

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



ĐỖ HẢI SƠN

**XÂY DỰNG HỆ THỐNG XÁC ĐỊNH HƯỚNG SÓNG ĐEN
SỬ DỤNG THUẬT TOÁN MUSIC TRÊN THIẾT BỊ SDR**

**KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ kỹ thuật điện tử, Truyền thông**

HÀ NỘI - 2020

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

ĐỖ HẢI SƠN

**XÂY DỰNG HỆ THỐNG XÁC ĐỊNH HƯỚNG SÓNG ĐẾN
SỬ DỤNG THUẬT TOÁN MUSIC TRÊN THIẾT BỊ SDR**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY
Ngành: Công nghệ kỹ thuật điện tử, Truyền thông

Cán bộ hướng dẫn: TS. Trần Thị Thúy Quỳnh

HÀ NỘI - 2020

LỜI CAM ĐOAN

Tôi xin cam đoan khóa luận tốt nghiệp **Xây dựng hệ thống xác định hướng sóng đến sử dụng thuật toán MUSIC trên thiết bị SDR** là công trình nghiên cứu thực sự của tôi, được thực hiện dựa trên cơ sở lý thuyết, kiến thức chuyên ngành dưới sự hướng dẫn khoa học của TS. Trần Thị Thúy Quỳnh.

Tôi xin cam đoan những công việc trong khóa luận thực hiện chưa từng được các tác giả khác đề xuất. Với sự hiểu biết của mình, tôi chắc chắn các số liệu, kết quả trong khóa luận là trung thực và chưa được công bố ở đâu và trong bất cứ công trình nào trừ công trình của tác giả và tài liệu tham khảo.

Nếu có gì sai trái, tôi xin hoàn toàn chịu trách nhiệm.

Hà Nội, ngày ... tháng ... năm 2020

Sinh viên

Đỗ Hải Sơn

LỜI CẢM ƠN

Tôi xin chân thành cảm ơn TS. Trần Thị Thúy Quỳnh, người đã dẫn dắt, định hướng cho tôi trong những nghiên cứu về hệ thống tìm hướng sóng đến. Tôi cũng xin cảm ơn các thầy, cô Khoa Điện tử - Viễn thông và các thầy cô phản biện đã cho nhiều góp ý giúp tôi khắc phục những điểm còn hạn chế trong khóa luận.

Tôi cũng xin cảm ơn bạn Trần Đức Mạnh K61DB cùng các bạn trong bộ môn Thông tin Vô tuyến đã hỗ trợ tôi rất nhiều trong việc thực nghiệm hệ thống.

Hà Nội, ngày ... tháng ... năm 2020

Sinh viên

Đỗ Hải Sơn

TÓM TẮT

Tóm tắt: Hệ tìm phương, hay còn gọi là tìm hướng sóng đến (DOA), luôn đóng vai trò quan trọng trong các ứng dụng: thông tin, định vị, giám sát, dẫn đường, tìm kiếm cứu nạn,... Với sự phát triển vượt bậc của xử lý tín hiệu, hệ tìm phương sử dụng thuật toán MUSIC cho phép xác định hướng sóng đến của nhiều nguồn phát ở độ chính xác, độ phân giải cao, cấu trúc mảng anten tùy ý, điều mà các hệ tìm phương truyền thống không thể. Các thiết bị vô tuyến định nghĩa bằng phần mềm SDR kết hợp với phần mềm GNU Radio được mong đợi sẽ trở thành công nghệ thống trị truyền thông vô tuyến trong tương lai. Bắt kịp xu thế đó, trong khóa luận này sẽ tập trung tìm hiểu về phần cứng và phần mềm cho SDR: BladeRF x115, lập trình các khồi phục vụ DOA trên GNU Radio. Qua đó triển khai thuật toán tìm phương MUSIC trên hệ BladeRF để kiểm nghiệm khả năng đồng bộ và kết quả ước lượng hướng sóng đến của hệ.

Từ khóa: DOA, MUSIC, SDR, GNU Radio.

MỤC LỤC

Lời cam đoan	
Lời cảm ơn	
Tóm tắt	
Mục lục	i
Tóm tắt	iii
Danh mục hình vẽ	iv
Danh mục bảng biểu	v
Mở đầu	1
CHƯƠNG 1. TỔNG QUAN VỀ CÁC HỆ THỐNG XÁC ĐỊNH HƯỚNG SÓNG ĐẾN	4
1.1. Khái niệm và ứng dụng của hệ xác định hướng sóng đến	4
1.2. Các công nghệ xác định hướng sóng đến điển hình	5
1.2.1. Phương pháp tạo búp sóng	5
1.2.2. Thuật toán Maximum Likelihood	9
1.3. Hệ thống xác định hướng sóng đến với thuật toán MUSIC	9
CHƯƠNG 2. TRIỂN KHAI THUẬT TOÁN MUSIC TRÊN THIẾT BỊ SDR	14
2.1. Phần mềm GNU Radio và thiết bị vô tuyến định nghĩa bằng phần mềm SDR	14
2.1.1. Phần mềm GNU Radio	14
2.1.2. Thiết bị vô tuyến định nghĩa bằng phần mềm SDR	16
2.2. Phần cứng và phần mềm của BladeRF	20
2.2.1. BladeRF x115	20
2.2.2. Phần mềm của BladeRF	21
2.3. Triển khai hệ thống xác định hướng sóng đến trên hệ BladeRF	24
CHƯƠNG 3. KẾT QUẢ MÔ PHỎNG, THỰC NGHIỆM HỆ THỐNG	35
3.1. Mô phỏng hệ thống trên GNU Radio	35
3.1.1. Mô phỏng với dữ liệu Matlab	35
3.1.2. Mô phỏng với dữ liệu giả lập BladeRF	37
3.2. Thực nghiệm hệ thống	39
3.2.1. Tín hiệu nguồn phát và thiết lập hệ thu	39
3.2.2. Tính huống thực nghiệm	40

3.2.3. Kết quả thực nghiệm	42
KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU TIẾP THEO	48
PHỤ LỤC	50
TÀI LIỆU THAM KHẢO	56

TÓM TẮT

Tóm tắt: Hệ tìm phương, hay còn gọi là tìm hướng sóng đến (DOA), luôn đóng vai trò quan trọng trong các ứng dụng: thông tin, định vị, giám sát, dẫn đường, tìm kiếm cứu nạn,... Với sự phát triển vượt bậc của xử lý tín hiệu, hệ tìm phương sử dụng thuật toán MUSIC cho phép xác định hướng sóng đến của nhiều nguồn phát ở độ chính xác, độ phân giải cao, cấu trúc mảng anten tùy ý, điều mà các hệ tìm phương truyền thống không thể. Các thiết bị vô tuyến định nghĩa bằng phần mềm SDR kết hợp với phần mềm GNU Radio được mong đợi sẽ trở thành công nghệ thống trị truyền thông vô tuyến trong tương lai. Bắt kịp xu thế đó, trong khóa luận này sẽ tập trung tìm hiểu về phần cứng và phần mềm cho SDR: BladeRF x115, lập trình các khôi phục vụ DOA trên GNU Radio. Qua đó triển khai thuật toán tìm phương MUSIC trên hệ BladeRF để kiểm nghiệm khả năng đồng bộ và kết quả ước lượng hướng sóng đến của hệ.

Từ khóa: DOA, MUSIC, SDR, GNU Radio.

DANH MỤC HÌNH VẼ

Hình 1.1	Các phương pháp xác định hướng sóng đến	5
Hình 1.2	Ví dụ bộ tạo búp sóng tổng quát	6
Hình 1.3	Mô phỏng thuật toán Delay and Sum và Capon	8
Hình 1.4	Sơ đồ khối hệ tìm hướng sóng đến sử dụng thuật toán MUSIC	10
Hình 1.5	Mô phỏng thuật toán MUSIC và Capon	13
Hình 2.1	Giao diện GNU Radio Companion	16
Hình 2.2	Sơ đồ làm việc của GNU Radio	17
Hình 2.3	Kiểu dữ liệu và xử lý tín hiệu trong GNU Radio	17
Hình 2.4	Sơ đồ khối phần cứng hệ thống SDR	18
Hình 2.5	Các thiết bị SDR thông dụng	19
Hình 2.6	Cấu hình của BladeRF x115	21
Hình 2.7	Lấy mẫu IQ	22
Hình 2.8	Tạo khối với Python Block trên GNU Radio	23
Hình 2.9	Module tạo bằng gr_modtool	23
Hình 2.10	Xung đồng hồ tham chiếu đầu ra của BladeRF	26
Hình 2.11	Xung đồng hồ tham chiếu được chia sẻ	27
Hình 2.12	Kết quả cân chỉnh tần số với kalibrate-bladeRF	28
Hình 2.13	Kết quả thu được trên máy phân tích phổ	28
Hình 2.14	Tương quan chéo của 2 nhiễu Gauss	30
Hình 2.15	Xác định độ trễ mẫu trên GNU Radio	31
Hình 2.16	Ước lượng và phản hồi hiệu chỉnh $\text{sample}_{\text{offset}}$	32
Hình 2.17	Ước lượng và phản hồi $\text{phase}_{\text{offset}}$	33
Hình 2.18	Kết quả đồng bộ hệ SDR	34
Hình 3.1	Sơ đồ làm việc DOA với dữ liệu mô phỏng Matlab	36
Hình 3.2	Kết quả ước lượng DOA với dữ liệu mô phỏng Matlab	37
Hình 3.3	CRB của mảng anten ULA	37
Hình 3.4	RMSE: Mô phỏng dữ liệu BladeRF qua kênh truyền	39
Hình 3.5	Phổ của tín hiệu tín hiệu NBFM	40
Hình 3.6	Sơ đồ bố trí hệ DOA thực	41
Hình 3.7	So sánh phổ không gian trên thực tế và mô phỏng	42
Hình 3.8	Kết quả DOA mô phỏng khi hệ SDR đồng bộ	43
Hình 3.9	Kết quả DOA thực nghiệm khi hệ SDR đồng bộ	43
Hình 3.10	RMSE: DOA thực nghiệm và mô phỏng	44
Hình 3.11	RMSE: DOA ở từng vị trí thực nghiệm	45
Hình 3.12	Phổ tín hiệu DVB-T và NBFM thu được	46
Hình 3.13	RMSE: DOA với tín hiệu NBFM và DVB-T	47
Hình A.1	Flow-graph ước lượng DOA với dữ liệu mô phỏng Matlab trên GNU Radio	50
Hình B.1	Mô phỏng dữ liệu BladeRF qua kênh truyền	51
Hình C.1	Flow-graph truyền tín hiệu NBFM	52
Hình D.1	Flow-graph ước lượng DOA	54
Hình E.1	Flow-graph truyền tín hiệu DVB-T	55

DANH MỤC BẢNG BIỂU

Bảng 2.1 Các phương pháp sử dụng GNU Radio	18
Bảng 2.2 Thông số kỹ thuật của BladeRF x115	20
Bảng 2.3 Phương pháp đồng bộ hệ BladeRF được sử dụng trong khóa luận . .	25
Bảng 3.1 Các thông số mô phỏng hệ DOA	35
Bảng 3.2 Các thông số mô phỏng tín hiệu BladeRF qua kênh truyền	38
Bảng 3.3 Kết quả mô phỏng đồng bộ	38
Bảng 3.4 Thông số máy tính	40
Bảng 3.5 Kết quả mô phỏng đồng bộ bằng tín hiệu DVB-T	45

MỞ ĐẦU

Lý do chọn đề tài

Trong thời đại 4.0 ngày nay, việc xác định hướng đến của đối tượng là nhu cầu không thể thiếu để xây dựng các hệ thống thông minh (xe tự hành, robot tự động, kết hợp với định vị để chỉ đường,...) cũng như những mục đích về an ninh như dò những máy phá sóng dựa trên tần số máy tạo ra, hay định vị hướng tín hiệu cấp cứu tàu thuyền trên biển để ứng cứu kịp thời.

