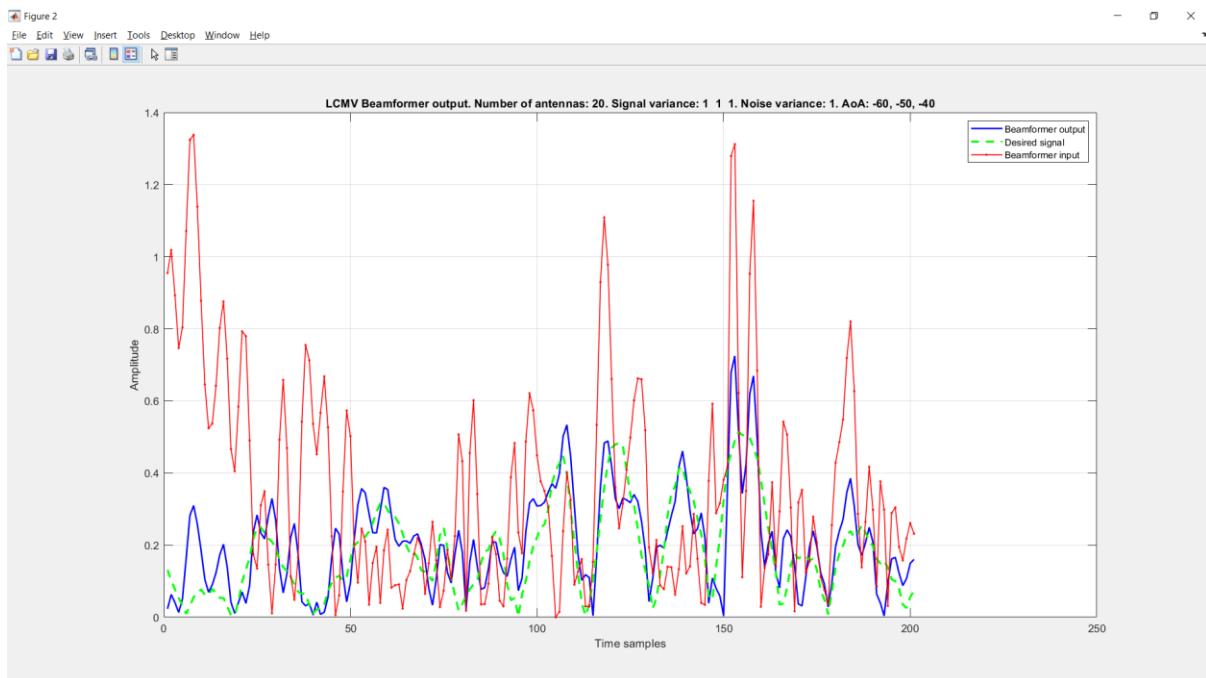
$RMSE = 0.041633$.

Nhận xét: Khi bắt đầu tăng phương sai nhiễu, sự sai lệch giữa tín hiệu tại ngõ ra và tín hiệu mong muốn bắt đầu xuất hiện. Mặc dù ở đồ thị tuyến tính, góc của búp sóng chính vẫn trùng với góc của hướng tới mong muốn và các điểm null (điểm không) vẫn ở góc của hướng tới can nhiễu, nhưng có thể thấy sự sai lệch nhiều hơn giữa đồ thị tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục). Để nhận thấy nhất chính là sự tăng lên rõ rệt của RMSE.

c) $nv = 1$:



Hình 59. Đồ thị tuyen tính của giải thuật với $nv = 1$.



Hình 60. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 1$.

$RMSE = 0.13208$.

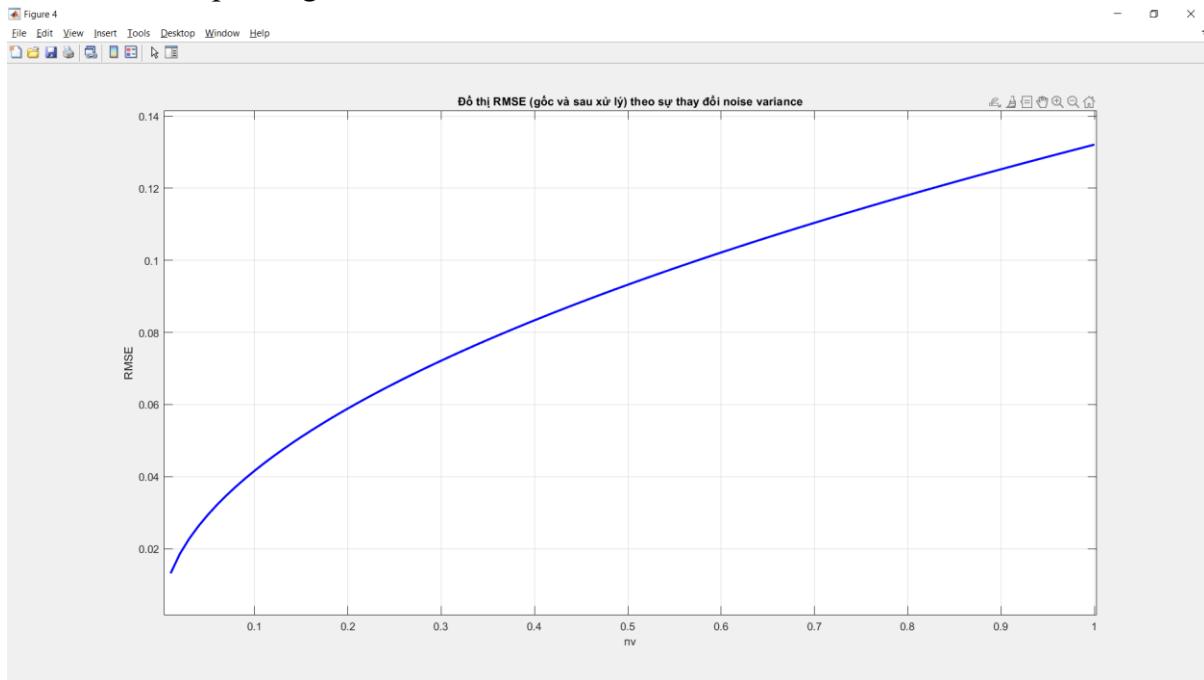
Nhân xét: Khi tăng phuông sai nhiều lớn như vậy, sự sai lệch giữa tín hiệu tại ngõ ra và tín hiệu mong muốn được nhận thấy rõ hơn nữa. Mặc dù ở đồ thị tuyen tính, góc của búp sóng chính vẫn trùng với góc của hướng tới mong muốn, nhưng góc của các điểm null (điểm không) đã lệch so với góc của hướng tới can nhiều. Đặc biệt, sự sai lệch giữa

đồ thị tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục) tăng lên rất nhiều và RMSE cũng tăng lên rất đáng kể.

d) Nhận xét chung:

Sự thay đổi của phương sai nhiễu ít có ảnh hưởng tới đồ thị tuyến tính, tín hiệu tại ngõ ra vẫn rất tương đồng với tín hiệu mong muốn và các can nhiễu hầu như được triệt tiêu. Tuy nhiên, RMSE lại có sự thay đổi rõ rệt. Phương sai nhiễu càng tăng thì RMSE càng tăng.

- ❖ Sự thay đổi RMSE giữa tín hiệu gốc và tín hiệu sau khi qua xử lý theo sự thay đổi của phương sai nhiễu:



Hình 61. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi phương sai nhiễu.

Nhận xét: Khi phương sai nhiễu tăng, RMSE cũng tăng lên rõ rệt, tín hiệu tại ngõ ra càng sai lệch so với tín hiệu mong muốn. Tuy nhiên, sự thay đổi của phương sai nhiễu ít có ảnh hưởng tới đồ thị tuyến tính, tín hiệu tại ngõ ra vẫn rất tương đồng với tín hiệu mong muốn và các can nhiễu hầu như được triệt tiêu. Mặc dù vậy, RMSE lại có sự thay đổi rõ rệt. Phương sai nhiễu càng tăng thì RMSE càng tăng.

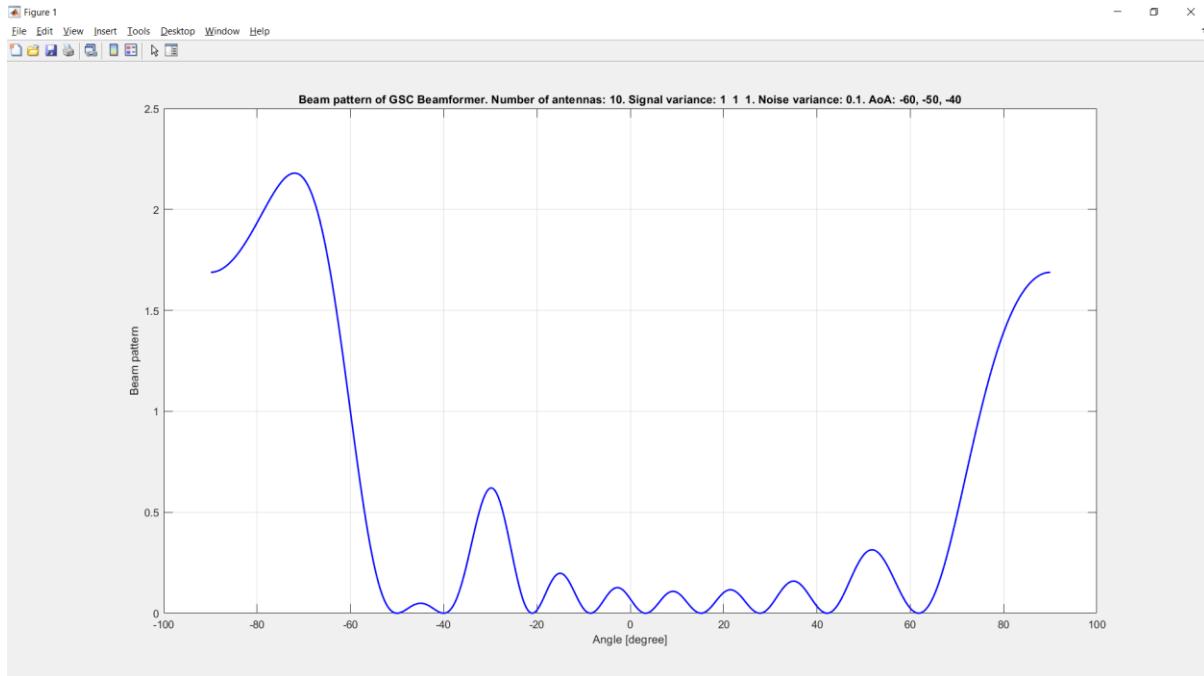
III.2. Bộ triệt các búp bên tổng quát (GSC):

III.2.1. Thay đổi số phần tử an-ten M:

Để thuận tiện cho việc quan sát sự hiệu quả của giải thuật, ta cần chú ý vào tham số Root Mean-square Error (RMSE) và quan sát đồ thị tuyến tính cùng đồ thị so sánh tín hiệu mong muốn và ngõ ra của các trường hợp.

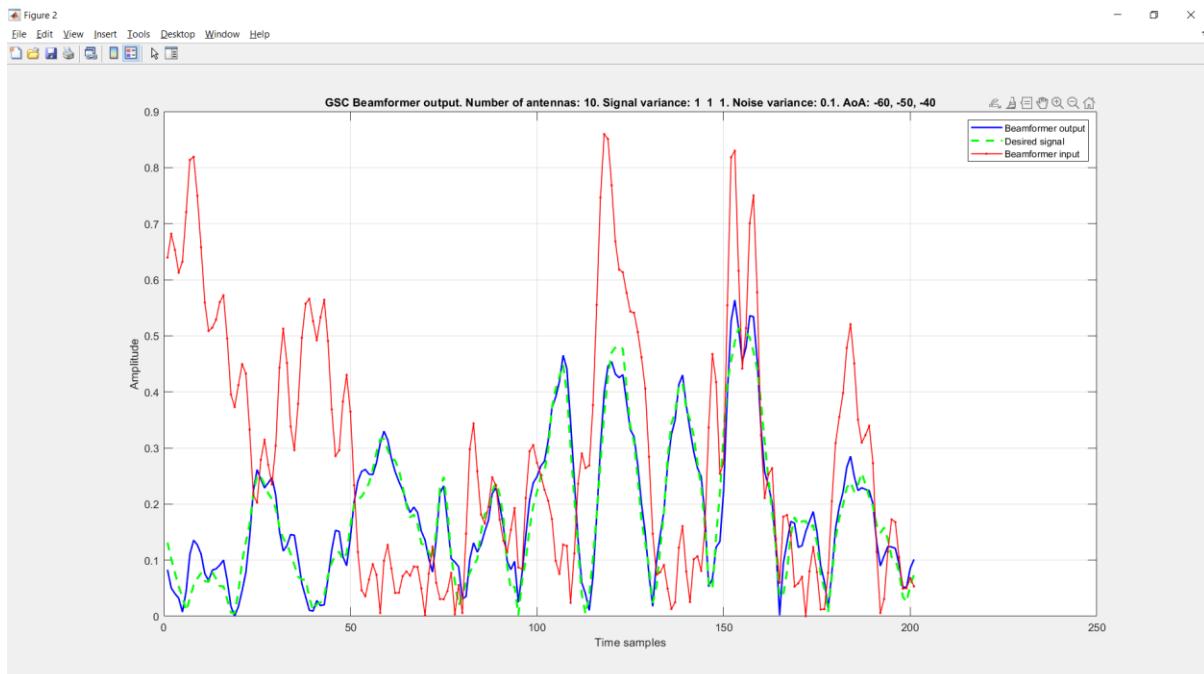
Thay đổi số phần tử $M = 10, 15, 17, 18, 20$ rồi quan sát kết quả:

- ❖ Đồ thị tuyển tính, đồ thị so sánh giữa các tín hiệu và RMSE tương ứng:
- a) $M = 10$:



Hình 62. Đồ thị tuyển tính của giải thuật với $M = 10$.

Nhân xét: Với số lượng phần tử антенна chưa đủ lớn, giải thuật chưa thể tối ưu tín hiệu mong muốn cần thu được tại ngõ ra, với việc góc của búp sóng chính không phải ở góc của tín hiệu mong muốn.