Direction of Arrival (DOA) hay ước lượng hướng sóng đến của tín hiệu tại một điểm đặt mảng anten, hệ thống DOA phải có các đặc tính: độ phân giải cao, ổn định, thích hợp với các thông số đầu vào khác nhau. Thuật toán MUSIC là một phương pháp phổ biến để ước lượng hướng sóng đến với nhiều ưu điểm về độ phân giải so với các thuật toán búp sóng cũng như độ phức tạp thuật toán ở mức trung bình, có thể xác định hướng của nhiều tín hiệu đến với nhiều kiểu điều chế khác nhau và cấu trúc mảng anten tùy ý.

Với ưu điểm truyền/nhận được nhiều loại chuẩn tín hiệu, ở tần số có thể tùy chỉnh dựa trên phần mềm nạp vào, các thiết bị vô tuyến định nghĩa bằng phần mềm SDR kết hợp với phần mềm GNU Radio được mong đợi sẽ trở thành công nghệ thống trị trong truyền thông vô tuyến. Bắt kịp xu thế đó, trong khóa luận này sẽ tập trung nghiên cứu đề tài: **Xây dựng hệ thống xác định hướng sóng đến sử dụng thuật toán MUSIC trên thiết bị SDR** nhằm thực hiện đồng bộ hệ BladeRF và triển khai xác định hướng sóng đến trên hệ BladeRF để kiểm nghiệm tính chính xác của hệ DOA trong môi trường thực tế.

Phương pháp nghiên cứu

Trong khóa luận, để đạt được mục đích nghiên cứu, sinh viên đã tìm hiểu các tài liệu, bài báo, tạp chí quốc tế,... có uy tín, thực hiện việc tính toán mô hình dữ liệu, phân tích số học để đưa ra các hướng giải quyết hợp lý, và sau đó kiểm nghiệm lại kết quả bằng hình thức mô phỏng trên Matlab, GNU Radio cuối cùng là thực nghiệm trên hệ BladeRF thực.

Cụ thể các phương pháp nghiên cứu sau đã được sử dụng trong khóa luận:

- Sử dụng thuật toán Delay and Sum, Capon và MUSIC xác định hướng sóng đến

cho mô hình tính hiệu băng hẹp.

- Sử dụng CRB: xác định ngưỡng phân giải của hệ anten ULA.
- Vận dụng tương quan chéo ước lượng độ lệch mẫu từ hệ SDR.
- Vận dụng thuật toán MUSIC tìm sự sai khác pha giữa các tín hiệu.

Nội dung nghiên cứu

- Tìm hiểu, mô phỏng và đánh giá độ phân giải của một số thuật toán ước lượng nhiều nguồn tín hiệu phổ biến, có thể áp dụng cho cấu trúc mảng tùy ý, bao gồm: CBF, Capon, MUSIC, ML.
- Tìm hiểu tổng quan về thiết bị SDR và phần mềm GNU Radio.
- Tìm hiểu về BladeRF x115 và lập trình khồi trên GNU Radio.
- Đồng bộ hệ BladeRF x115 cho việc triển khai hệ DOA.
- Kiểm nghiệm thực tế trên hệ BladeRF thực, đánh giá kết quả ước lượng DOA.

Đóng góp của đề tài

Với sự hiểu biết của sinh viên, những kết quả nghiên cứu trong khóa luận đã đạt được mục đích nghiên cứu đề ra. Những kết quả này bao gồm:

- Tìm hiểu tổng quan về các hệ thống DOA.
- Tổng quan về thiết bị SDR: BladeRF x115 và phần mềm GNU Radio.
- Các phương pháp đồng bộ hóa hệ thu SDR.
- Ứng dụng thuật toán MUSIC vào thiết bị SDR thời gian thực.
- Đánh giá, phân tích kết quả thu được với kết quả mô phỏng.

Bố cục của khóa luận

Nội dung chính của khóa luận được trình bày như sau:

- Mở đầu: Trình bày mục đích, phương pháp nghiên cứu, nội dung, đóng góp và bố cục của khóa luận.

- Chương 1: Tổng quan về các hệ thống xác định hướng sóng đến – Trình bày các công nghệ xác định hướng sóng đến điển hình hiện nay và chi tiết về hệ tìm hướng sóng đến bằng thuật toán MUSIC.
- Chương 2: Triển khai thuật toán MUSIC trên thiết bị SDR – Tìm hiểu tổng quan về phần mềm GNU Radio và thiết bị SDR, phương pháp tạo khối trên GNU Radio và giải quyết việc đồng bộ hóa SDR.
- Chương 3: Kết quả mô phỏng, thực nghiệm hệ thống – Sử dụng dữ liệu mô phỏng BladeRF trên GNU Radio xác định hướng sóng đến và thực nghiệm hệ thống với tín hiệu BladeRF thực. Qua đó so sánh, nhận xét về kết quả thu được.
- Kết luận và hướng nghiên cứu tiếp theo: Đưa ra kết luận về việc sử dụng thiết bị SDR trong việc xác định hướng sóng đến và đề xuất các giải pháp để cải thiện hệ thống.

CHƯƠNG 1

TỔNG QUAN VỀ CÁC HỆ THỐNG XÁC ĐỊNH HƯỚNG SÓNG ĐẾN

[Tổng quan về các hệ thống xác định hướng sóng đến] Tổng quan về các hệ thống xác định hướng sóng đến

Công nghệ xác định hướng sóng đến không dây có lịch sử phát triển lâu đời, bắt đầu ngay từ khi truyền thông không dây ra đời vào đầu thế kỷ 20th, và trong suốt chiều dài lịch sử, rất nhiều phương pháp đã được đề xuất. Bắt đầu với những nghiên cứu như sử dụng mảnh dao động vòng kín để truyền nhận sóng không dây cũng như xác định hướng đến bằng cách sử dụng mảng anten định hướng (dipoles, loop, ...) của E. Bellini và A. Tosi [6] vào năm 1907. Hay cuốn sách Direction and Position Finding by Wireless [22] của tác giả Ronald Keen xuất bản lần đầu năm năm 1922 trình bày việc sử dụng mảng pha để giải quyết vấn đề tìm hướng sóng đến. Rất nhiều phương pháp với các cách tiếp cận khác nhau được ra đời sau đó, mục đích chung là nâng cao tính chính xác và độ phân giải của việc xác định hướng sóng đến.

1.1. Khái niệm và ứng dụng của hệ xác định hướng sóng đến

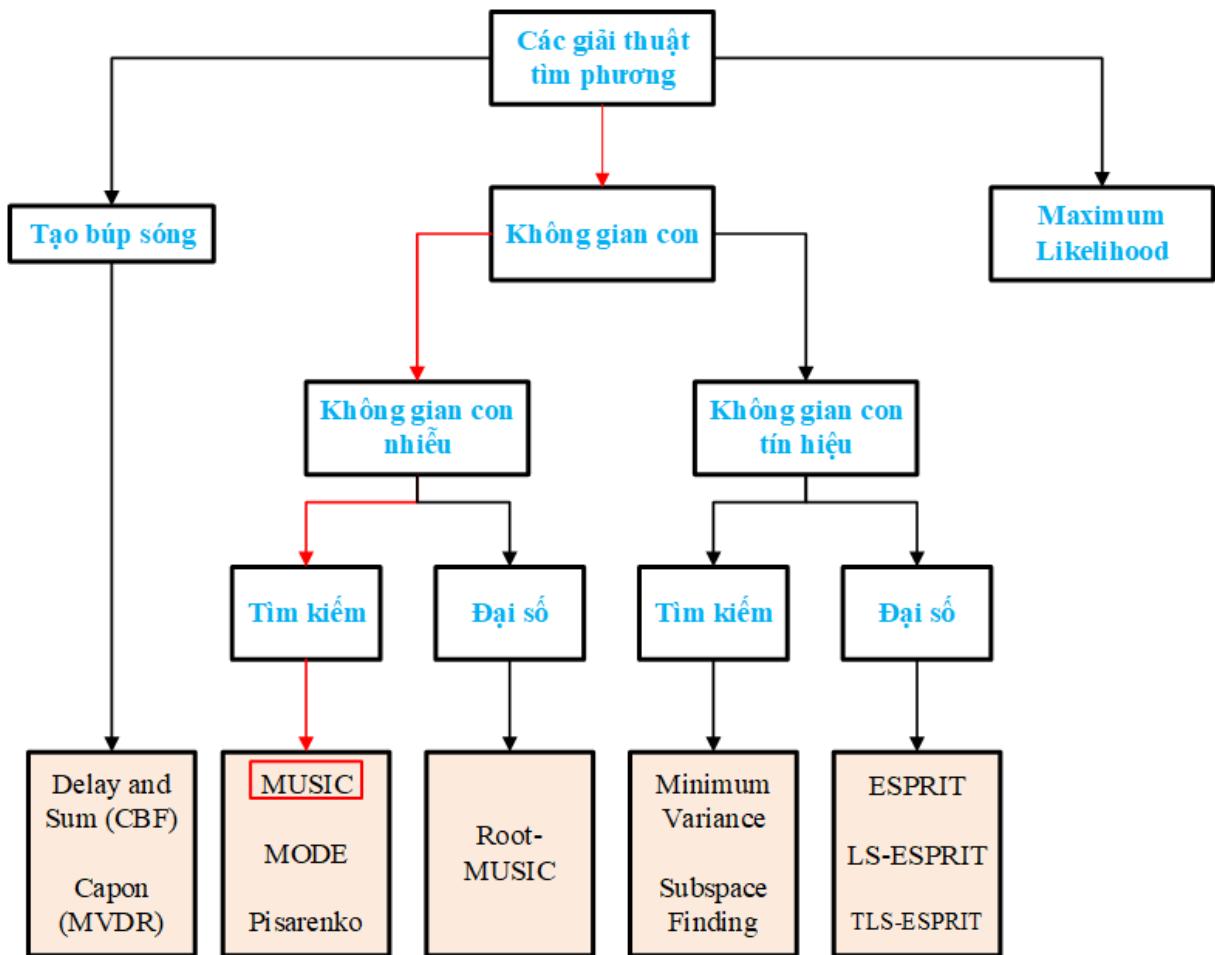
Ước lượng hướng sóng đến của tín hiệu tại một điểm đặt mảng anten phải có các đặc tính: độ phân giải cao, ổn định, thích hợp với các thông số đầu vào khác nhau.

Hiện nay, với sự phát triển của xử lý tín hiệu, các hệ tìm hướng sóng đến xử lý mảng cho phép ước lượng cùng lúc nhiều tham số: hướng sóng đến, thời gian truyền, tần số,... của nhiều tín hiệu mà các hệ tìm phương truyền thông không thể.

Trong vấn đề xử lý tín hiệu thực tế, việc ứng tính được chính xác hướng đến của tất cả các tín hiệu truyền đến mảng anten góp phần tối đa hóa hiệu suất khôi phục lại tín hiệu quan tâm và loại bỏ các tín hiệu khác can thiệp vào.

Một hệ DOA hoàn chỉnh có thể áp dụng cho nhiều mục đích khác nhau: tìm hướng nguồn phát tín hiệu với mục đích phá sóng, tìm hướng tín hiệu cứu hộ, định vị vị trí robot sử dụng trong IoT, xác định vị trí mục tiêu để giám sát hay thăm dò địa chấn, ...

1.2. Các công nghệ xác định hướng sóng đến điển hình



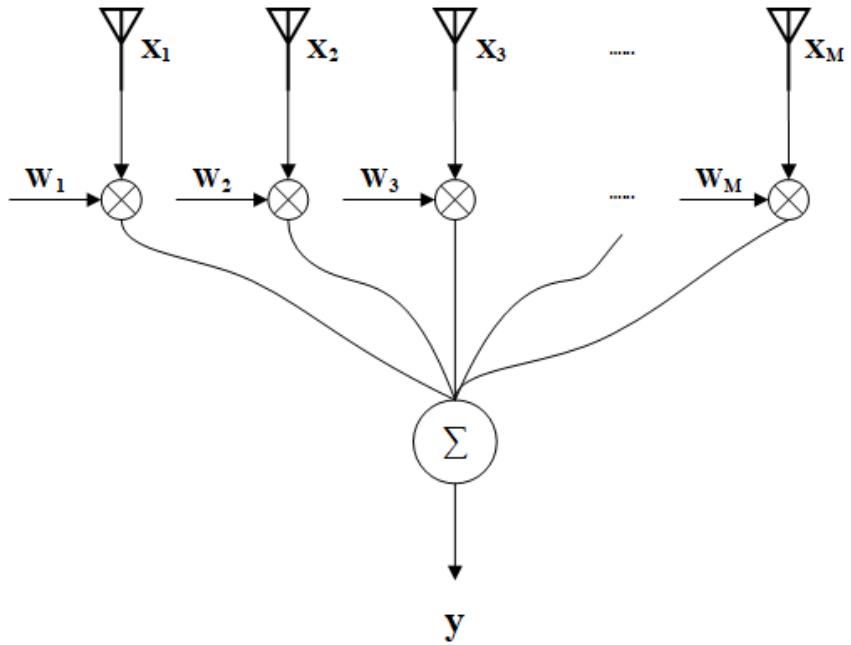
Hình 1.1. Các phương pháp xác định hướng sóng đến

Trong hình 1.1, các thuật toán tìm hướng sóng đến phổ biến hiện nay được chia vào các nhóm chính, dựa trên hướng tiếp cận để giải quyết bài toán. Mục này sẽ làm nhiệm vụ giới thiệu các thuật toán tìm hướng sóng đến điển hình, mô phỏng chúng trên Matlab, qua đó đưa ra đánh giá về các ưu nhược điểm của từng thuật toán, tập trung chủ yếu vào thuật toán MUSIC sẽ sử dụng trong Chương 3.

Phương pháp thông thường để tính toán hướng sóng đến là dựa trên ý tưởng về tạo búp sóng (beamforming), không khai thác bản chất của vector tín hiệu nhận được $\mathbf{x}(k)$ hoặc mô hình thống kê tín hiệu tạp âm. Các phương pháp thông thường được trình bày dưới đây là phương pháp tạo búp sóng cổ điển (Delay and Sum) và phương pháp phương sai tối thiểu của Capon. Cùng với đó là thuật toán ML là một phương pháp tham số tìm mô hình giống nhất với tín hiệu thu được.

1.2.1. Phương pháp tạo búp sóng

Hình 1.2 biểu diễn sơ đồ bộ tạo búp sóng tổng quát băng hẹp, với tín hiệu đầu ra $y(t)$ được tính bằng tổng của vector các trọng số $\mathbf{w} = [w_1, \dots, w_M]^T$ nhân với các dữ liệu



Hình 1.2. Ví dụ bộ tạo búp sóng tổng quát

thu thập được từ anten.

$$\mathbf{y}(t) = \mathbf{w}^H \mathbf{x}(t) \quad (1.1)$$

Công suất lối ra được tính bởi:

$$\mathbf{P}_{CBF}(\mathbf{w}) = E[|\mathbf{y}(t)|^2] = E[|\mathbf{w}^H \mathbf{x}(t)|^2] = \mathbf{w}^H E[\mathbf{x}(t)\mathbf{x}^H(t)]\mathbf{w} = \mathbf{w}^H \mathbf{R}_x \mathbf{w} \quad (1.2)$$

với \mathbf{R}_x là ma trận tự tương quan lối vào. Với K mẫu $[y_1, \dots, y_K]$, phương trình (1.2) trở thành:

$$\begin{aligned} \mathbf{P}_{CBF}(\mathbf{w}) &= \frac{1}{K} \sum_{k=1}^K |\mathbf{y}(t)|^2 = \frac{1}{K} \sum_{k=1}^K \mathbf{w}^H \mathbf{x}(k) \mathbf{x}^H(k) \mathbf{w} = \mathbf{w}^H \tilde{\mathbf{R}}_x \mathbf{w} \\ \tilde{\mathbf{R}}_x &= \frac{1}{K} \sum_{k=1}^K \mathbf{x}(k) \mathbf{x}^H(k) : \text{ ước lượng của } \mathbf{R}_x \end{aligned} \quad (1.3)$$

Phổ không gian của thuật toán là công suất lối ra ứng với góc quét. Các góc ứng với đỉnh trong phổ không gian chính là góc sóng đến cần ước lượng. Tới đây, việc chọn vector trọng số cho bộ tạo búp sóng được thực hiện theo nhiều phương pháp, mà dưới đây là hai phương pháp điển hình và thường xuyên được sử dụng nhất là Delay and Sum (hay bộ tạo búp sóng truyền thống) và Capon (hay bộ tạo búp sóng MVDR).

* Delay and Sum (Conventional Beamforming Method)

Phương pháp tạo búp sóng truyền thống hay Delay-and-Sum [10] là phương pháp

đơn giản nhất trong các kỹ thuật tìm hướng sóng đến bằng việc tạo búp sóng. Ý tưởng của phương pháp là quét qua các vùng góc quan tâm (thường là trong các phần rìa rạc) và hướng nào tạo ra công suất đầu ra lớn nhất là được coi là hướng tín hiệu mong muốn cần ước lượng.

Xét một tín hiệu $\mathbf{x}(k)$ có hướng đến so với mảng anten thu là ϕ_0 , theo mô hình băng hẹp hình 1.2 với $\mathbf{x}(t) = \sum_{k=1}^K \mathbf{a}_k(t)\mathbf{s}_k(t) + \mathbf{n}(t)$ (được diễn giải chi tiết ở công thức (1.13)), công suất đầu ra của bộ tạo búp sóng được tính bởi:

$$\begin{aligned}\mathbf{P}_{CBF}(\phi_0) &= E[|\mathbf{w}^H \mathbf{x}(k)|^2] = E[|\mathbf{w}^H (\mathbf{a}(\phi_0)\mathbf{s}(k) + \mathbf{n}(k))|^2] \\ &= [|\mathbf{w}^H \mathbf{a}(\phi_0)|^2(\sigma_s^2 + \sigma_n^2)]\end{aligned}\quad (1.4)$$

với $\mathbf{a}(\phi_0)$ là vector đáp ứng mảng với sóng đến góc ϕ_0 , $\mathbf{n}(k)$ là vector tạp âm ở mảng đầu vào, $\sigma_s = E[\mathbf{s}(k)^2]$ và $\sigma_n = E[\mathbf{n}(k)^2]$ lần lượt là công suất của tín hiệu và tạp âm. Theo công thức (1.4), công suất đầu ra tối đa khi $\mathbf{w} = \mathbf{a}(\phi_0)$, hay trong tất cả các vector trọng số có thể, anten thu có mức tăng cao nhất theo hướng ϕ_0 khi $\mathbf{w} = \mathbf{a}(\phi_0)$. Điều này có được do $\mathbf{w} = \mathbf{a}(\phi_0)$ căn chỉnh góc của các thành phần tín hiệu đến từ góc ϕ_0 tại mảng anten thu, vậy nên chúng đẩy giá trị công suất lên.

Thay $\mathbf{w} = \mathbf{a}(\phi_0)$ vào công thức của công suất lỗi ra ứng với góc quét (1.3), phổ không gian vẽ bởi công suất lỗi ra từ bộ tạo búp sóng truyền thông tương ứng:

$$\mathbf{P}_{CBF}(\phi) = \mathbf{w}^H \tilde{\mathbf{R}}_{\mathbf{x}} \mathbf{w} = \mathbf{a}^H(\phi) \tilde{\mathbf{R}}_{\mathbf{x}} \mathbf{a}(\phi) \quad (1.5)$$

Vì là phương pháp đơn giản nhất nên phương pháp tạo búp sóng truyền thông có nhiều nhược điểm: độ rộng của búp sóng lớn dẫn đến khi có nhiều nguồn đến với góc gần nhau, công suất của chúng sẽ bị cộng lại thành một búp sóng duy nhất. Mặc dù có thể tăng độ phân giải bằng cách tăng số lượng máy thu nhưng cách tốt hơn là sử dụng một thuật toán tạo búp sóng khác, đó là Capon được trình bày dưới đây.