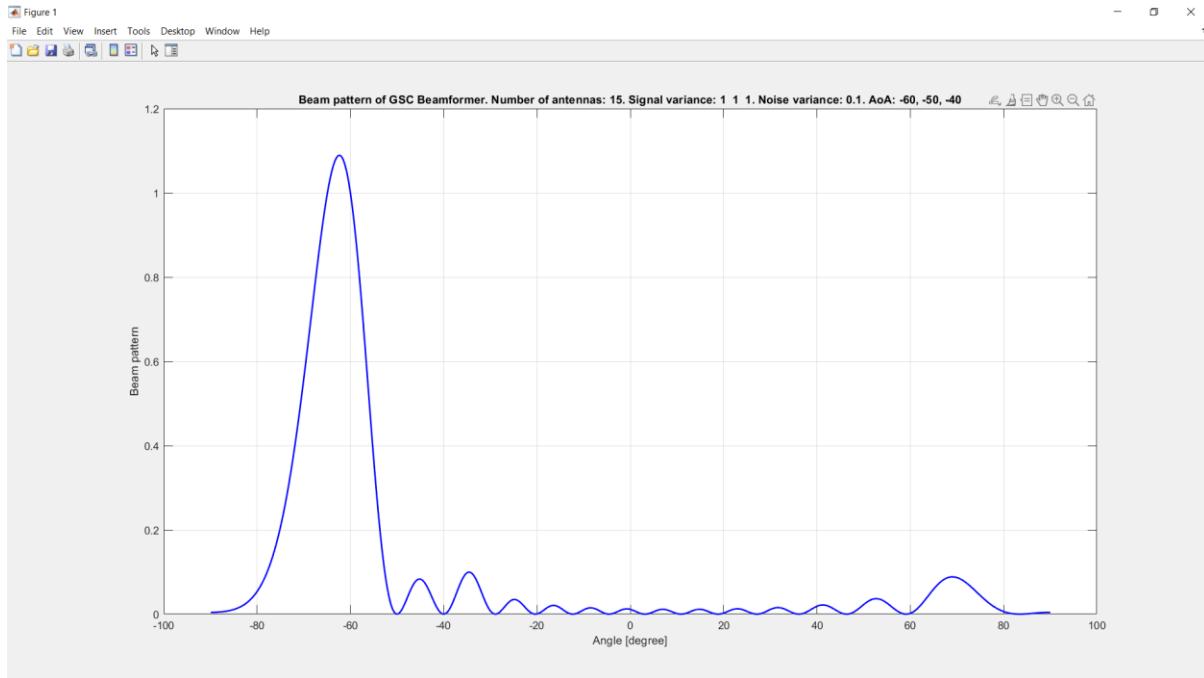


Hình 63. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 10$.

$RMSE = 0.041618$.

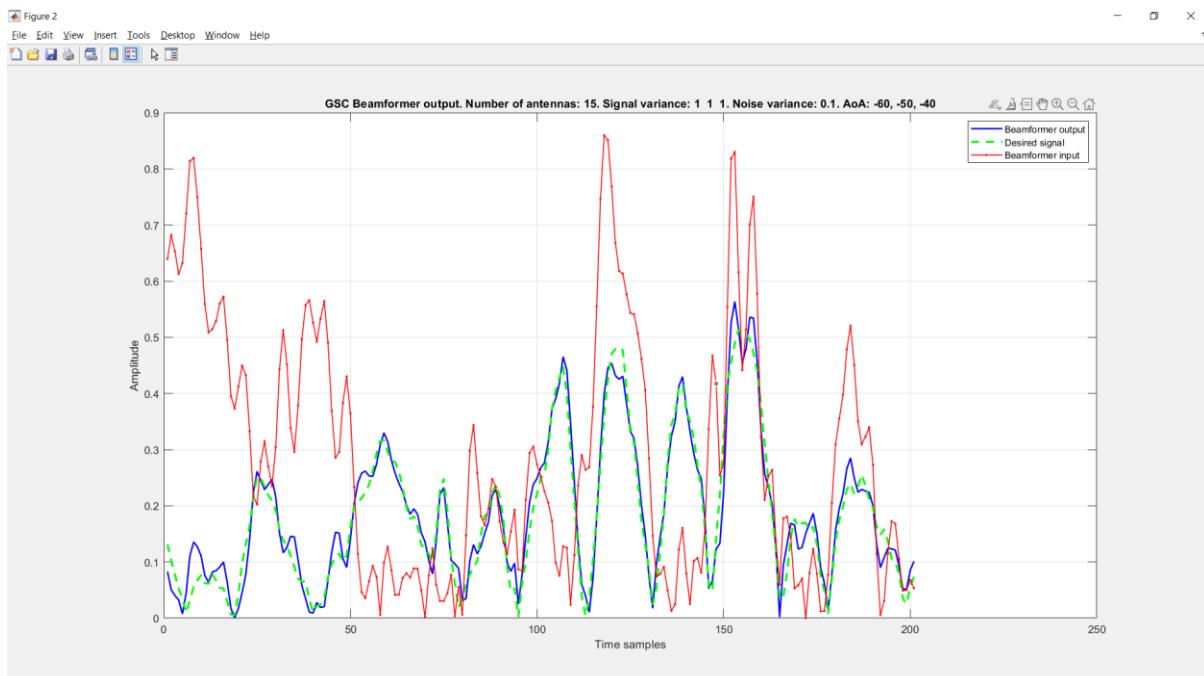
Nhân xét: Đồ thị vẫn cho thấy kết quả là tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra khá tương đồng nhau. Đồng thời, RMSE nhỏ cho thấy tín hiệu tại ngõ ra sau khi qua xử lý cũng tương đồng với tín hiệu mong muốn.

b) $M = 15$:



Hình 64. Đồ thị tuyền tính của giải thuật với $M = 15$.

Nhân xét: Với số phần tử an-ten $M = 15$ là đủ lớn để giải thuật cho kết quả tối ưu tốt hơn, khi mà góc của búp sóng chính gần với góc hướng tới mong muốn đạt cực đại.

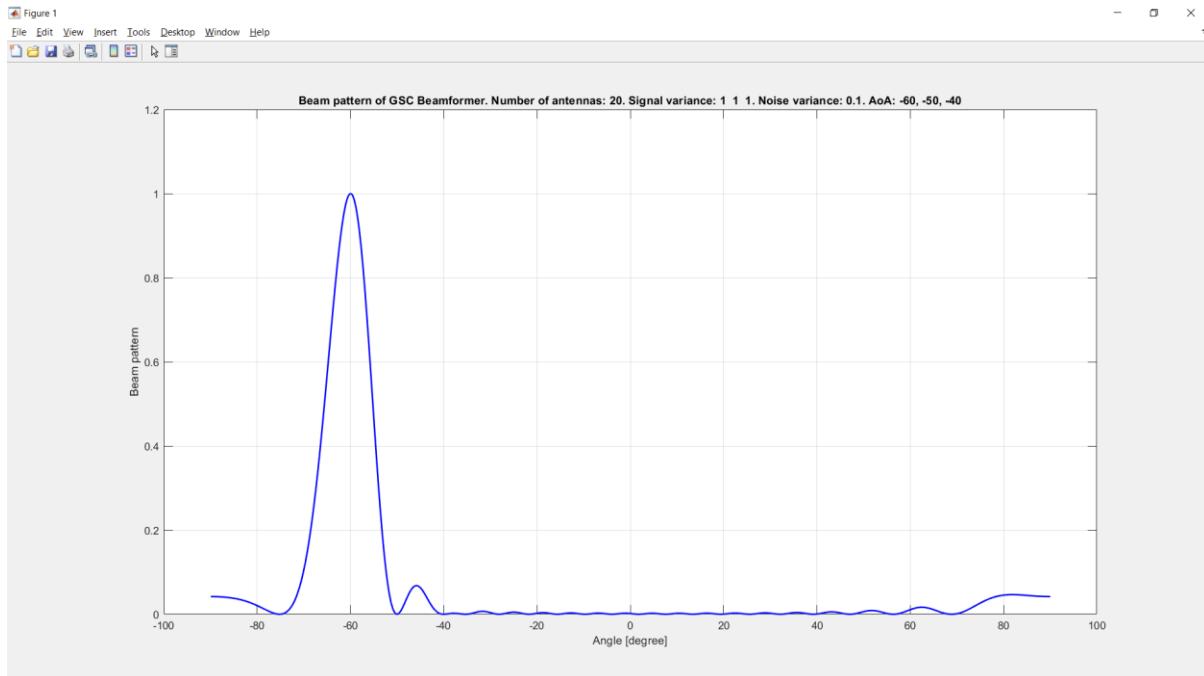


Hình 65. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 15$.

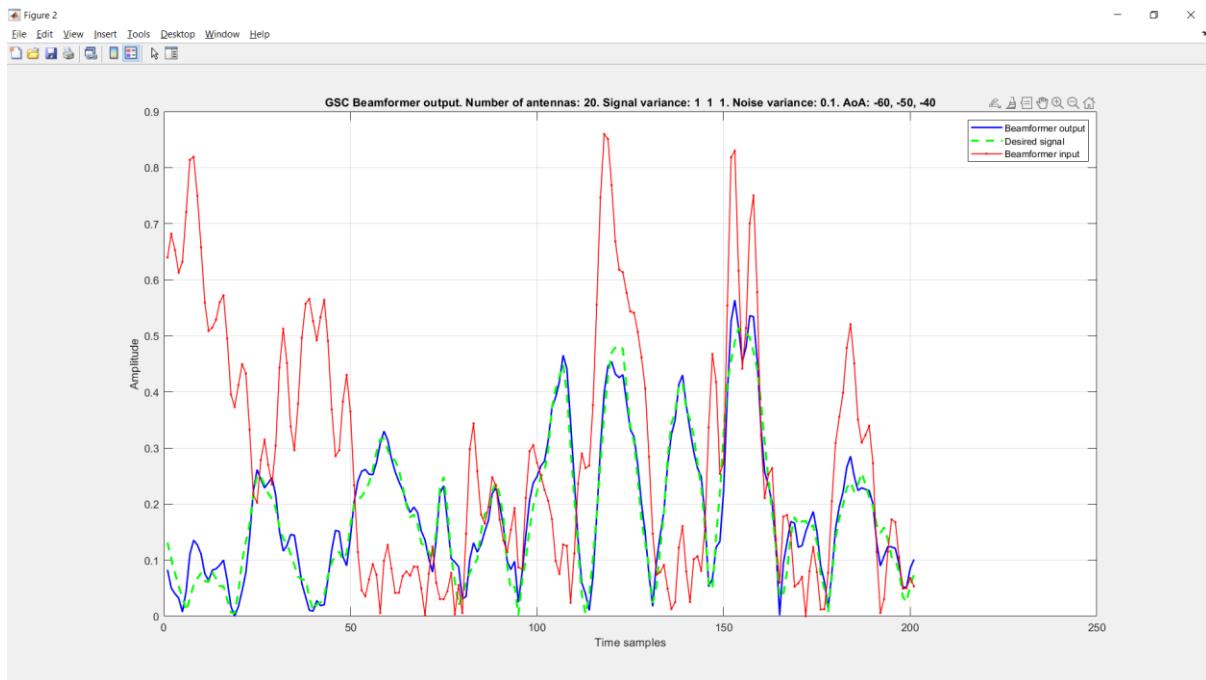
$RMSE = 0.041618$.

Nhận xét: Đồ thị và RMSE cũng cho thấy kết quả là tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra khá tương đồng nhau. Tuy nhiên, khác với LCMV, RMSE không đổi khi tăng số phần tử an-ten.

c) $M = 20$:



Hình 66. Đồ thị tuyen tính của giải thuật với $M = 20$.



Hình 67. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $M = 20$.

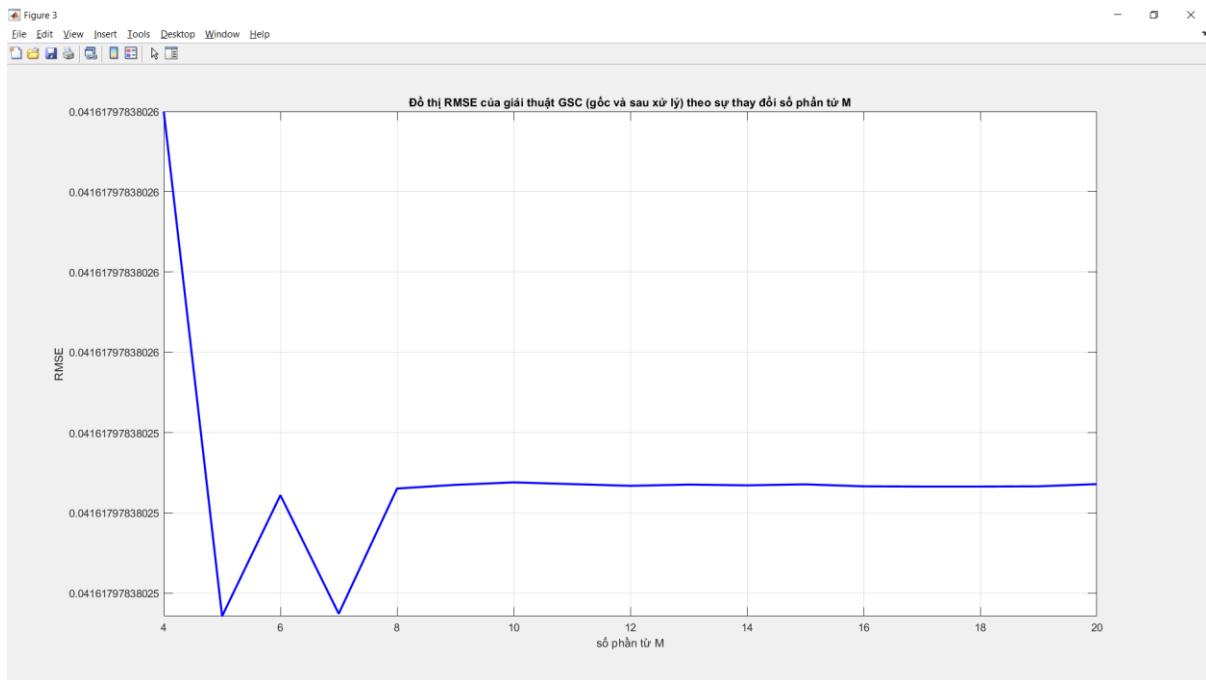
$RMSE = 0.041618$.

Nhận xét: Sau khi khảo sát thêm $M = 17, 18$, kết quả tốt nhất là với $M = 20$, độ tương đồng giữa tín hiệu tại ngõ ra với tín hiệu mong muốn cao hơn so với các kết quả trước. Búp sóng chính ở chính xác góc của tín hiệu mong muốn, đồng thời tại các góc của hướng tới can nhiễu là các điểm null (điểm không), chứng tỏ khả năng triệt can nhiễu và tối ưu tín hiệu mong muốn của giải thuật với $M = 20$ là rất tốt.

d) Nhận xét chung:

Dựa vào sự thay đổi ở góc của búp sóng chính ở đồ thị tuyến tính, có thể thấy với số lượng phần tử M càng tăng thì góc của búp sóng chính càng gần với góc của hướng tới tín hiệu mong muốn hơn, đồng thời khả năng triệt can nhiễu của giải thuật cũng rất tốt (ở các góc của hướng tới can nhiễu thì luôn là các điểm null (điểm không)) và ở số phần tử $M = 20$ thì giải thuật đạt hiệu quả tốt nhất (đúng với hướng tới mong muốn là ở góc -60°). Ngoài ra, $RMSE$ luôn có giá trị 0.041618, chứng tỏ giải thuật có tính ổn định cao và hiệu quả.

- ❖ Sự thay đổi $RMSE$ giữa tín hiệu gốc và tín hiệu sau khi qua xử lý theo sự thay đổi của số phần tử an-ten M :



Hình 68. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi số phần tử an-ten M.