* Thuật toán tạo búp sóng của Capon

Bộ tạo búp sóng Capon [9] hay Minimum Variance Distortionless Response cũng là một phương pháp phổ biến để khắc phục hạn chế về độ phân giải của phương pháp CBF. Phương pháp này thực hiện tối thiểu hóa công suất tạp âm và công suất theo các hướng khác, ngoại trừ hướng của tín hiệu đến ϕ_0 vì vậy vector trọng số \mathbf{w} trong phương trình (1.6) còn được gọi là vector trọng số của búp sóng đáp ứng không méo phương sai

tối thiểu (MVDR). Phương pháp này có phương trình tối ưu:

$$\min_{\mathbf{w}} E[|\mathbf{y}(k)|^2] = \min_{\mathbf{w}} \mathbf{w}^H \mathbf{R}_x \mathbf{w} \text{ với } \mathbf{w}^H \mathbf{a}(\phi_0) = 1 \quad (1.6)$$

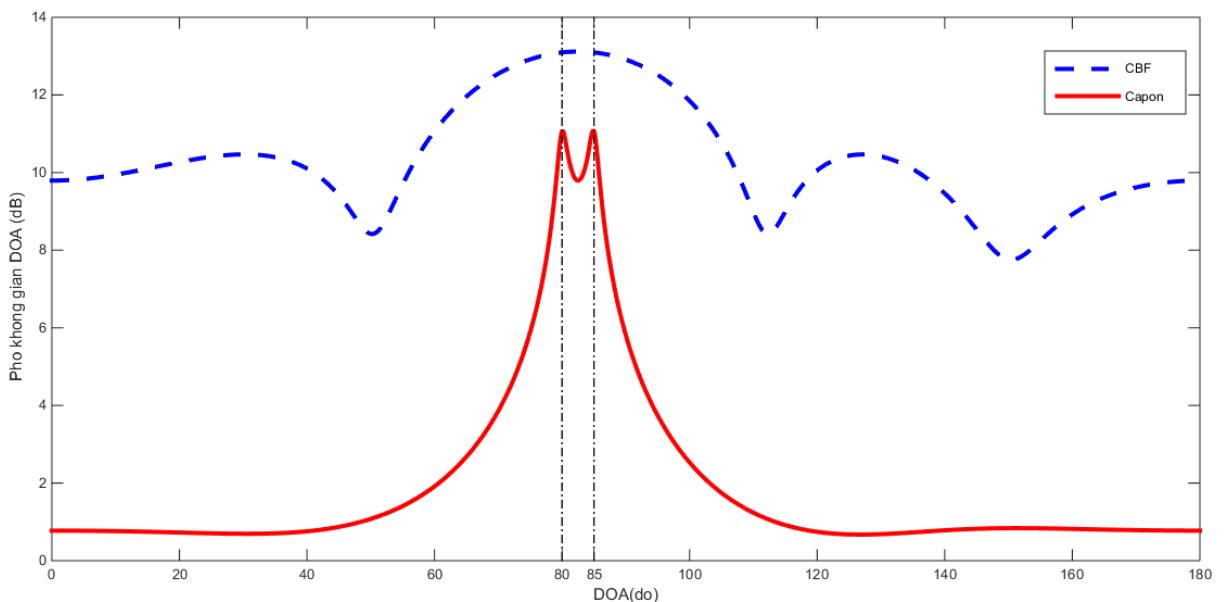
Sử dụng phương pháp nhân tử Lagrange, giải được nghiệm của phương trình (1.6) là vector trọng số:

$$\mathbf{w} = \frac{\mathbf{R}_x^{-1} \mathbf{a}(\phi)}{\mathbf{a}^H(\phi) \mathbf{R}_x^{-1} \mathbf{a}(\phi)} \quad (1.7)$$

Với nghiệm trên, thay vào (1.3), phổ không gian vẽ bởi công suất lối ra từ bộ tạo búp sóng Capon, với các đỉnh của phổ không gian chính là ước lượng hướng sóng đến của tín hiệu tương ứng:

$$\mathbf{P}_{Capon}(\phi) = \frac{1}{\mathbf{a}^H(\phi) \mathbf{R}_x \mathbf{a}(\phi)} \quad (1.8)$$

Hình 1.3 là kết quả mô phỏng 2 thuật toán tìm hướng sóng đến CBF và Capon trên Matlab với 2 nguồn phát ở góc 80° và 85° , mảng anten thu gồm 4 phần tử anten đặt thẳng cách đều nhau $0,5\lambda = 0,5 \times 0,325(m)$. Có thể thấy phương pháp CBF chỉ nhận diện được một búp sóng ở khoảng góc từ 60° đến 100° còn Capon đã khắc phục được độ phân giải kém này khi xác định được 2 đỉnh ở 80° và 85° . Tuy nhiên, dễ nhận thấy khi góc đến thu hẹp hơn nữa chỉ 1° hay 2° thì phương pháp Capon cũng khó có thể phân biệt được. Hơn nữa, phương pháp Capon cần có điều kiện là không có các tín hiệu khác tương quan với tín hiệu cần xác định hướng sóng đến và trong thuật toán có phương trình yêu cầu nghịch đảo ma trận, nếu tăng số phần tử thu lớn có thể dẫn đến tốn kém bộ nhớ.



Hình 1.3. Mô phỏng thuật toán Delay and Sum và Capon

1.2.2. Thuật toán Maximum Likelihood

Maximum Likelihood (ML) là một trong những thuật toán đầu tiên được phát triển để tính toán DOA. Tuy không phổ biến bằng thuật toán MUSIC trình bày trong mục 1.3 bên dưới, tuy nhiên do sử dụng phương pháp tham số nên tránh được các vấn đề nhiễu đa đường, tín hiệu tương quan cao, số lượng mẫu nhỏ của các phương pháp dùng phổ [8].

Với nguyên tắc cơ bản là tìm mô hình dữ liệu giống nhất với dữ liệu được thu thập tại mảng anten thu. Vẫn tiếp tục sử dụng mô hình tín hiệu nguồn như phương trình (1.13), hàm mật độ xác suất chung của K mẫu thu thập $\mathbf{x}(1), \dots, \mathbf{x}(K)$ là:

$$f(\mathbf{x}) = \prod_{k=1}^K \frac{1}{\pi \det[\sigma^2 \mathbf{I}]} \cdot \exp\left(-\frac{1}{\sigma^2} |\mathbf{x}(k) - A(\Phi)\mathbf{s}(k)|^2\right) \quad (1.9)$$

với \det là định thức của ma trận. Tiếp theo, loại bỏ các thành phần không phụ thuộc vào Φ , ta có hàm log likelihood:

$$L(\Phi) = -Mp \log \sigma^2 - \frac{1}{\sigma^2} \sum_{k=1}^K |\mathbf{x}(k) - A(\Phi)\mathbf{s}(k)|^2 \quad (1.10)$$

Phương pháp ML thực hiện tìm giá trị cực đại của hàm log likelihood khi thay đổi các giá trị Φ và \mathbf{s} , bỏ qua các giá trị không thay đổi, ta thu được:

$$\max_{\Phi, \mathbf{s}} \left[-Mp \log\left(\frac{1}{Mp} \sum_{k=1}^K |\mathbf{x}(k) - A(\Phi)\mathbf{s}(k)|^2\right) \right] \quad (1.11)$$

Do log là một hàm đơn điệu nên tìm cực đại của phương trình (1.11) tương đương với:

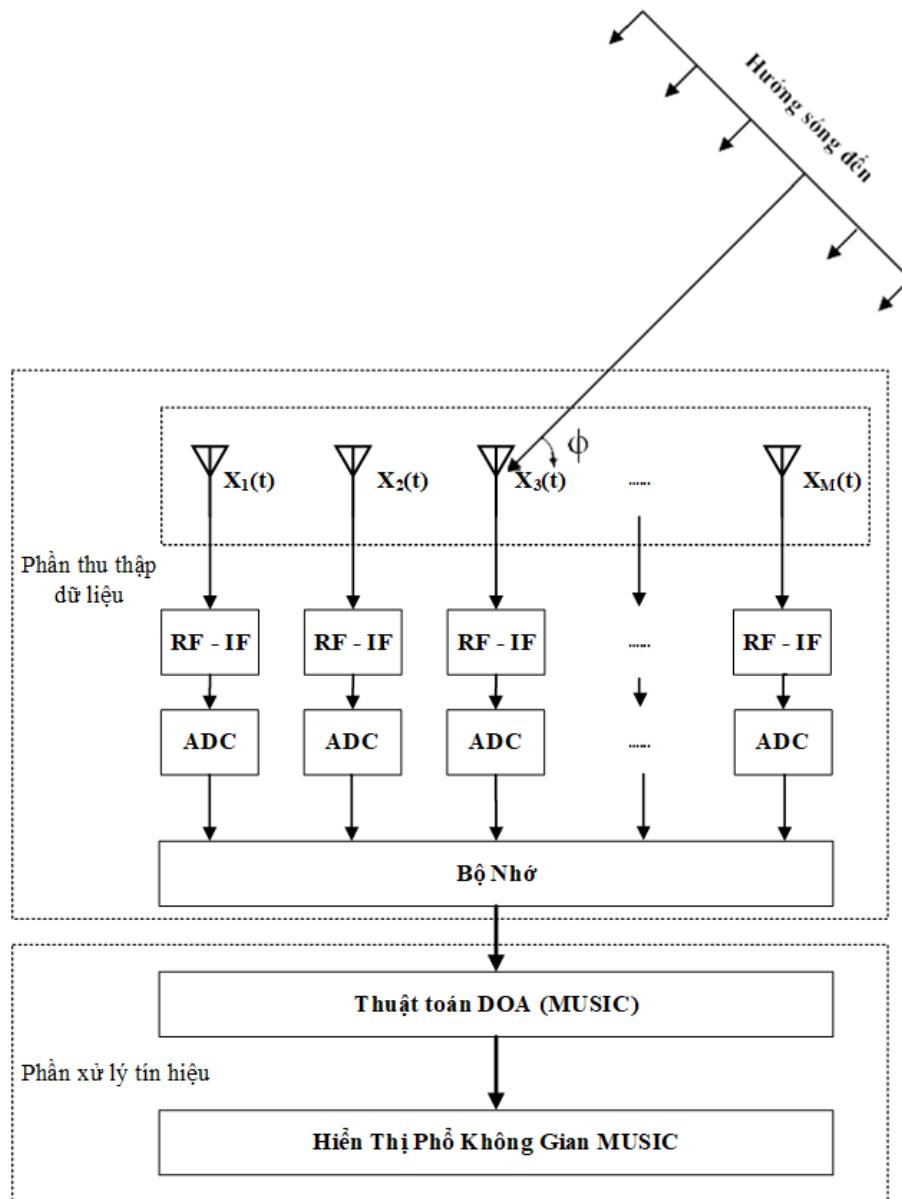
$$\min_{\Phi, \mathbf{s}} \left[\sum_{k=1}^K |\mathbf{x}(k) - A(\Phi)\mathbf{s}(k)|^2 \right] \quad (1.12)$$

Các bước tiếp thuộc vấn đề tối ưu hóa phi tuyến tính nhiều chiều và phức tạp, cụ thể được trình bày ở [8].

1.3. Hệ thống xác định hướng sóng đến với thuật toán MUSIC

Phương pháp phương sai tối thiểu của Capon thường chính xác và được sử dụng rộng rãi, tuy nhiên phương pháp này có một số hạn chế cơ bản trong độ phân giải. Còn phương pháp tham số: ML lại có độ phức tạp cao khi thực hiện tìm kiếm D-D (D chiều). Schmidt đã đưa ra một giải pháp hoàn chỉnh dựa trên thống kê bậc hai cho vấn đề ước lượng DOA, có khả năng áp dụng cho mọi loại tín hiệu với cấu trúc anten tùy ý. Kỹ thuật được đề xuất bởi Schmidt vào năm 1979 với tên gọi: thuật toán phân loại đa tín hiệu

(MUSIC) [23]. Sơ đồ khối của hệ tìm hướng sóng đến sử dụng thuật toán MUSIC được đưa ra trên hình 1.4.



Hình 1.4. Sơ đồ khối hệ tìm hướng sóng đến sử dụng thuật toán MUSIC

Xét D nguồn tín hiệu đến với các hướng không biết trước ϕ_1, \dots, ϕ_D đến mảng anten thu gồm M (điều kiện $M > D$) phần tử vô hướng được đặt tùy ý trong mặt phẳng ở các tọa độ $(\bar{x}_1, \bar{y}_1), \dots, (\bar{x}_M, \bar{y}_M)$.

Vector tín hiệu bên thu nhận được $\mathbf{x}(t) \in \mathbb{C}^M$ tại thời điểm t được biểu diễn bởi công thức (1.13), với $\Phi = [\phi_1, \dots, \phi_D]^T$ tương ứng với hướng của các nguồn tín hiệu đến mảng anten thu, $\mathbf{A}(\Phi) = [\mathbf{a}(\phi_1), \dots, \mathbf{a}(\phi_D)] \in \mathbb{C}^{M \times D}$ là mảng chứa các vector đáp ứng mảng $\mathbf{a}(\phi) \in \mathbb{C}^M$; $\mathbf{s}(t) \in \mathbb{C}^D$ là vector nguồn tín hiệu và $\mathbf{n}(t) \in \mathbb{C}^M$ tương ứng là vector

tạp âm:

$$\begin{bmatrix} x_1(t) \\ \dots \\ x_M(t) \end{bmatrix} = \begin{bmatrix} \mathbf{a}(\phi_1) & \dots & \mathbf{a}(\phi_D) \end{bmatrix} \begin{bmatrix} s_1(t) \\ \dots \\ s_D(t) \end{bmatrix} + \begin{bmatrix} n_1(t) \\ \dots \\ n_M(t) \end{bmatrix} \quad (1.13)$$

hay được viết dưới dạng ma trận

$$\mathbf{x}(t) = \mathbf{A}(\Phi)\mathbf{s}(t) + \mathbf{n}(t)$$

Trong đó, vector đáp ứng mảng được biểu diễn chi tiết như công thức (1.14) với λ là bước sóng của tín hiệu:

$$\mathbf{a}(\phi) = \begin{bmatrix} e^{-j\frac{2\pi}{\lambda}(\bar{x}_1 \cos \phi + \bar{y}_1 \sin \phi)} \\ \dots \\ e^{-j\frac{2\pi}{\lambda}(\bar{x}_M \cos \phi + \bar{y}_M \sin \phi)} \end{bmatrix} \quad (1.14)$$

Các vector đáp ứng mảng này là các giá trị cần được giả thiết là biết trước tùy theo cấu hình anten, được lưu trữ làm các giá trị chuẩn hóa cần thiết cho mọi bước tính toán; vector tín hiệu và tạp âm (nhiều trăng) có trung bình bằng 0; vector tạp âm trăng hóa không gian (tạp âm đến mỗi phần tử anten đều có phương sai giống nhau là σ_n^2 không tương quan với nhau) và độc lập thống kê với các tín hiệu nguồn.

Ma trận hiệp phương sai trong không gian được biểu diễn bởi công thức (1.15), với \mathcal{E} là ký hiệu của kỳ vọng thống kê, $\mathcal{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\} = \mathbf{R}_s$, $\mathcal{E}\{\mathbf{n}(t)\mathbf{n}^H(t)\} = \sigma_n^2 \mathbf{I}_n$ lần lượt là ma trận tương quan của tín hiệu nguồn và của tạp âm, $\mathbf{I}_n \in \mathbb{C}^{M \times M}$ là ma trận đơn vị.

$$\begin{aligned} \mathbf{C}_x &= \mathbf{R}_x = \mathcal{E}\{\mathbf{x}(t)\mathbf{x}^H(t)\} \\ &= \mathbf{A}\mathcal{E}\{\mathbf{s}(t)\mathbf{s}^H(t)\}\mathbf{A}^H + \mathcal{E}\{\mathbf{n}(t)\mathbf{n}^H(t)\} \\ &= \mathbf{A}\mathbf{R}_s\mathbf{A}^H + \sigma_n^2 \mathbf{I}_n \end{aligned} \quad (1.15)$$

Tiếp tục khai triển ma trận \mathbf{R}_x thành các giá trị riêng và các vector riêng tương ứng là λ_m và $\mathbf{e}_m (m = 1, \dots, M)$.

$$\begin{aligned} \mathbf{R}_x &= \sum_{m=1}^M \lambda_m \mathbf{e}_m \mathbf{e}_m^H = \mathbf{E} \Lambda \mathbf{E}^H = \mathbf{E}_s \Lambda_s \mathbf{E}_s^H + \mathbf{E}_n \Lambda_n \mathbf{E}_n^H \\ &= \mathbf{E}_s \Lambda_s \mathbf{E}_s^H + \sigma_n^2 \mathbf{E}_n \mathbf{E}_n^H = \mathbf{E}_s \tilde{\Lambda}_s \mathbf{E}_s^H + \sigma_n^2 \mathbf{I}_n \end{aligned} \quad (1.16)$$

với $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_M]$

$$\mathbf{E}_s = [\mathbf{e}_1, \dots, \mathbf{e}_D]$$

$$\mathbf{E}_n = [\mathbf{e}_{D+1}, \dots, \mathbf{e}_M]$$

$$\Lambda = diag\{\lambda_1, \dots, \lambda_M\}$$

$$\Lambda_s = diag\{\lambda_1, \dots, \lambda_D\}$$

$$\Lambda_n = diag\{\lambda_D, \dots, \lambda_M\}$$

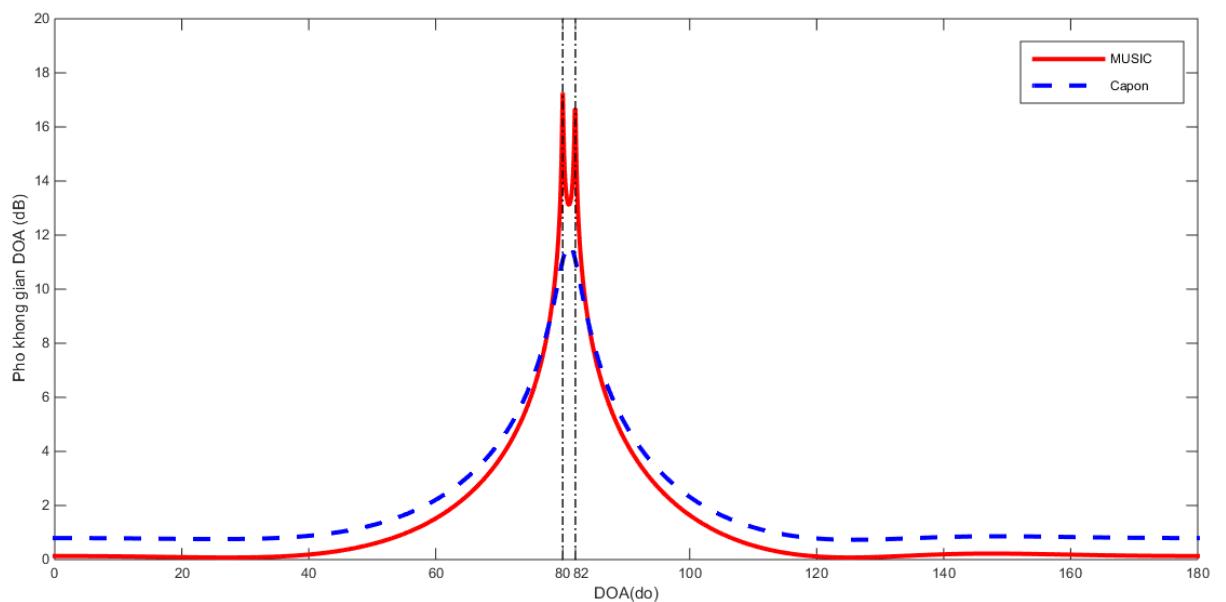
$$\tilde{\Lambda}_s = \Lambda_s - \sigma_n^2 \mathbf{I}_n$$

Vector riêng $\mathbf{E} = [\mathbf{E}_s, \mathbf{E}_n]$ có thể giả sử để tạo thành một cơ sở trực giao ($\mathbf{E}\mathbf{E}^H = \mathbf{E}^H\mathbf{E} = \mathbf{I}$). Từ \mathbf{E} có thể tách được ra hai ma trận: ma trận chứa D vector riêng tương ứng với không gian con tín hiệu là $\mathbf{E}_s \triangleq [\mathbf{e}_1, \dots, \mathbf{e}_D]$; ma trận chứa $M - D$ vector riêng tương ứng với không gian con của tạp âm là $\mathbf{E}_n \triangleq [\mathbf{e}_{D+1}, \dots, \mathbf{e}_M]$. Để phân tích chi tiết các thuộc tính cấu trúc riêng của ma trận hiệp phương sai \mathbf{R}_x tham khảo tại [23].