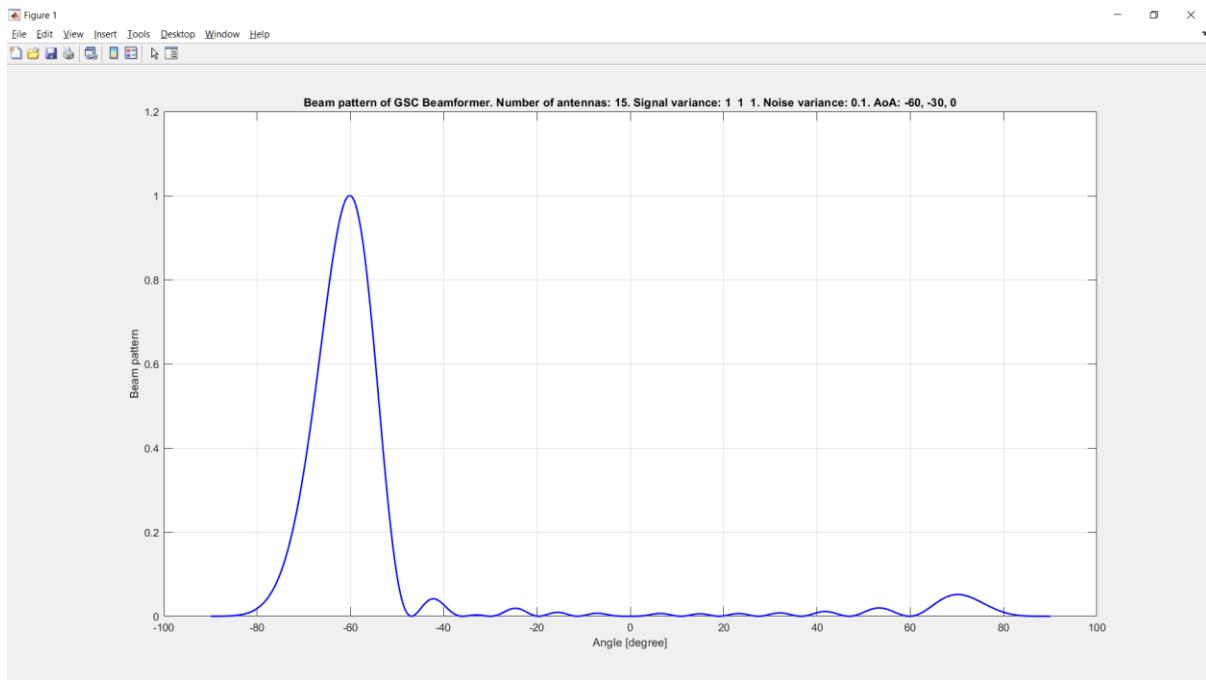
Nhân xét: Khi tăng số phần tử an-ten M thì RMSE của giải thuật cũng không thay đổi nhiều, điều này chứng tỏ giải thuật có tính ổn định và xử lý tín hiệu tốt hơn so với giải thuật LCMV. Với số lượng phần tử an-ten M càng tăng, giải thuật càng tối ưu cho việc loại bỏ nhiễu và can nhiễu, tín hiệu tại ngõ ra càng giống với tín hiệu mong muốn thu được. Đặc biệt, với số phần tử M đủ nhiều thì RMSE hầu như không đổi, chứng tỏ giải thuật có tính ổn định càng cao (tối ưu với số phần tử M đó).

III.2.2. Giảm khoảng cách góc giữa các tín hiệu mong muốn và can nhiễu:

Để thuận tiện cho việc quan sát sự hiệu quả của giải thuật, ta cần chú ý vào tham số Root Mean-square Error (RMSE) và quan sát đồ thị tuyến tính cùng đồ thị so sánh tín hiệu mong muốn và ngõ ra của các trường hợp.

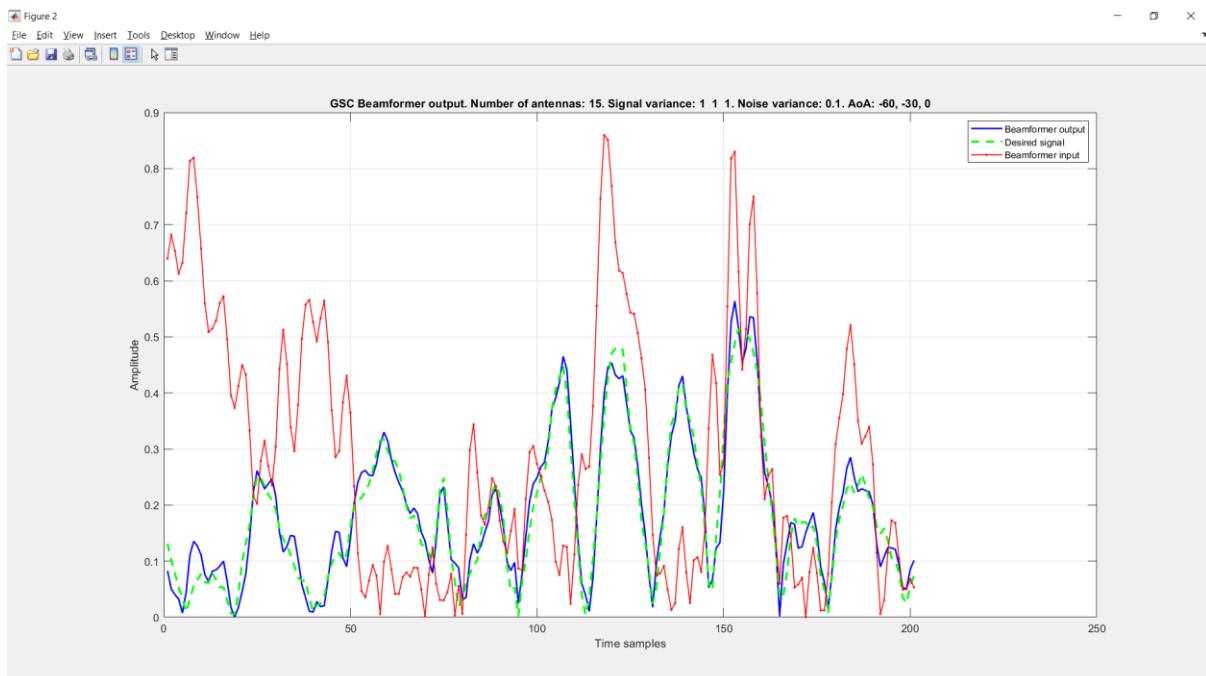
Thay đổi góc của các hướng tới $[\varphi_0, \varphi_1, \varphi_2]$ lần lượt là $[-60^\circ, -30^\circ, 0^\circ]$; $[-60^\circ, -35^\circ, -10^\circ]$; $[-60^\circ, -40^\circ, -20^\circ]$; $[-60^\circ, -45^\circ, -30^\circ]$; $[-60^\circ, -50^\circ, -40^\circ]$ rồi quan sát kết quả (sử dụng số phần tử $M = 15$ để dễ dàng quan sát sự khác biệt):

- ❖ Đồ thị tuyến tính, đồ thị so sánh giữa các tín hiệu và RMSE tương ứng:
- $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$:



Hình 69. Đồ thị tuyền tính của giải thuật với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$.

Nhân xét: Với khoảng cách góc giữa tín hiệu mong muốn và can nhiễu lớn như vậy, giải thuật có thể xử lý dễ dàng. Tại góc của hướng tới tín hiệu mong muốn là búp sóng chính, đồng thời tại góc của các hướng tới can nhiễu là các điểm null (điểm không), cho thấy giải thuật xử lý tín hiệu tốt với trường hợp này.

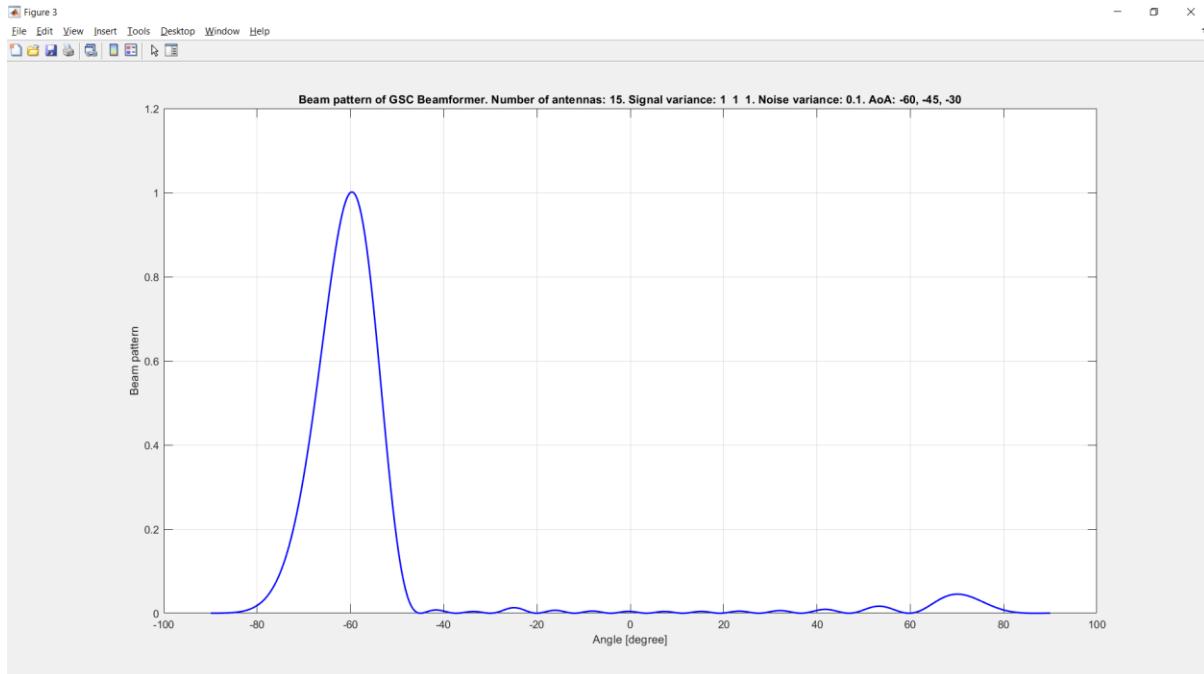


Hình 70. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -30^\circ, 0^\circ]$.

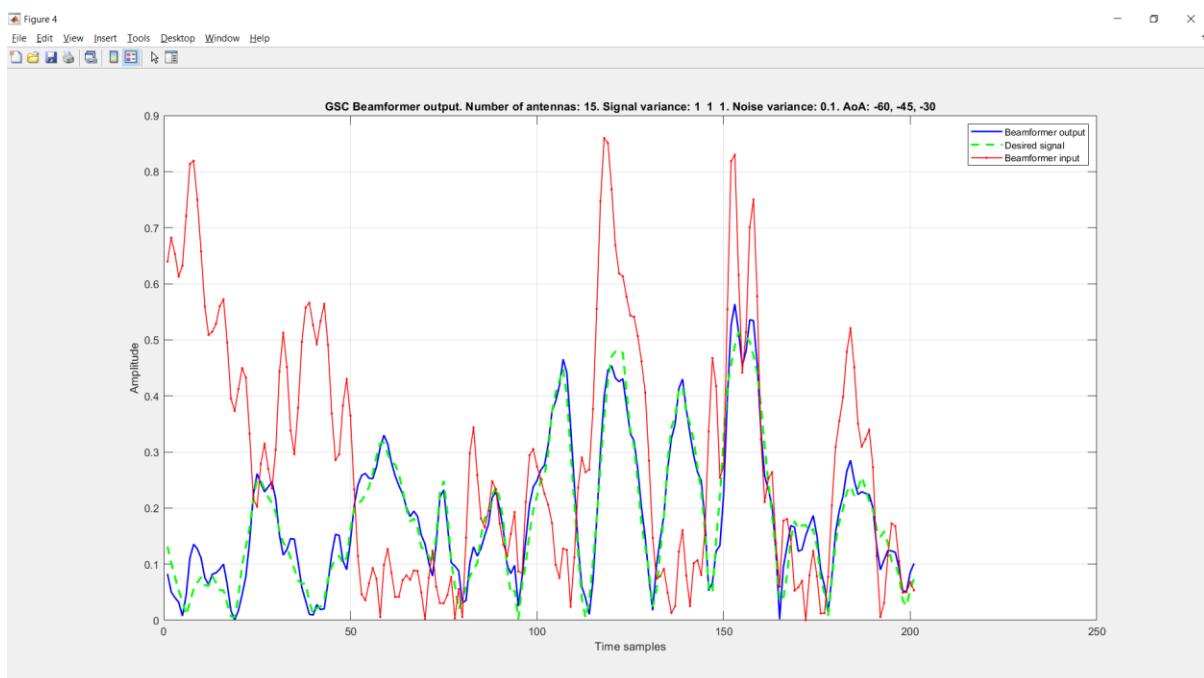
$$RMSE = 0.041618.$$

Nhân xét: Với khoảng cách góc giữa hướng tới của tín hiệu mong muốn và *can nhiễu* ở trường hợp này, tín hiệu tại *ngõ ra* (màu lục) rất tương đồng với tín hiệu mong muốn (màu xanh) và khác biệt rất nhiều với tín hiệu trộn tại *ngõ vào* (màu đỏ). RMSE cũng rất nhỏ cho thấy sự tương đồng giữa tín hiệu tại *ngõ ra* sau khi qua xử lý và tín hiệu mong muốn.

b) $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$:



Hình 71. Đồ thị tần số của giải thuật với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$.

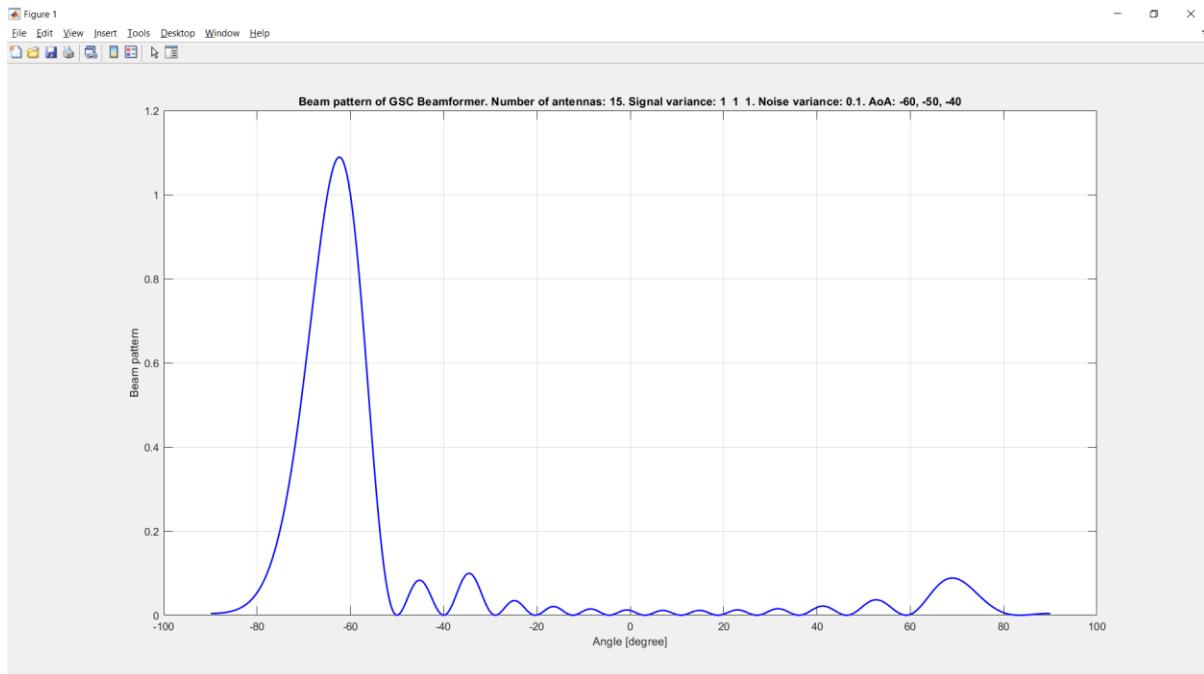


Hình 72. Đồ thị so sánh tín hiệu trộn tại *ngõ vào*, tín hiệu mong muốn và tín hiệu sau khi xử lý tại *ngõ ra* với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -45^\circ, -30^\circ]$.