Khi các không gian con đã được xác định, có thể ước tính hướng đến của các tín hiệu mong muốn bằng cách tính phổ không gian MUSIC trên vùng quan tâm:

$$\mathbf{P}_{MUSIC}(\phi) = \frac{\mathbf{a}^H(\phi)\mathbf{a}(\phi)}{\mathbf{a}^H(\phi)\mathbf{E}_n\mathbf{E}_n^H\mathbf{a}(\phi)} \quad (1.17)$$

Thực hiện mô phỏng với thông số mảng thu tương tự như thuật toán Capon trên Matlab, tuy nhiên thay đổi góc của 2 nguồn tín hiệu phát là 80° và 82° , hình 1.5 dưới đây biểu diễn kết quả thu được. Có thể thấy rõ sự vượt trội của thuật toán MUSIC trong việc ước tính DOA.



Hình 1.5. Mô phỏng thuật toán MUSIC và Capon

CHƯƠNG 2

TRIỂN KHAI THUẬT TOÁN MUSIC TRÊN THIẾT BỊ SDR

Chương này sẽ giới thiệu khái quát về lịch sử, cách thức hoạt động của phần mềm GNU Radio và thiết bị vô tuyến định nghĩa bằng phần mềm SDR. Tập trung tìm hiểu về phần cứng của BladeRF và phần mềm trên GNU Radio để thực thi trên BladeRF, cùng với đó là việc đồng bộ hệ nhiều BladeRF trước khi triển khai thuật toán MUSIC trên BladeRF.

2.1. Phần mềm GNU Radio và thiết bị vô tuyến định nghĩa bằng phần mềm SDR

2.1.1. Phần mềm GNU Radio

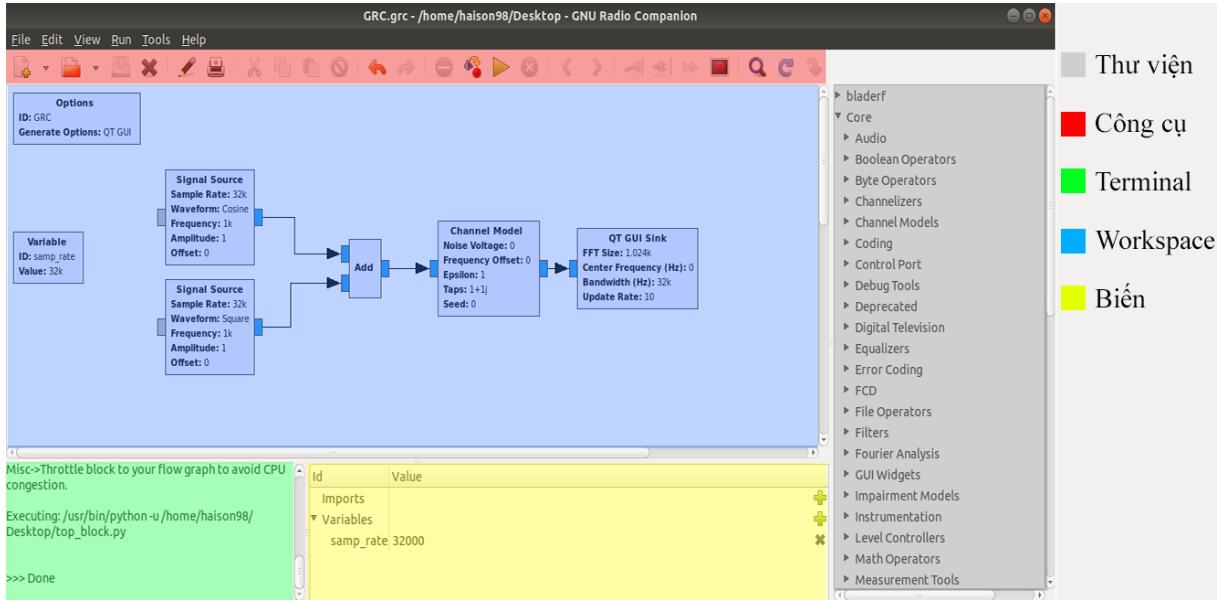
GNU Radio là bộ công cụ phát triển phần mềm, mã nguồn mở và hoàn toàn miễn phí, cung cấp các khối xử lý tín hiệu để mô phỏng và thực hiện các hệ truyền thông. Phần mềm tương thích với các thiết bị ngoại vi vô tuyến để tạo thành thiết bị vô tuyến được định nghĩa bằng phần mềm (Software Define Radio). GNU Radio sử dụng rộng rãi trong giảng dạy, nghiên cứu cũng như thương mại, hỗ trợ cho việc nghiên cứu các hệ truyền thông không dây và các hệ truyền thông vô tuyến trong thực tế.

Được khởi xướng bởi nhà hảo tâm John Gilmore và trưởng dự án Eric Blossom từ năm 2001 liên tục được phát triển, cho đến năm 2009, Josh Blum phát hành phiên bản GNU Radio Companion 3.2 đầu tiên. Năm 2011, đội ngũ các nhà phát triển lõi của GNU Radio bắt đầu tổ chức hội nghị hàng năm đầu tiên với tên gọi GRCon. Hiện tại, GNU Radio đã phát hành đến phiên bản 3.8 với nhiều cải tiến và liên tục được cập nhật bởi nhà phát hành cùng với các nhà phát triển độc lập [19]. GNU Radio có sẵn rất nhiều khối được hoàn thiện và sẵn sàng sử dụng, dưới đây là những khối điển hình thường xuyên sử dụng trong hệ truyền thông không dây 2G, 3G, 4G và 5G:

- Waveform Generators
 - ◊ Signal Source (e.g. Sine, Square, Saw Tooth,...)
 - ◊ Noise Source
 - ◊ Constant Source
- Modulator

- ◊ AM Demod
- ◊ Continuous Phase Modulation
- ◊ PSK Mod / Demod
- ◊ GFSK Mod / Demod
- ◊ QAM Mod / Demod
- ◊ WBFM / NBFM Receive
- ◊ DVB-T / DVB-T2
- ◊ CDMA [16]
- ◊ OFDM [3]
- Channel Models
 - ◊ Channel Model
 - ◊ Fading Model
 - ◊ Frequency Selective Fading Model
- Filters
 - ◊ Low / High Pass Filter
 - ◊ Band Pass / Reject Filter
 - ◊ IIR Filter
 - ◊ Decimating FIR Filter
 - ◊ Hilbert
 - ◊ Root Raised Cosine Filter
 - ◊ FFT Filter
- Fourier Analysis
 - ◊ FFT
 - ◊ FLL Band-Edge
 - ◊ Rational Resampler (Synchronizers)
 - ◊ PN Correlator

Giao diện cho người dùng thông thường là GNU Radio Companion (GRC) được hiển thị như hình 2.1. Với giao diện khá tương đồng với Matlab Simulink, gồm các khối nguồn, xử lý tín hiệu, hiển thị, widget,... dễ dàng chọn lựa, tùy chỉnh và kết nối với nhau thành flow-graph. Tất cả được GRC biên dịch ra file python và đẩy cho GNU Radio xử lý và hiển thị kết quả dựa trên QT GUI (QT Toolkit), hoặc Wx GUI (WxPython, tuy nhiên Wx Python đã bị xóa bỏ ở phiên bản GNU 3.8 do tốc độ đáp ứng chậm).



Hình 2.1. Giao diện GNU Radio Companion

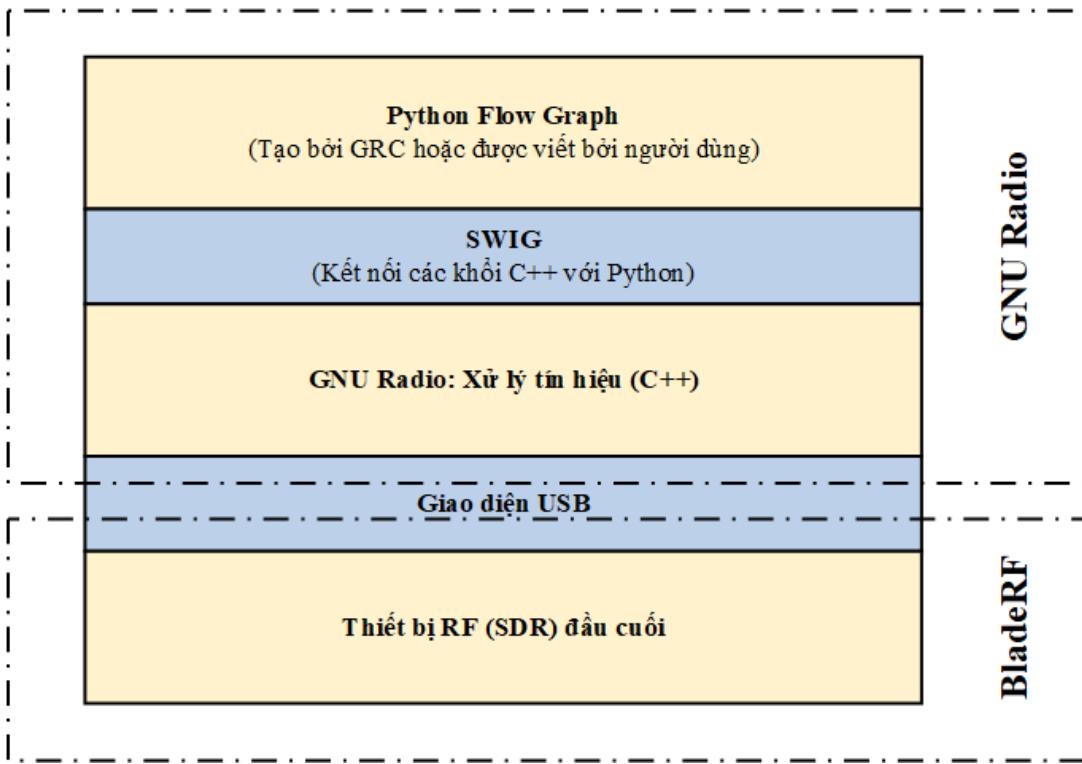
GNU Radio được viết hoàn toàn bằng ngôn ngữ C++ để đảm bảo hiệu năng, tuy nhiên người dùng vẫn có thể sử dụng Python để viết các khối do GNU Radio sử dụng SWIG để liên kết C++ với các ngôn ngữ bậc cao mà ở đây là Python. Hình 2.2 mô tả cách thức vận hành của GNU Radio và cách nó liên kết với thiết bị SDR.

Các luồng dữ liệu trong GNU Radio đi qua các khối thực thi được phân chia rõ ràng về kiểu dữ liệu như trên hình 2.3(a) và độ dài của vector dữ liệu như hình 2.3(b), chỉ khi chúng khớp nhau, GRC thông báo hợp lệ thì chương trình mới có thể chạy. Điều này có được là do ngay trong việc khởi tạo khối, các thông số trên đã được định sẵn để đảm bảo việc xử lý tín hiệu là đúng với mục đích ban đầu của khối.

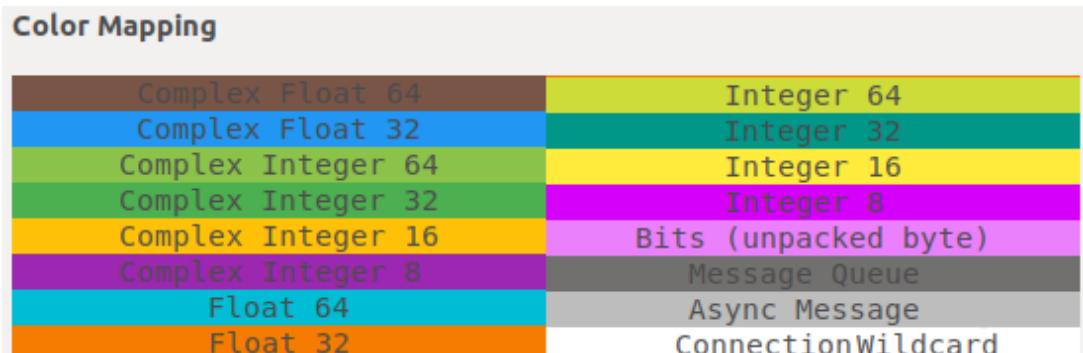
Đối với các nhà phát triển muốn tạo thêm các khối thực hiện các chức năng đặc biệt, GNU Radio cung cấp khả năng tương thích với cả C++ và Python. Bảng 2.1 dưới đây so sánh tính năng của các phương pháp lập trình khối trong GNU Radio.

2.1.2. Thiết bị vô tuyến định nghĩa bằng phần mềm SDR

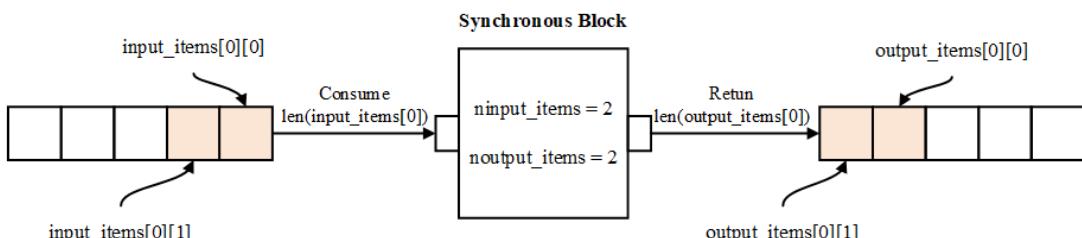
Vô tuyến định nghĩa bằng phần mềm (Software Define Radio) là một hệ thống truyền thông vô tuyến được thực hiện chủ yếu bằng phần mềm trên máy tính cá nhân



Hình 2.2. Sơ đồ làm việc của GNU Radio



(a) Mã màu của các kiểu dữ liệu



(b) Luồng dữ liệu qua các khối xử lý

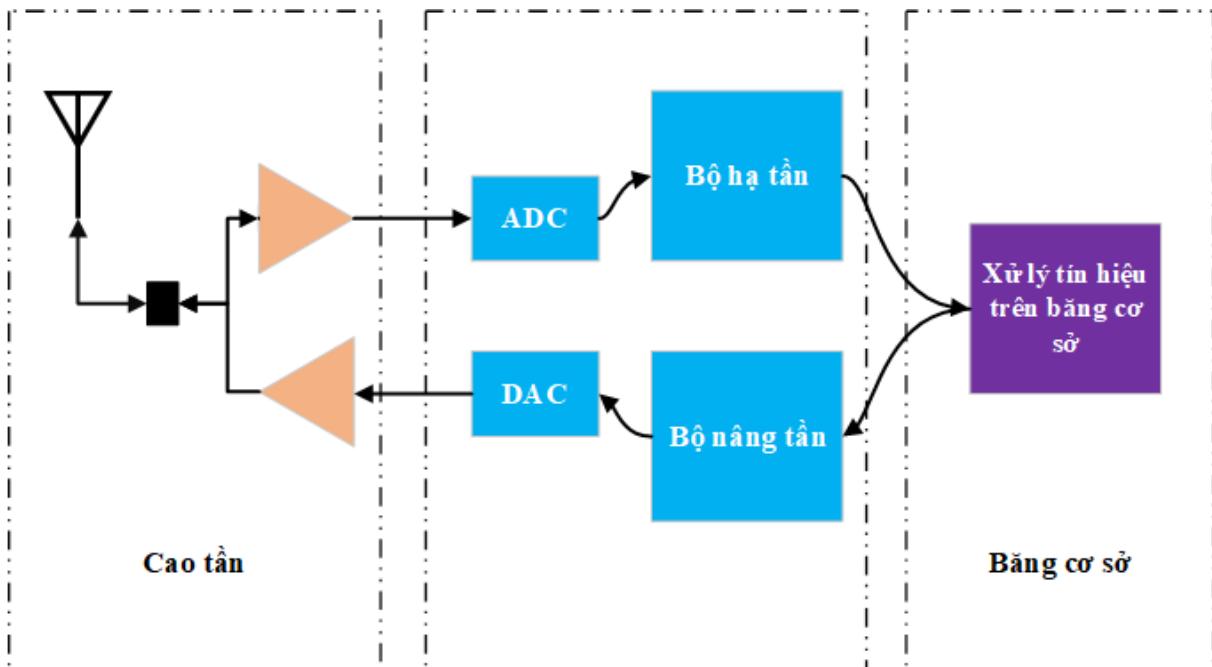
Hình 2.3. Kiểu dữ liệu và xử lý tín hiệu trong GNU Radio

hoặc hệ thống nhúng. Một SDR cơ bản bao gồm máy tính có sound card hoặc thiết bị chuyển đổi tương tự sang số (ADC) cùng thiết bị chuyển đổi số sang tương tự (DAC) và đầu cuối cao tần để thu phát tín hiệu, mô tả khái quát ở hình 2.4. Phần xử lý tín hiệu hầu hết được thực hiện bởi bộ xử lý đa năng (GPP) thay vì phần cứng chuyên dụng (mạch

Bảng 2.1. Các phương pháp sử dụng GNU Radio

GNU Radio Companion	C++	Python
Dễ dàng và trực quan.	Đầy đủ các tùy chỉnh.	Đầy đủ các tùy chỉnh.
Tạo flow-graph bằng cách kết nối và tinh chỉnh các khối có sẵn trong GRC.	Tốc độ thực thi nhanh.	Lập trình nhanh chóng (Numpy, Scipy,...)
	Thực hiện tốt các hệ truyền thông thời gian thực.	Tốt trong việc thực hiện các thực nghiệm nhanh chóng.
	Độ phức tạp và thời gian lập trình lâu.	
	Kết nối với Python bằng SWIG.	

diện tử). Với thiết kế này một thiết bị SDR có thể truyền nhận được tín hiệu ở nhiều tần số, loại điều chế khác nhau chỉ dựa trên phần mềm sử dụng.

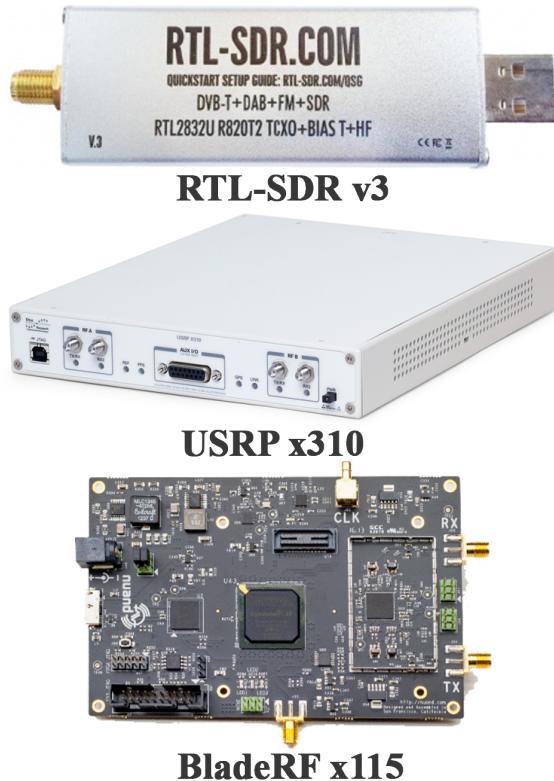


Hình 2.4. Sơ đồ khái niệm của một hệ thống SDR

Software Define Radio được sử dụng trong các ứng dụng quân sự [5] và trong viễn thông di động [7, 11], cả hai đều có thể sử dụng nhiều giao thức vô tuyến và đáp ứng thời gian thực. Về lâu dài, SDR được mong đợi sẽ trở thành công nghệ thống trị trong truyền thông vô tuyến bằng cách kết hợp với Anten định nghĩa bằng phần mềm (Software Define Antennas) tạo thành vô tuyến nhận thức (Cognitive Radio).

Với tên ban đầu máy thu kỹ thuật số (Digital Receiver) được tạo ra vào năm 1970

bởi một nhà nghiên cứu thuộc bộ quốc phòng Hoa Kỳ, từ năm 1990 đến 1995, ứng dụng trong dự án SpeakEasy [21], một chương trình cho các đài phát thanh điều khiển không quân mặt đất chiến thuật của không quân Hoa Kỳ có thể hoạt động từ 2 MHz đến 2 GHz. Thuật ngữ Software Define Radio do Stephen Blust đặt ra và đề xuất tại Modular Multifunction Information Transfer Systems năm 1996. Tiếp tục được nghiên cứu và phát triển liên tục sau đó, đến nay đã có nhiều sản phẩm thương mại SDR đã bán ra trên toàn cầu cho mục đích quân sự, nghiên cứu, giảng dạy hay ứng dụng vào sản phẩm truyền thông không dây thực tế. Có thể kể đến trong quân sự, dự án Joint Tactical Radio System (JTRS) [2] sản xuất bộ đàm cung cấp liên lạc linh hoạt và có thể tương tác, nhưng do vấn đề chi phí cao hơn so với việc sử dụng các linh kiện chuyên dụng cho một mục đích nên chưa được sử dụng rộng rãi. Tuy nhiên, tính linh hoạt của SDR có thể là chìa khóa cho truyền thông vô tuyến trong tương lai, một khi chi phí cố định để sản xuất một SDR đã giảm đủ để vượt qua chi phí thiết kế lại các hệ thống sử dụng linh kiện điện tử cho một mục đích.