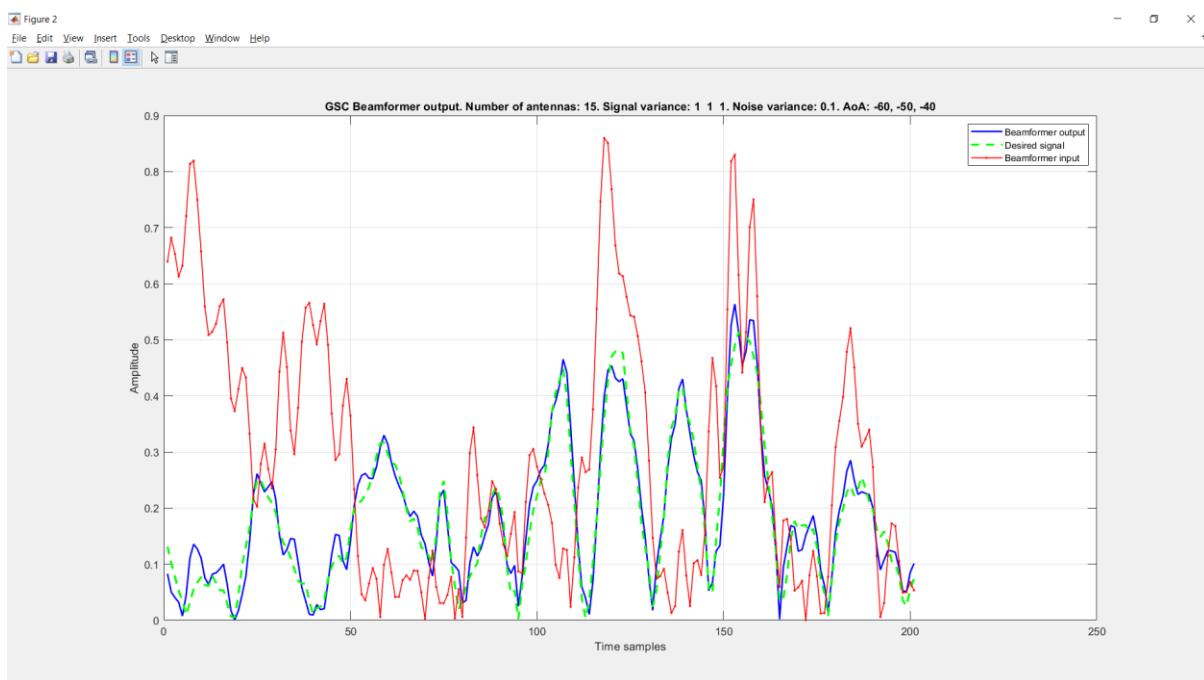
$RMSE = 0.041618$.

Nhân xét: Tương tự như trên, khi càng giảm khoảng cách góc giữa tín hiệu mong muốn và can nhiễu, sự sai lệch cũng đã xuất hiện. Mặc dù vậy, độ lệch giữa góc của búp sóng chính so với góc của hướng tới tín hiệu mong muốn cũng rất nhỏ (tương tự như trường hợp trước) và sự sai lệch giữa đồ thị tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục) tăng lên không đáng kể, RMSE vẫn không đổi.

c) $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$:



Hình 73. Đồ thị tuyến tính của giải thuật với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$.



Hình 74. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $[\varphi_0, \varphi_1, \varphi_2] = [-60^\circ, -50^\circ, -40^\circ]$.

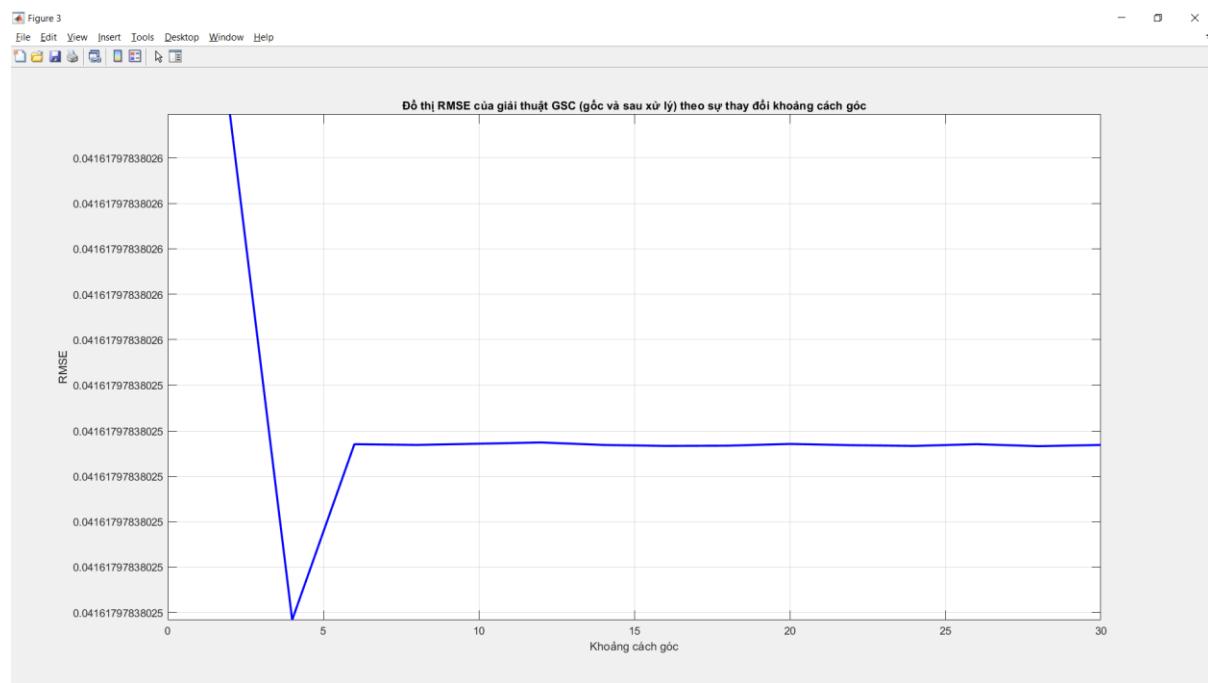
RMSE = 0.041618.

Nhận xét: Qua các trường hợp khảo sát cho thấy khi khoảng cách góc giữa tín hiệu mong muốn và can nhiễu giảm dần ảnh hưởng đến khả năng xử lý tín hiệu của giải thuật. Búp sóng chính bây giờ nằm lệch so với góc của hướng tới mong muốn một khoảng tương đối lớn. Mặc dù vậy, sự sai lệch giữa đồ thị tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục) tăng lên không đáng kể, RMSE cũng vẫn như các trường hợp trước.

d) Nhận xét chung:

Dựa vào sự thay đổi vị trí góc của búp sóng chính ở đồ thị tuyến tính, có thể thấy khoảng cách góc giữa các tín hiệu càng gần thì việc thu tín hiệu mong muốn và triệt các can nhiễu càng khó hơn (sai lệch góc thu tín hiệu mong muốn và sai lệch các góc cần triệt can nhiễu). Tuy nhiên, khác biệt so với giải thuật LCMV chính là đồ thị tuyến tính tại các góc của hướng tới can nhiễu luôn bằng 0 (điểm null (điểm không)) và RMSE không thay đổi.

- ❖ Sự thay đổi RMSE giữa tín hiệu gốc và tín hiệu sau khi qua xử lý theo sự thay đổi của khoảng cách góc giữa hướng tới của tín hiệu mong muốn và can nhiễu:



Hình 75. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý theo sự thay đổi khoảng cách góc giữa các tín hiệu.

Nhận xét: Khi bắt đầu giảm khoảng cách góc giữa tín hiệu mong muốn và can nhiễu, RMSE cũng tăng dần lên, thể hiện sự sai lệch giữa tín hiệu tại ngõ ra và tín hiệu mong muốn. Đặc biệt, với khoảng cách góc đủ lớn (trên 6°) thì RMSE không đổi, và mặc dù

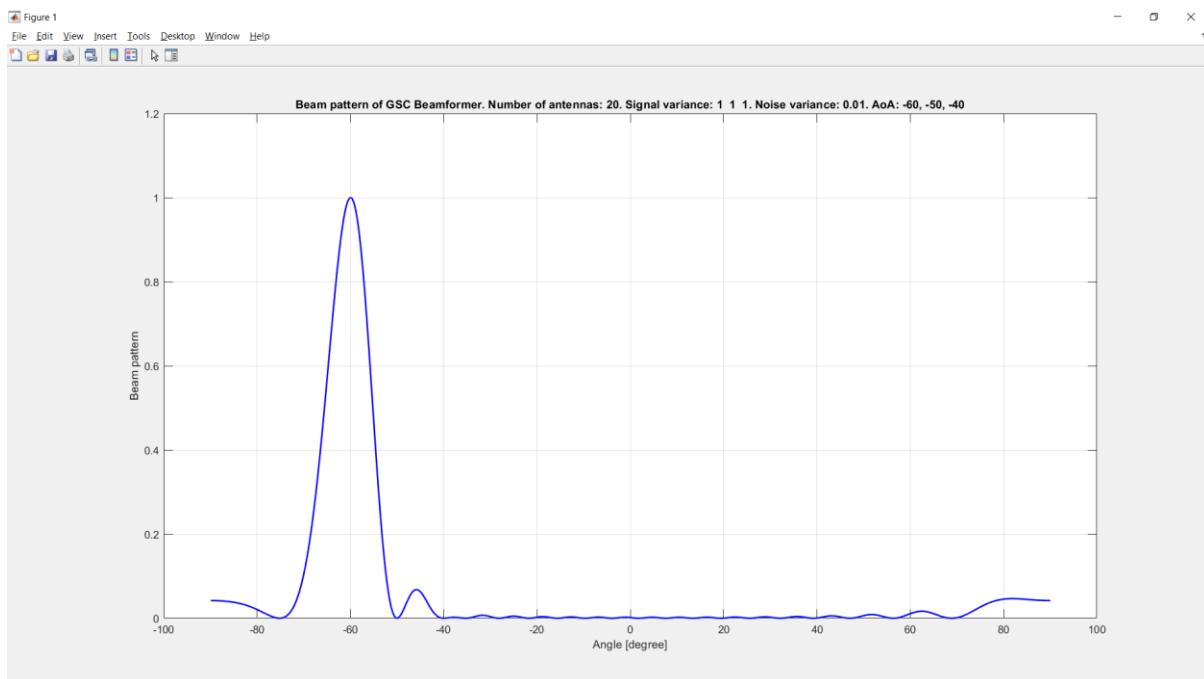
khi can nhiễu và tín hiệu mong muốn càng gần nhau ($dưới 4^\circ$), RMSE tăng không đáng kể. Đây là điểm cải thiện so với giải thuật LCMV. Khi khoảng cách góc giữa các hướng của tín hiệu mong muốn và tín hiệu can nhiễu càng gần nhau, thì việc xử lý tín hiệu càng trở nên phức tạp, khó để tối ưu tín hiệu mong muốn hơn. Tuy nhiên với GSC, RMSE hầu như không đổi, có giá trị ổn định, đây là điểm tối ưu hơn so với LCMV, cho thấy giải thuật này xử lý tốt hơn và ổn định hơn.

III.2.3. Thay đổi phương sai nhiễu (noise variance):

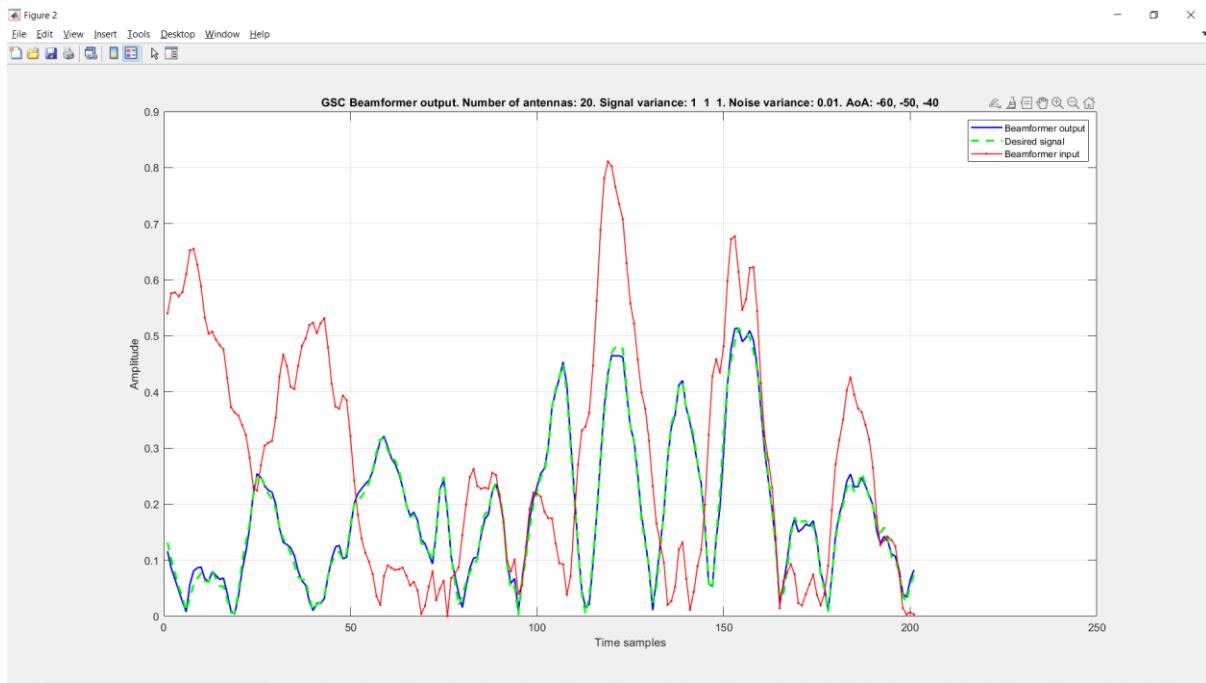
Để thuận tiện cho việc quan sát sự hiệu quả của giải thuật, ta cần chú ý vào tham số Root Mean-square Error (RMSE) và quan sát đồ thị tuyển tính cùng đồ thị so sánh tín hiệu mong muốn và ngõ ra của các trường hợp.

Thay đổi phương sai nhiễu $nv = 0.01, 0.1, 1$ rồi quan sát kết quả:

- ❖ Đồ thị tuyển tính, đồ thị so sánh giữa các tín hiệu và RMSE tương ứng:
- a) $nv = 0.01$:



Hình 76. Đồ thị tuyển tính của giải thuật với $nv = 0.01$.

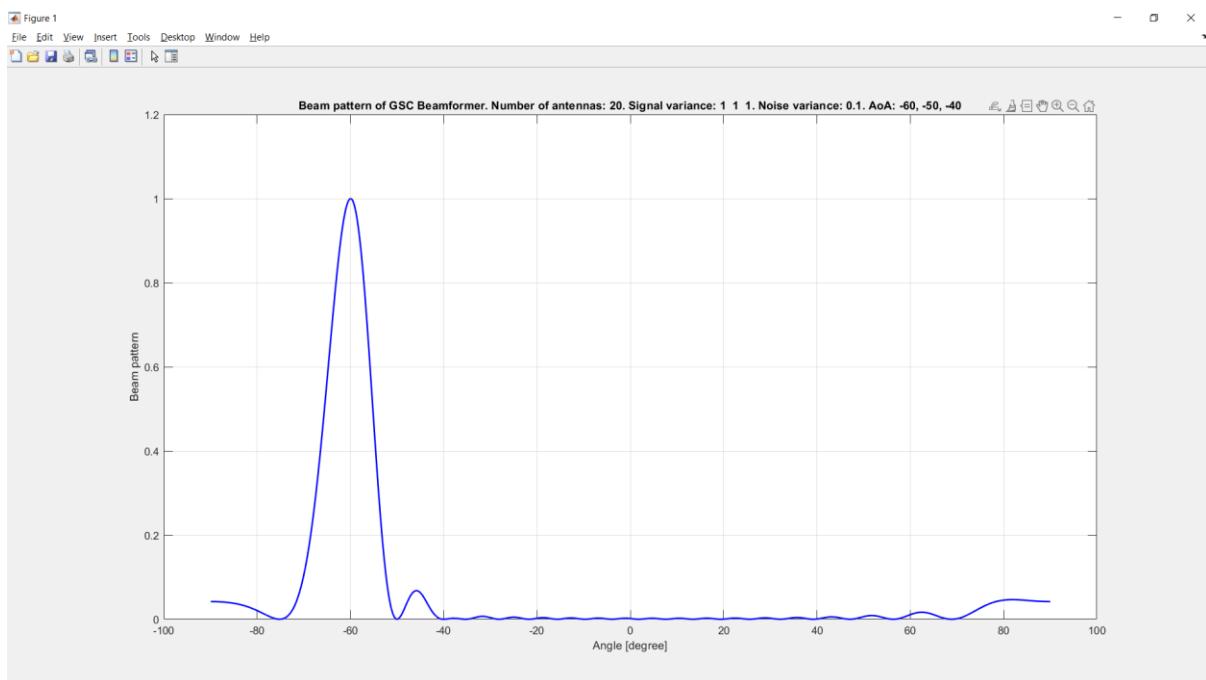


Hình 77. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.01$.

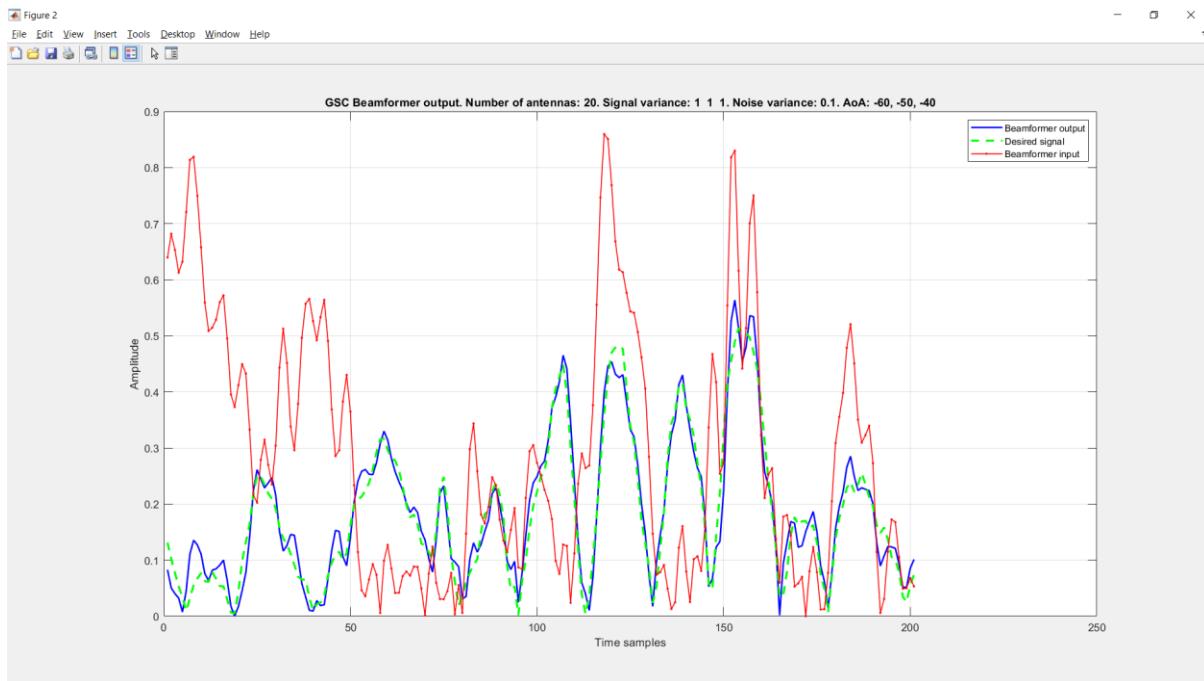
$RMSE = 0.013161$.

Nhận xét: Với phương sai nhiễu nhỏ, khả năng xử lý của bộ lọc tốt, bằng chứng là búp sóng chính ở góc của hướng tới tín hiệu mong muốn và các điểm null (điểm không) ở góc của hướng tới can nhiễu. RMSE cũng rất nhỏ cho thấy sự tương đồng lớn giữa tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục).

b) $nv = 0.1$:



Hình 78. Đồ thị tuyến tính của giải thuật với $nv = 0.1$.

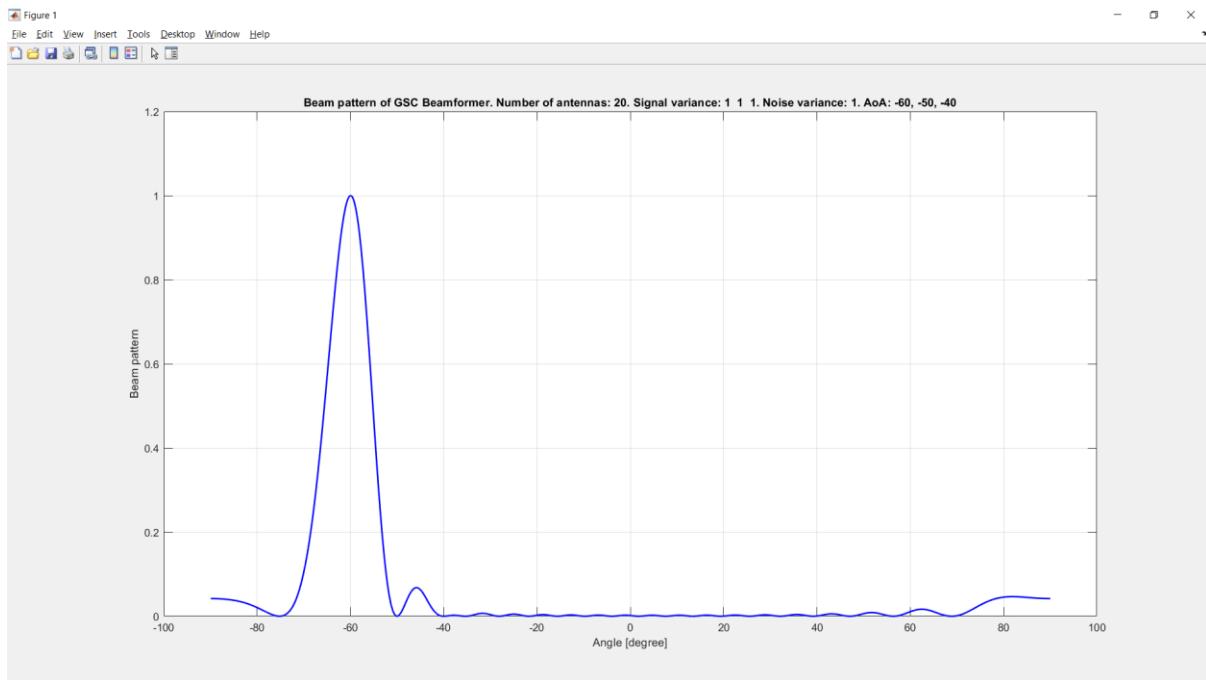


Hình 79. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 0.1$.

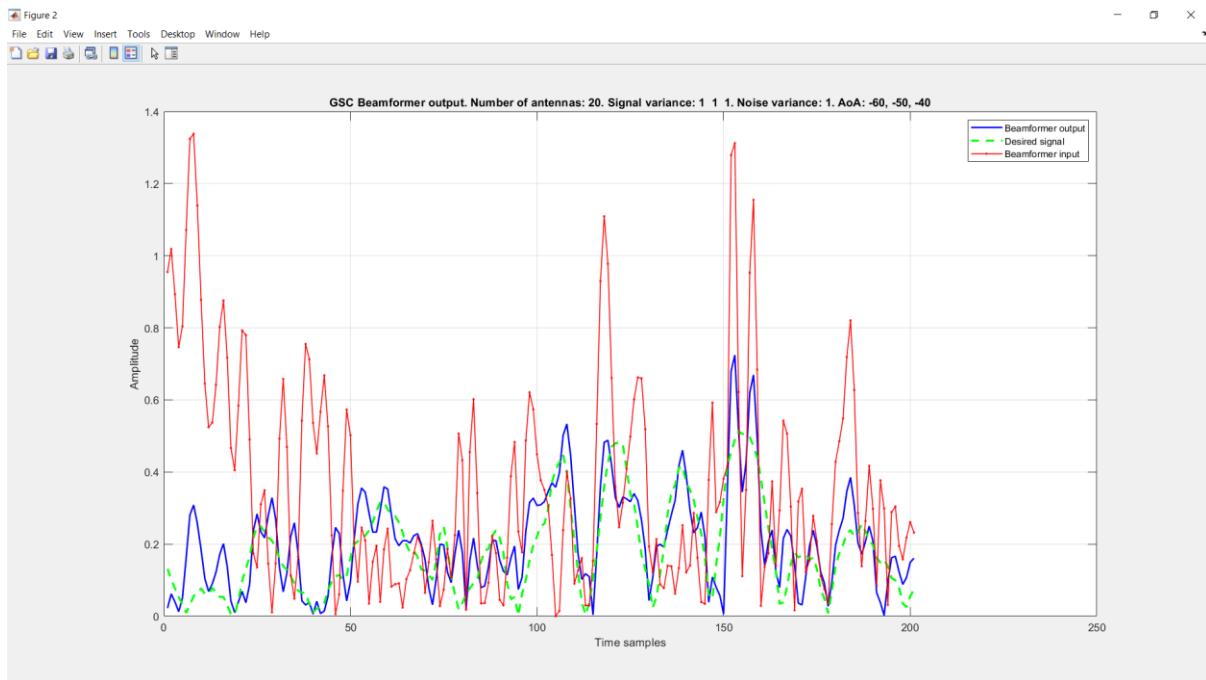
$RMSE = 0.041618$.

Nhận xét: Khi bắt đầu tăng phương sai nhiễu, sự sai lệch giữa tín hiệu tại ngõ ra và tín hiệu mong muốn bắt đầu xuất hiện. Mặc dù ở đồ thị tuyến tính, góc của búp sóng chính vẫn trùng với góc của hướng tới mong muốn và các điểm null (điểm không) vẫn ở góc của hướng tới can nhiễu, nhưng có thể thấy sự sai lệch nhiều hơn giữa đồ thị tín hiệu tại ngõ ra (màu xanh) và tín hiệu mong muốn (màu lục). Để nhận thấy nhất chính là sự tăng lên rõ rệt của RMSE.

c) $nv = 1$:



Hình 80. Đồ thị tuyen tính của giải thuật với $nv = 1$.



Hình 81. Đồ thị so sánh tín hiệu trộn tại ngõ vào, tín hiệu mong muốn và tín hiệu sau khi xử lý tại ngõ ra với $nv = 1$.

$RMSE = 0.13161$.

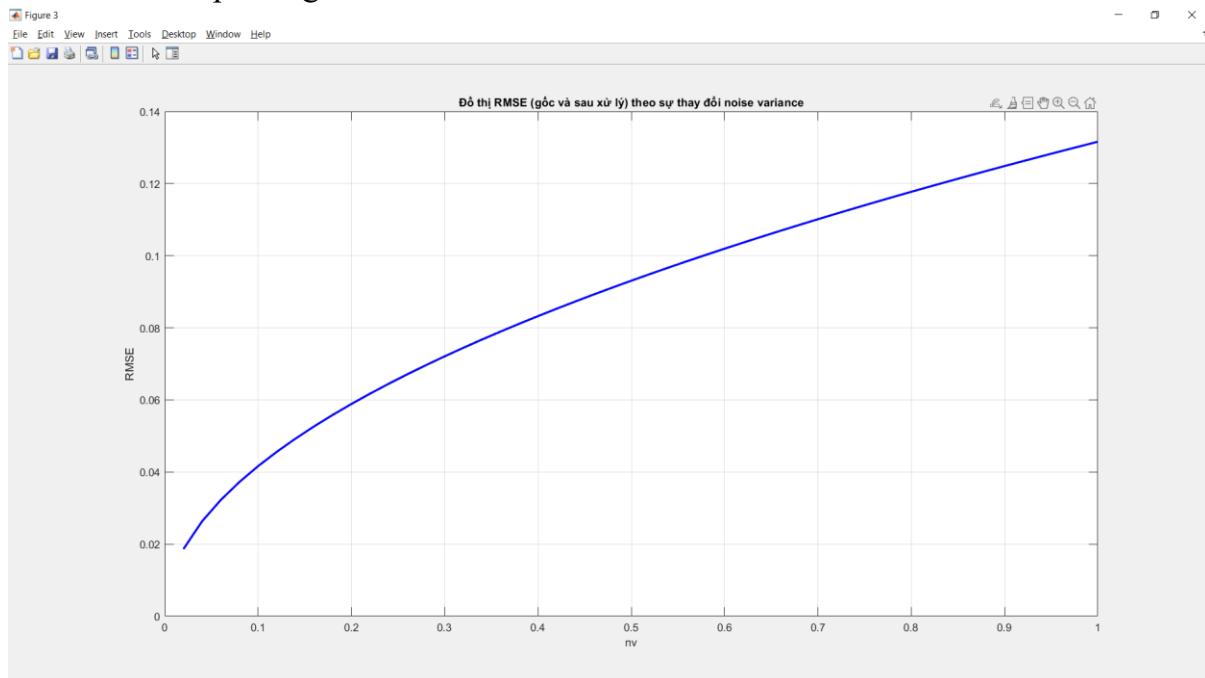
Nhận xét: Khi tăng phương sai nhiễu lớn như vậy, ở đồ thị tuyen tính, góc của búp sóng chính vẫn trùng với góc của hướng tới mong muốn, và góc của các điểm null (điểm không) cũng trùng với góc của hướng tới can nhiễu. Tuy nhiên sự sai lệch giữa đồ thị

tín hiệu tại ngo^o ra (màu xanh) và tín hiệu mong muốn (màu lục) tăng lên rất nhiều và RMSE cũng tăng lên rất đáng kể.

d) Nhận xét chung:

Sự thay đổi của phương sai nhiễu ít có ảnh hưởng tới đồ thị tuyến tính, tín hiệu mong muốn vẫn được tối ưu và các can nhiễu cũng được triệt tiêu. Tuy nhiên, RMSE lại có sự thay đổi rõ rệt. Phương sai nhiễu càng tăng thì RMSE càng tăng, đây vẫn là một điểm đáng lưu ý trong xử lý tín hiệu.

- ❖ Sự thay đổi RMSE giữa tín hiệu gốc và tín hiệu sau khi qua xử lý theo sự thay đổi của phương sai nhiễu:



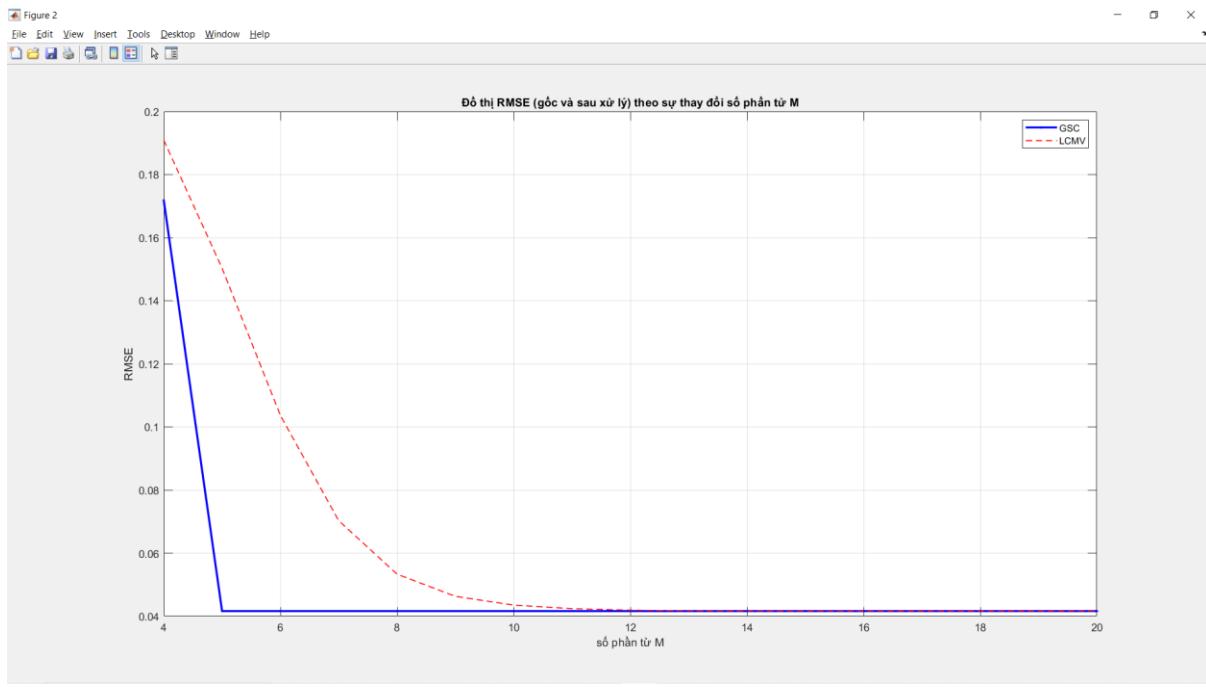
Hình 82. RMSE giữa tín hiệu gốc và tín hiệu trộn sau khi xử lý theo sự thay đổi phương sai nhiễu.

Nhận xét: Khi phương sai nhiễu tăng, RMSE cũng tăng lên rõ rệt, tín hiệu tại ngo^o ra càng sai lệch so với tín hiệu mong muốn. Mặc dù RMSE vẫn còn khá nhỏ ($RMSE \cong 0.13$ với $nv = 1$ khi so sánh giữa tín hiệu gốc và tín hiệu sau xử lý), chúng tôi giải thuật có hiệu quả cao trong việc triệt can nhiễu và tối ưu tín hiệu mong muốn. Mặc dù sự thay đổi của phương sai nhiễu cũng ít ảnh hưởng đến khả năng triệt can nhiễu, nhưng đây vẫn là một thông số cần phải lưu ý trong quá trình xử lý tín hiệu.

III.3. Tổng quát về hai giải thuật và phần mở rộng, cải tiến:

III.3.1. Thực hiện so sánh giữa 2 giải thuật LCMV và GSC dựa trên bài toán đã nêu trên:

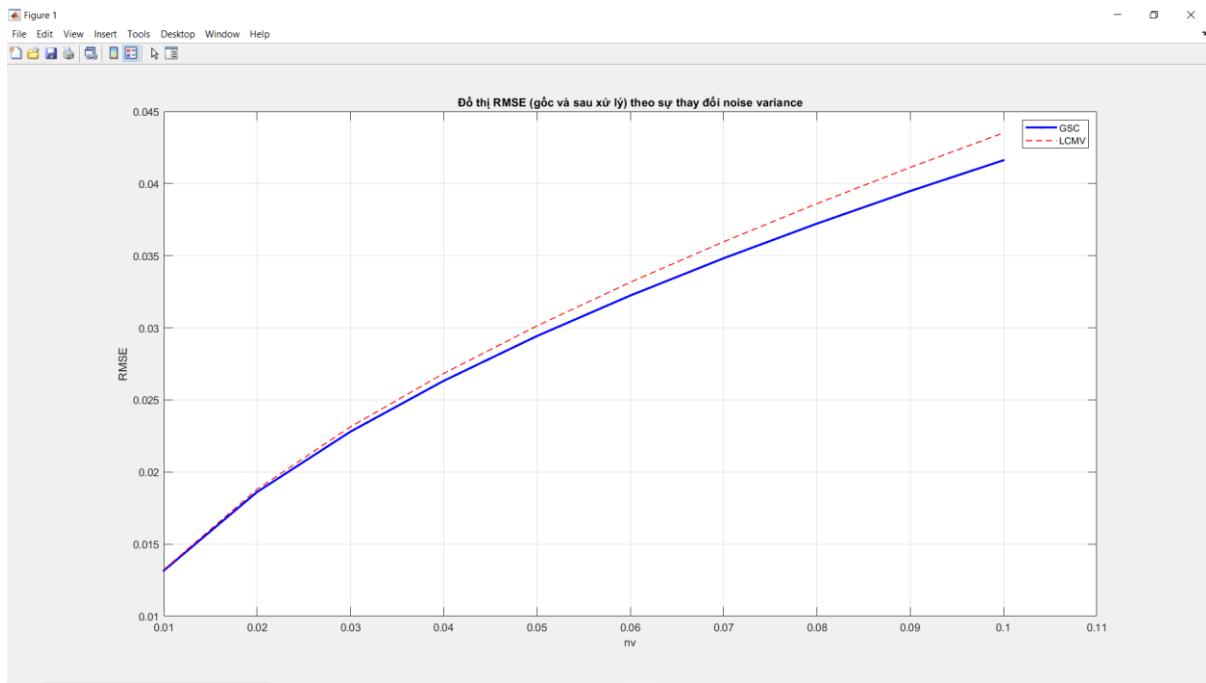
- ❖ RMSE (tín hiệu gốc và sau khi xử lý) của GSC và LCMV khi thay đổi số phần tử an-ten M :



Hình 83. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý của GSC và LCMV theo sự thay đổi số phần tử M.

Nhân xét: Khi số phần tử an-ten M tăng, RMSE của hai giải thuật đều giảm. Đặc biệt, RMSE hầu như không đổi khi số phần tử M đạt đến số lượng nhất định. Tuy nhiên, so sánh giữa hai giải thuật, GSC sẽ cho RMSE nhỏ hơn và ổn định hơn rất nhiều so với LCMV.

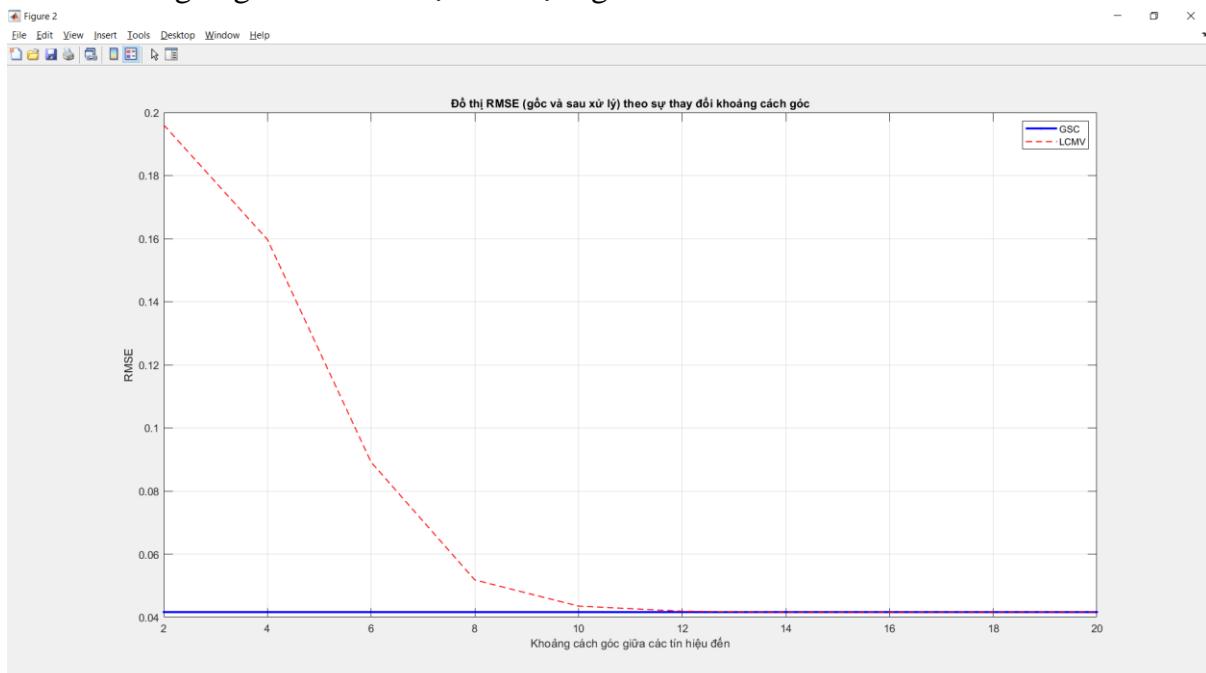
- ❖ RMSE (tín hiệu gốc và sau khi xử lý) của GSC và LCMV khi thay đổi phương sai nhiễu:



Hình 84. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý của GSC và LCMV theo sự thay đổi phương sai nhiễu.

Nhân xét: Khi tăng phương sai nhiễu, sự sai lệch giữa tín hiệu tại ngõ ra sau khi xử lý và tín hiệu trộn tại ngõ vào cũng tăng, biểu thị qua việc RMSE tăng dần. Tuy nhiên, giải thuật GSC cho RMSE tăng ít hơn so với LCMV, cho thấy giải thuật này tối ưu hơn.

- ❖ RMSE (tín hiệu gốc và sau khi xử lý) của GSC và LCMV khi thay đổi khoảng cách góc giữa các tín hiệu đến tại ngõ vào:

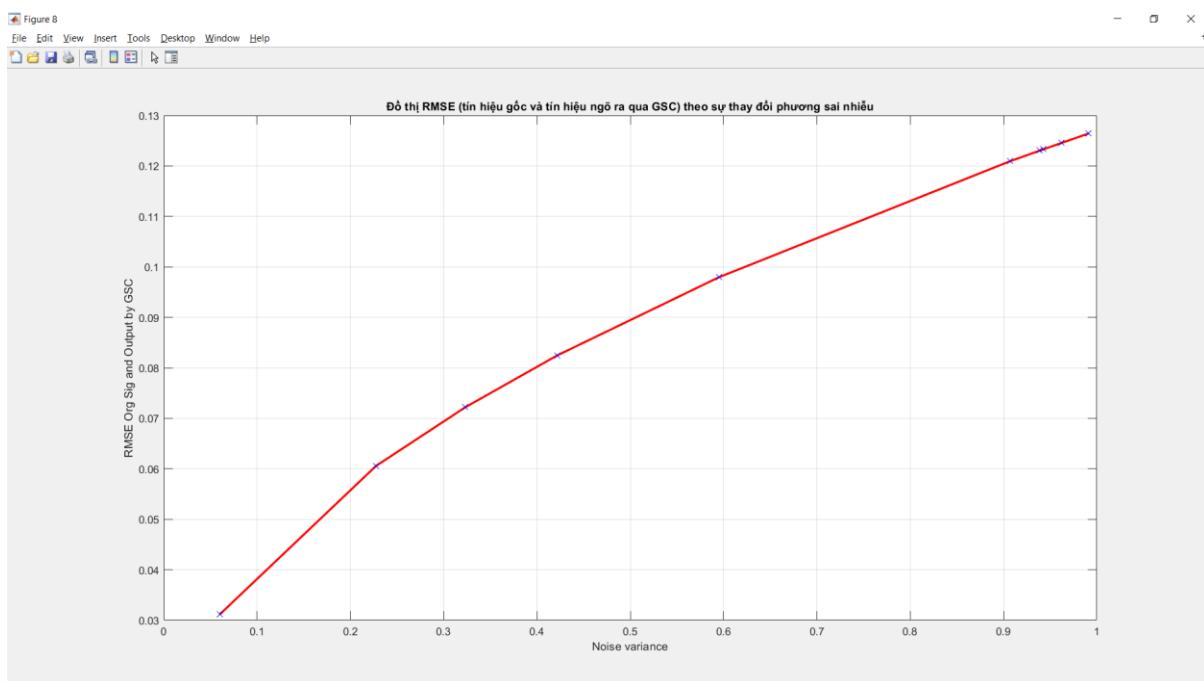


Hình 85. RMSE giữa tín hiệu gốc và tín hiệu sau khi xử lý của GSC và LCMV theo sự thay đổi khoảng cách góc giữa các tín hiệu ngõ vào.

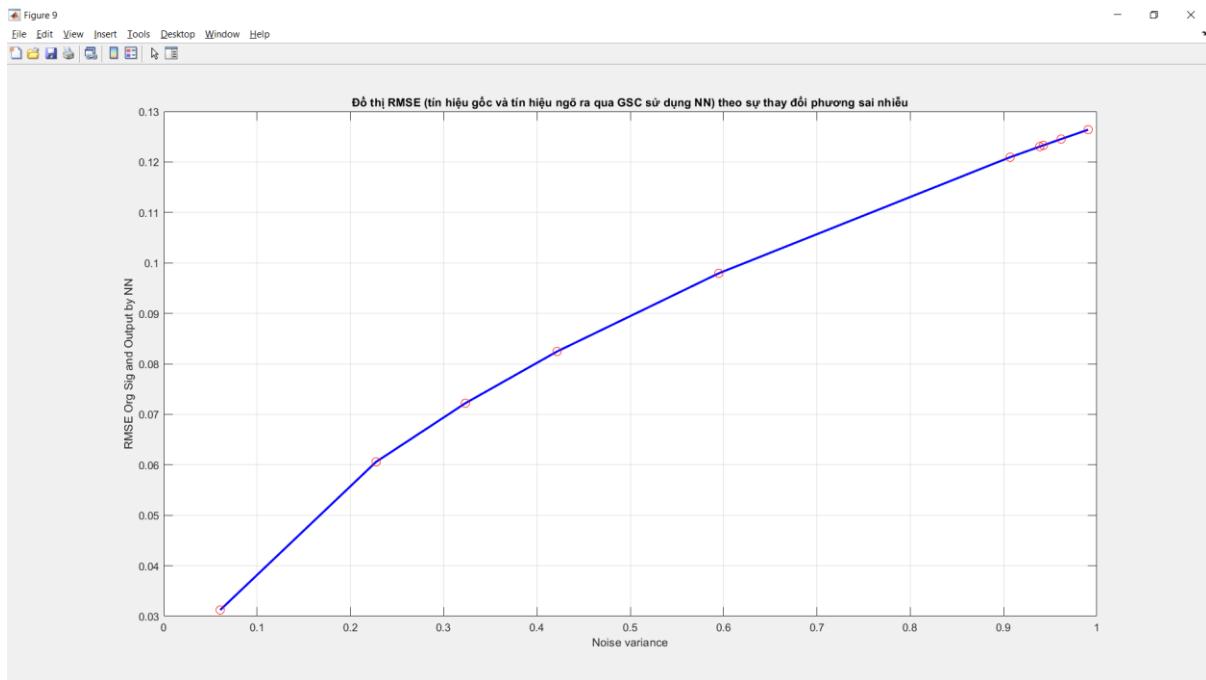
Nhân xét: Khi khoảng cách góc giữa các tín hiệu tại ngõ vào giảm dần, việc loại bỏ can nhiễu và cho ra tín hiệu tương đồng với tín hiệu mong muốn càng khó hơn, biểu thị qua sự khác biệt giữa tín hiệu tại ngõ ra và tín hiệu mong muốn gốc. Với GSC, sự khác biệt này luôn rất nhỏ và ổn định ($RMSE = 0.041618$) dù giảm dần khoảng cách góc đi rất nhỏ, trong khi với giải thuật LCMV, sự khác biệt này bắt đầu tăng lên khi khoảng cách góc càng nhỏ đi, đặc biệt khi khoảng cách góc nhỏ hơn 10° . Điều này cho thấy GSC là giải thuật tối ưu hơn cho việc xử lý tín hiệu.

III.3.2. Thực hiện so sánh “Bộ triệt các búp bên tổng quát” được cải tiến với mạng Neural và truyền thống theo 10 phương sai nhiễu bất kỳ (nằm trong khoảng $0 \div 1$):

- ❖ $RMSE$ của tín hiệu gốc với tín hiệu qua GSC truyền thống và GSC được cải tiến với mạng Neural:



Hình 86. $RMSE$ (tín hiệu gốc và tín hiệu tại ngõ ra GSC truyền thống) theo sự thay đổi phương sai nhiễu.

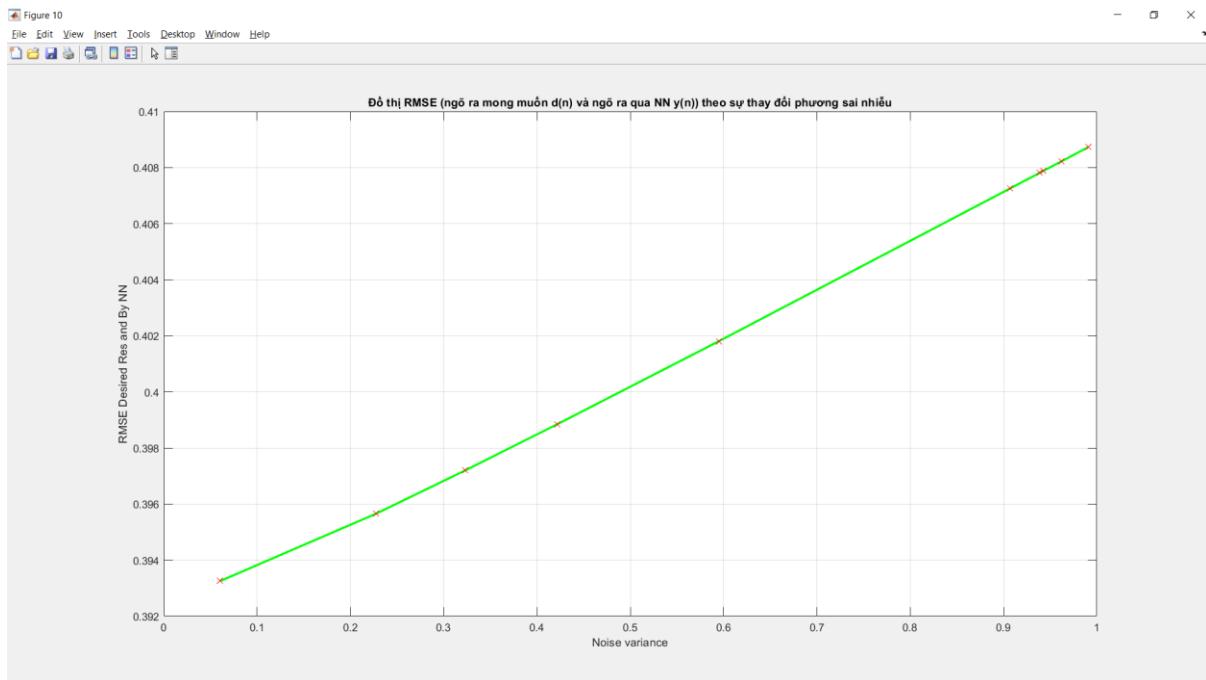


Hình 87. RMSE (tín hiệu gốc và tín hiệu tại ngõ ra GSC được cải tiến với mạng Neural) theo sự thay đổi phương sai nhiễu.

```
Command Window
> In testfornewy (Line 266)
Rmse for nv = 0.1 by GSC using NN is: 0.040156
Test for 10 new random noise variance
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.06048 là: 0.031229
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.06048 là: 0.39325
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.06048 là: 0.031229
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.22754 là: 0.060574
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.22754 là: 0.39566
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.22754 là: 0.060574
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.32325 là: 0.072198
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.32325 là: 0.3972
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.32325 là: 0.072198
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.42149 là: 0.082442
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.42149 là: 0.39884
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.42149 là: 0.082443
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.59478 là: 0.097934
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.59478 là: 0.40181
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.59478 là: 0.097934
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.90701 là: 0.12094
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.90701 là: 0.40725
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.90701 là: 0.12094
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.9388 là: 0.12304
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.9388 là: 0.40781
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.9388 là: 0.12304
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.94274 là: 0.1233
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.94274 là: 0.40788
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.94274 là: 0.1233
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.96179 là: 0.12454
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.96179 là: 0.40822
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.96179 là: 0.12454
-----
Rmse tin hiệu gốc và tín hiệu tại ngõ ra với phương sai nhiễu 0.9907 là: 0.12639
RMSE ngõ ra mong muốn (n) và ngõ ra qua NN y(n) cho phương sai nhiễu 0.9907 là: 0.40873
Rmse tin hiệu gốc và tín hiệu tại ngõ ra qua NN với phương sai nhiễu 0.9907 là: 0.12639
```

Hình 88. RMSE (tín hiệu gốc và tín hiệu tại ngõ ra GSC truyền thống, GSC được cải tiến với mạng Neural) theo sự thay đổi phương sai nhiễu.

- ❖ RMSE của ngõ ra mong muốn $d(n)$ và ngõ ra qua mạng Neural $y(n)$ (hình 10) theo sự thay đổi phương sai nhiễu:



Hình 89. RMSE (ngõ ra mong muốn và ngõ ra qua mạng Neural) theo sự thay đổi phương sai nhiễu.

Nhân xét: Khi tăng phương sai nhiễu thì RMSE của ngõ ra mong muốn $d(n)$ và ngõ ra qua mạng Neural $y(n)$ cũng tăng lên. Điều này cũng tương đồng với việc khi phương sai nhiễu tăng, tín hiệu tại ngõ ra của bộ lọc cũng khác biệt nhiều hơn so với tín hiệu gốc mong muốn thu được.

III.3.3. Kết luận chung về các giải thuật:

Bản chất của giải thuật LCMV là giảm thiểu công suất ngõ ra trung bình, tuân theo một tập hợp các ràng buộc tuyến tính trên vector trọng số. Các ràng buộc được đặt ra để ngăn vector trọng số hủy bỏ tín hiệu mong muốn. Để đáp ứng yêu cầu của nhiều ràng buộc, có thể sử dụng bộ triết các búp sóng bên tống quát (GSC) có vector trọng số được tách thành hai thành phần:

- Một vector trọng số tĩnh, thỏa mãn các ràng buộc quy định.
- Một vector trọng số không bị ràng buộc, việc tối ưu hóa vector trọng số này, theo lý thuyết bộ lọc Wiener, sẽ giảm thiểu ảnh hưởng của nhiễu tại ngõ vào phía thu và các tín hiệu can nhiễu.

Khi so sánh giữa hai giải thuật, dựa vào việc thay đổi số lượng phần tử M , khoảng cách góc giữa các hướng tới của tín hiệu mong muốn và các can nhiễu, phương sai của nhiễu tai ngõ vào, có thể dễ dàng nhận thấy giải thuật GSC sẽ tối ưu tín hiệu mong muốn tốt hơn với bản chất giải thuật là triết tiêu hoàn toàn các búp phụ, qua đó ngõ ra chỉ xuất hiện tín hiệu tương đồng với tín hiệu mong muốn thu được, trong khi LCMV là triết tiêu hết mọi búp và tránh triết tiêu tại hướng tới mong muốn dựa vào ràng buộc tuyến tính tại ngõ vào.[1]

Mở rộng giải thuật GSC với dãy an-ten 2 chiều để phù hợp với các ứng dụng thực tế. Kết quả vẫn cho thấy GSC là giải thuật tối ưu tín hiệu mong muốn và triệt tiêu can nhiễu rất tốt với các thông số kết quả tương đồng như với dãy an-ten 1 chiều. Ngoài ra, trong thực tế, dữ liệu ngõ vào tại các bộ thu rất đa dạng, có thể chứa không chỉ 1 mà là nhiều tín hiệu mong muốn cùng với số can nhiễu tăng lên gấp bội và đi kèm là sự biến thiên của nhiễu. Việc cải tiến giải thuật với mạng Neural đã cho thấy kết quả khả quan với độ chính xác tương đồng và khả năng tính toán dữ liệu ngõ vào cực lớn, mở đường cho các nghiên cứu và ứng dụng thực tiễn khác nhau.[2]

KẾT LUẬN

Bộ lọc thích nghi tuyển tính luôn là một vấn đề quan trọng trong xử lý tín hiệu số. Với khả năng xử lý các tín hiệu tới thay đổi liên tục tốt hơn rất nhiều so với các phương pháp truyền thống nhờ vào việc điều chỉnh bộ trọng số thích hợp theo các tiêu chí và ràng buộc mong muốn, bộ lọc thích nghi đã được ứng dụng thành công trong hệ thống truyền thông của nhiều lĩnh vực đa dạng như thông tin liên lạc, điều khiển, radar, sóng siêu âm, địa chấn và kỹ thuật y sinh, cùng nhiều lĩnh vực khác.

Sau khoảng thời gian nghiên cứu, đồ án đã tóm gọn đầy đủ về cơ sở lý thuyết bộ lọc tuyển tính thời gian rời rạc, qua đó phát triển hai thuật toán tối ưu việc thu tín hiệu mong muốn và triệt các can nhiễu là “bộ lọc tối thiểu phương sai có ràng buộc tuyển tính LCMV” và “bộ triệt các búp bên tổng quát GSC” đối với dãy an-ten một chiều, mở rộng giải thuật tối ưu hơn là GSC đối với dãy an-ten hai chiều và cải tiến với mạng Neural. Với góc nhìn bao quát và đầy đủ về hai thuật toán trên cùng phần mở rộng của nó, tùy vào các điều kiện khác nhau mà chúng ta có thể chọn được thuật toán phù hợp nhằm tối ưu cho việc thiết kế hệ thống an-ten cùng các ứng dụng của nó.

Với các ưu điểm của mình, bộ lọc thích nghi tuyển tính sử dụng các giải thuật tối ưu tín hiệu mong muốn đã được ứng dụng trong nhiều lĩnh vực, đặc biệt là trong các hệ thống truyền thông không dây, góp phần đưa nền công nghệ thế giới ngày càng phát triển, đến gần hơn với công nghệ 5G và xa hơn nữa.

Mặc dù đã rất cố gắng trong việc nghiên cứu và thực hiện đồ án, nhưng với thời gian và hiểu biết còn hạn chế nên đồ án chỉ dừng lại ở mức tìm hiểu về bộ lọc thích nghi tuyển tính thời gian rời rạc với trọng tâm là phát triển hai thuật toán tối ưu tín hiệu mong muốn và mở rộng, cải tiến với mạng Neural, chưa kịp triển khai ứng dụng vào thực tế bởi việc này cần thêm nhiều kiến thức, hiểu biết ở các lĩnh vực công nghệ khác nhau. Đồng thời, đồ án chắc chắn cũng không tránh khỏi những thiếu sót, nên em rất mong nhận được ý kiến đóng góp từ thầy.

Em xin chân thành cảm ơn!

TÀI LIỆU THAM KHẢO

- [1] Simon Haykin, “Adaptive Filter Theory”, 5th edition, Pearson Education, 2014.
- [2] Simon Haykin, “Neural Networks and Learning Machines, 3rd edition, Pearson Education, 2009.
- [3] Tiep Vu Huu, “Multi-layer Perceptron và Backpropagation”,
<https://machinelearningcoban.com/2017/02/24/mlp/>, 24/02/2017.

PHỤ LỤC

- ❖ *Code thực hiện cho giải thuật GSC:*

```
%testGSC
%dữ liệu nhập vào cho lập trình
M = 20;
g = [0;0;0];
theta = [-60;-50;-40];
theta0 = -60;
sv = [1, 1, 1];
nv = 1;
for i=1:length(theta)
    if theta(i)==theta0
        chiso = i;
    end;
end;
g(chiso) = 1;
%lấy tín hiệu âm thanh vào
[data, fs] = audioread('nhac.mp3');
[tapam1, fs1] = audioread('tapam1.mp3');
[tapam2, fs2] = audioread('tapam2.mp3');

%xác định N
N = length(data);

%xử lý tín hiệu âm thanh vào
kenhtraidata = data(:, 1); %chỉ lấy kênh trái của data
kenhtraitapam1 = tapam1(:, 1);
kenhtraitapam2 = tapam2(:, 1);

%đưa tín hiệu các âm thanh về cùng độ dài snapshot N
kenhtraidata = kenhtraidata(1:N);
kenhtraitapam1 = kenhtraitapam1(1:N);
kenhtraitapam2 = kenhtraitapam2(1:N);
input =
[transpose(kenhtraidata);transpose(kenhtraitapam1);transpose(kenhtraitapam2)];

%thu nhiễu
nhieu = audioread('tiengmuaro.mp3');
kenhtrainhieu = nhieu(:,1);
kenhtrainhieu = kenhtrainhieu(1:N);
nhieu = transpose(kenhtrainhieu);
giatrichuanhoa = max(abs(nhieu));
nhieuchuanhoa = nhieu ./ giatrichuanhoa;

d=length(theta);
C = []; %ma trận C của vector lái (ma trận ràng buộc tuyến tính)
A=[];
a=[1];
a0=[1];
U=[];
Utotal=zeros(N,M);
Mx=zeros(N,1);
Rs=zeros(M);%signal correlation matrix
if nv==0
    nv1=0.00001;
else
```

```

        nv1=nv;
end;
noise=sqrt(nv1).*nhieuchuanhoa;%noise
Rn=nv1*eye(M);%noise correlation matrix
delta=0.5;% inter-element spacing over wavelength

%Find steering vectors and correlation matices
for jj=1:d
    for i=1:(M-1)
        a=[a exp(-1j*i*2*pi*delta*sin(theta(jj)*pi/180))];%steering vector
    end
if jj == chiso
    vectorlai = a;
    disp('Âm thanh cần phải thu');
    soundsc(transpose(input(jj, :)),fs);
    %biểu diễn tín hiệu gốc data
    figure;
    t = (1:N)/fs;
    plot(t, transpose(input(jj, :)), 'color', 'b');
    xlabel('Thời gian (giây)');
    ylabel('Biên độ');
    title('Đồ thị tín hiệu âm thanh cần thu');
    grid;
end;
u= sqrt(sv(jj)).*input(jj, :);
u = imresize(u, [1, N]);
U=[U,transpose(u)];%matrix of beamformer inputs
Mx=Mx+transpose(u+noise);%mixed beamformer inputs plus noise
uu=transpose(u+noise)*a;%outter product giving a NxM matrix
Utotal=Utotal+uu;
a=transpose(a);
C = [C,a];
Rs=Rs+[a*sv(jj)*ctranspose(a)];
a=[1];
end;
%tin hieu tron lai mix
mix = transpose(Mx);
pause(10);
disp('Âm thanh tại ngõ tới có cả nhiễu và can nhiễu');
soundsc(real(mix), fs);
figure;
t = (1:N)/fs;
plot(t, mix, 'color', 'r');
xlabel('Thời gian (giây)');
ylabel('Biên độ');
title('Đồ thị tín hiệu âm thanh trộn với tạp âm và nhiễu');
grid;
%Find total correlation matrix
R=Rs+Rn;%total correlation matrix

%Tìm ma trận Ca:
Ca = null(ctranspose(C));

%Find optimum weight vector
wq = C*pinv(ctranspose(C)*C)*g;
wao = pinv(ctranspose(Ca)*R*Ca)*ctranspose(Ca)*R*wq;
wopt = wq - Ca*wao;

%Plot beam-pattern

```

```

aa=[1];
r=0.1; %Angle resolution, [deg.]
Cout = [];
for k=1:180/r;
th(k)=-90+k*r;
for i=1:(M-1)
aa=[aa exp(-1j*i*2*pi*delta*sin(th(k)*pi/180))];%steering vector
end
aa=transpose(aa);
Cout = [Cout, aa];
aa=[1];
end;
g_out = ctranspose(Cout)*wopt;
figure;
plot([-90:r:90],(abs(g_out).^2),'b','LineWidth',1.5);
xlabel('Angle [degree]');ylabel('Đồ thị tuyến tính');
title(['Đồ thị tuyến tính of GSC Beamformer. Number of antennas: ',num2str(M),'. Signal variance: ', num2str(sv), '. Noise variance: ', num2str(nv), '. AoA: -60, -50, -40']);
grid;
%Plot beamformer output
yy=[];
y=ctranspose(wopt)*transpose(Utotal);
yy=[yy,y];
k=find(theta==theta0);
startindex = floor(N/2);
endindex = startindex + 200;
yyplot = yy(startindex:endindex);
Mxplot = Mx(startindex:endindex);
signaloutput = U(:,k);
signaloutputplot = signaloutput(startindex:endindex);
figure;
plot(abs(yyplot),'b','LineWidth',1.5);hold;
plot(abs(signaloutputplot),'g--','LineWidth',2);
plot(abs(Mxplot),'r.-','LineWidth',1);hold
xlabel('Time samples');ylabel('Amplitude');title(['GSC Beamformer output. Number of antennas: ',num2str(M),'. Signal variance: ', num2str(sv), '. Noise variance: ', num2str(nv), '. AoA: -60, -50, -40']);
legend('Beamformer output','Desired signal','Beamformer input');
grid;
plot output signal
figure;
t = (1:N)/fs;
plot(t, transpose(yy), 'color', 'g');
xlabel('Thời gian (giây)');
ylabel('Biên độ');
title('Đồ thị tín hiệu âm thanh sau khi xử lý');
grid;
soundgora = real(transpose(yy));
pause(10);
disp('Âm thanh sau khi xử lý');
soundsc(soundgora, fs);
%Calculate Root Mean Square Error
output = yy;
desired = U(:,k);
e = output - transpose(desired);
MSE = abs(e).^2;
rmse = sqrt(sum(MSE)*(1/N));
disp(['rmse for nv = ', num2str(nv), ' is: ', num2str(rmse)]);
disp('Kết thúc xử lý');

```

❖ *Code thực hiện cho việc huấn luyện và kiểm tra mạng Neural:*

```
%testGSCusingNN
%dữ liệu nhập vào cho lập trình
M = 4;
g = [0;0;0];
phi = [-60;-150;120];
phi = phi * pi/180;
phi0 = -60*pi/180;
theta = [30;50;80];
theta = theta * pi/180;
theta0 = 30*pi/180;
sv = [1, 1, 1];
NN_input = [];
DESIRED_output = [];
L = length(phi); %number of coming signals
for i=1:length(phi)
    if phi(i)==phi0 && theta(i)==theta0
        chiso = i;
    end;
end;
g(chiso) = 1;
%lấy tín hiệu âm thanh vào
[data, fs] = audioread('nhac.mp3');
[tapam1, fs1] = audioread('tapam1.mp3');
[tapam2, fs2] = audioread('tapam2.mp3');
data = data(1:length(data)/40,:);

%xác định N
N = length(data);
M_new = M * M;
N_new = N * (M_new - L);

%xử lý tín hiệu âm thanh vào
kenhtraidata = data(:, 1); %chỉ lấy kênh trái của data
kenhtraitapam1 = tapam1(:, 1);
kenhtraitapam2 = tapam2(:, 1);

%đưa tín hiệu các âm thanh về cùng độ dài snapshot N
kenhtraidata = kenhtraidata(1:N);
kenhtraitapam1 = kenhtraitapam1(1:N);
kenhtraitapam2 = kenhtraitapam2(1:N);
input =
[transpose(kenhtraidata); transpose(kenhtraitapam1); transpose(kenhtraitapam2)];

%thu nhiễu
nhieu = audioread('tiengmuaro.mp3');
kenhtrainhieu = nhieu(:,1);
kenhtrainhieu = kenhtrainhieu(1:N);
nhieu = transpose(kenhtrainhieu);
giatrichuanhoa = max(abs(nhieu));
nhieuchuanhoa = nhieu ./ giatrichuanhoa;
for p = 1:200
    nv = 0.005*p;
    %coding
    d=length(phi);
    C = []; %ma trận C của vector lái (ma trận ràng buộc tuyến tính)
    A=[];
    a=[1];
```

```

a0=[1];
U=[];
Utotal=zeros(N,M_new);
Mx=zeros(N,1);
Rs=zeros(M_new);%signal correlation matrix
if nv==0
    nv1=0.00001;
else
    nv1=nv;
end;
noise=sqrt(nv1).*nhieuchuanhoa;%noise
Rn=nv1*eye(M_new);%noise correlation matrix
delta=0.5;% inter-element spacing over wavelength

x_i = zeros(d,1);
y_i = zeros(d,1);
nuy_i = zeros(d,1);
v_i = zeros(d,1);

dx = [];
dy = [];
vitri_x = 0;
vitri_y = 0;
for i = 1:M
    dx = [dx vitri_x];
    vitri_x = vitri_x + 1;
    dy = [dy (i-1)*ones(1, M)];
    i = i+1;
end;
final_dx = dx;
for i = 1:(M-1)
    dx = flip(dx);
    final_dx = [final_dx dx];
    i = i + 1;
end;
dx = dy;
dy = final_dx;
D = vertcat(dx, dy);
dx = dx*delta;
dy = dy*delta;

%Find steering vectors and correlation matrices
for jj=1:d
    x_i(jj) = cos(phi(jj))*sin(theta(jj));
    y_i(jj) = sin(phi(jj))*sin(theta(jj));
    nuy_i(jj) = -2*pi*x_i(jj);
    v_i(jj) = -2*pi*y_i(jj);
    for i=1:(M_new - 1)
        a=[a exp(1j*(dx(i+1)*nuy_i(jj) + dy(i+1)*v_i(jj)))];%steering vector
    end;
    if jj == chiso
        vectorlai = a;
    end;
    u= sqrt(sv(jj)).*input(jj, :);
    u = imresize(u, [1, N]);
    U=[U,transpose(u)];%matrix of beamformer inputs
    Mx=Mx+transpose(u+noise);%mixed beamformer inputs plus noise
    uu=transpose(u+noise)*a;%outer product giving a M_new x N matrix
    Utotal=Utotal+uu;
end;

```

```

a=transpose(a);
C = [C,a];
Rs=Rs+[a*sv(jj)*ctranspose(a)];
a=[1];
end;
%tin hieu tron lai mix
mix = transpose(Mx);

%Find total correlation matrix
R=Rs+Rn;%total correlation matrix

%Tìm ma trận Ca:
Ca = null(ctranspose(C));

%Find optimum weight vector
wq = C*pinv(ctranspose(C)*C)*g;
wao = pinv(ctranspose(Ca)*R*Ca)*ctranspose(Ca)*R*wq;
wopt = wq - Ca*wao;
nn_input = reshape(ctranspose(Ca)*transpose(Utotal), [N_new 1]);
desired_output = ctranspose(wq)*transpose(Utotal);
NN_input = [NN_input nn_input];
DESIRED_output = [DESIRED_output desired_output];
end

NN_input = transpose(NN_input);
DESIRED_output = transpose(DESIRED_output);

%Initiate training and testing data
[rows, columns] = size(NN_input);
num_train = 0.5;
num_test = 1-num_train;
num_training_rows = round(num_train * rows);
training_indices = randperm(rows, num_training_rows);
test_indices = setdiff(1:rows, training_indices);

NN_input_training = NN_input(training_indices, :);
NN_input_test = NN_input(test_indices, :);
DESIRED_output_training = DESIRED_output(training_indices, :);
DESIRED_output_test = DESIRED_output(test_indices, :);

disp(['Starting calculate using neural network']);
tic;

%Building Neural Network
n_i = N_new;
n_h = 40;
n_o = N;

W1 = randn(n_i, n_h);
b1 = randn(1, n_h);
W2 = randn(n_h, n_o);
b2 = randn(1, n_o);

epochs = 5000;
learning_rate = 0.1;
costs = []; %list of costs for each epoch
Errors = [];

for e = 1:epochs

```

```

%feed foward
hidden_layer_input = NN_input_training * W1 + b1;
hidden_layer_output = sigmoid(hidden_layer_input); % f(x) = sigmoid(x);
output_layer_input = hidden_layer_output * W2 + b2;
predicted_output = sigmoid(output_layer_input);

%cost function
Errors = abs(DESIRED_output_training - predicted_output);
squarederrors = (Errors).^2;
sumsquareerrors = sum(squarederrors(:));
mse = sumsquareerrors / numel(Errors);
cost = sqrt(mse); % rmse of errors
if mod(e, 200) == 0
    disp(['rmse for epoch ', num2str(e), ' is: ', num2str(cost)]);
end
if cost <= 1e-6
    cost = 0;
    disp(['rmse for epoch ', num2str(e), ' of n_h ', num2str(n_h), ' is: ', num2str(cost)]);
    costs = [costs cost];
    break;
end
costs = [costs cost];

%back propagation
error = DESIRED_output_training - predicted_output;
d_predicted_output = error .* ((sigmoid(predicted_output) .* (1 - sigmoid(predicted_output)))) % f'(x) = f(x)*(1-f(x))

error_hidden_layer = d_predicted_output * transpose(W2);
d_hidden_layer = error_hidden_layer .* ((sigmoid(hidden_layer_output) .* (1 - sigmoid(hidden_layer_output))));

%update weights and bias
W2 = W2 + transpose(hidden_layer_output) * d_predicted_output * learning_rate;
b2 = b2 + sum(d_predicted_output) * learning_rate;
W1 = W1 + transpose(NN_input_training) * d_hidden_layer * learning_rate;
b1 = b1 + sum(d_hidden_layer) * learning_rate;
end
elapsedTime = toc;
disp(['Thời gian tính toán NN : ', num2str(elapsedTime), ' giây.']);

%calculate for testing data
z1 = NN_input_test * W1 + b1;
a1 = sigmoid(z1);
z2 = a1 * W2 + b2;
predicted_test = sigmoid(z2);
Errors_test = abs(DESIRED_output_test - predicted_test);
squarederrors_test = (Errors_test).^2;
sumsquareerrors_test = sum(squarederrors_test(:));
mse_test = sumsquareerrors_test / numel(Errors_test);
cost_test = sqrt(mse_test); % rmse of errors
disp(['RMSE khi dùng NN cho tập test: ', num2str(cost_test)]);

```