Hình 2.5. Các thiết bị SDR thông dụng

Các ứng dụng trong giảng dạy và nghiên cứu với SDR thông dụng hơn, với rất nhiều thiết bị SDR ở nhiều mức giá và mục đích khác nhau (hình 2.5). Từ các thiết bị giá rẻ chỉ 22 \$ như DVB-T USB dongles sử dụng IC điều khiển Realtek RTL2832U, và Rafael Micro R820T làm bộ chỉnh tần số, thiết bị chỉ thực hiện chức năng thu tín hiệu, có thể hoạt động trong dải từ 500 kHz đến 1,75 GHz để thu FM, DVB-T, GSM,... Với các

nghiên cứu chuyên sâu hơn Universal Software Radio Peripheral (USRP) là lựa chọn tốt với dải thu siêu cao tần lên đến 6 GHz băng thông rộng 200 MHz cùng khả năng MIMO được tích hợp sẵn.

2.2. Phần cứng và phần mềm của BladeRF

2.2.1. BladeRF x115

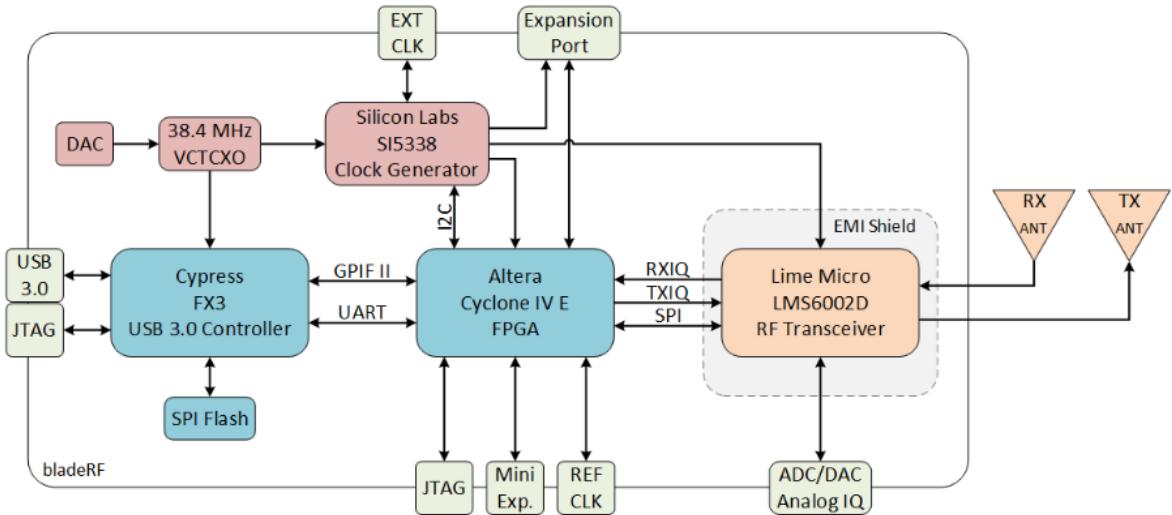
Trong nghiên cứu này sử dụng các Kit SDR là BladeRF x115 của Nuand, ra mắt năm 2013, được tích hợp nhiều công nghệ mới: hỗ trợ truyền nhận đồng thời (full duplex), bộ ADC, DAC tốc độ cao, khả năng lập trình FPGA, tương thích giao diện USB 3.0, cổng kết nối mở rộng (GPIO), MIMO,... Các thông số chi tiết ở bảng 2.2.

Bảng 2.2. Thông số kỹ thuật của BladeRF x115

Thông Số	Nhỏ nhất	Thông thường	Lớn nhất	Đơn vị
Thông số kỹ thuật RF				
ADC/DAC Sample Rate	0,16		40	MHz
ADC/DAC Resolution		12		bits
VCTCXO Accuracy		1		ppm
RF Tuning Range	300		3 800	MHz
RF Bandwidth Filter	1,5		28	MHz
CW Output		+6		dBm
Thông số kỹ thuật FPGA				
Logic Elements			114 480	LE
Embedded 18x18 Multipliers			266	
BRAM			3 888	kbits
Thông số vật lý				
Kích thước		8,7 x 13,1 x 1,8		cm
Cân nặng		80		g
Nhiệt độ làm việc	0		70	°C

Cấu hình bên trong của BladeRFx115 biểu diễn trên hình 2.6, các thành phần chính bao gồm:

- Altera Cyclone IV EP4CE15 FPGA: thành phần xử lý trung tâm của BladeRF, có thể lập trình lại qua cổng JTAG.
- Lime Micro LMS6002D RF Transceiver: IC thu phát cao tần có thể lập trình, hoạt động trong dải 300 MHz đến 3,8 GHz, trên cả băng tần được cấp phép và không được cấp phép và với bất cứ tiêu chuẩn truyền thông không dây nào. Các bộ ADC, DAC 12 bits truyền trực tiếp mẫu đến FPGA. Tích hợp sẵn LNA, PA, bộ trộn TX/RX, bộ lọc TX/RX, ...



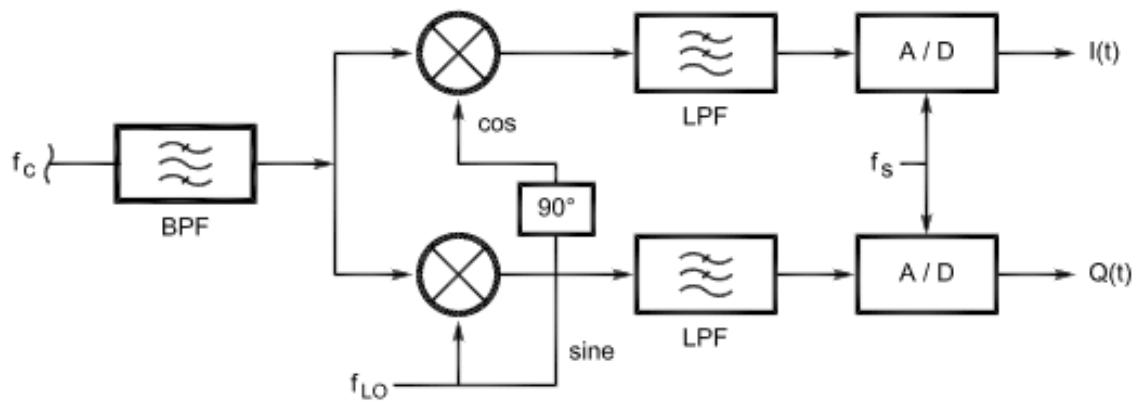
Hình 2.6. Cấu hình của BladeRF x115

- Cypress FX3 USB 3.0 Controller: bộ điều khiển ngoại vi USB 3.0 tốc độ cao băng thông 5 Gbps, có cấu hình qua GPIF™II. Là cầu nối giao tiếp giữa BladeRF và máy tính hay các thiết bị ngoại vi khác như SPI Flash.
- SPI Flash: IC MX25U3235EM2I-10G cung cấp 32 Mb bộ nhớ Flash.
- Silicon Labs Si5338 Clock Generator: bộ tạo xung đồng hồ chuẩn cho toàn BladeRF, giao tiếp với FPGA qua chuẩn I2C.
- VCTCXO: bộ tạo dao động tinh thể để bù cho nhiệt độ, được hiệu chuẩn ở mức sai số dưới 1 Hz trong 38,4 MHz.
- Expansion port, EXT CLK, Mini Exp., REF CLK, JTAG, ADC/DAC Analog IQ: các cổng mở rộng để phục vụ từng mục đích chuyên biệt cho BladeRF.

2.2.2. Phần mềm của BladeRF

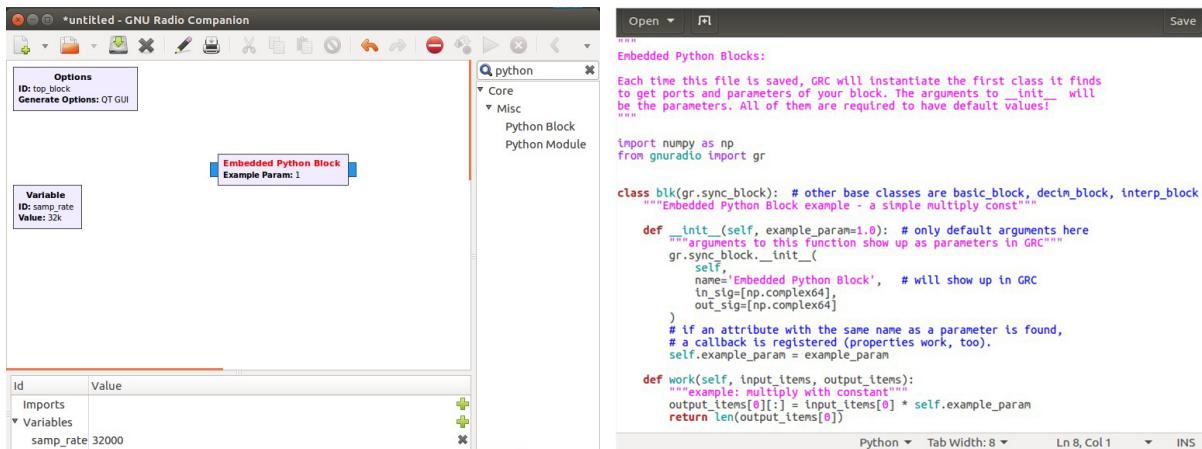
Dữ liệu đầu ra từ BladeRF truyền về máy tính là các mẫu IQ được lấy mẫu như hình 2.7, lưu ở chuẩn SC16Q11, khi đưa vào các khối trong GNU Radio sẽ chuyển đổi sang Complex Float32.

Để sử dụng dữ liệu đầu ra của BladeRF cho công việc xác định hướng sóng đến, cần phải viết thêm những khối OTT (Out Of Tree Modules) vào GNU Radio để đảm nhiệm việc xử lý tín hiệu. Có nhiều cách để tạo một khối OTT trên GNU Radio, đơn giản nhất là sử dụng ngay Python Block là một khối trong thư viện của GNU Radio, cho phép lập trình một khối mới ngay lập tức với ngôn ngữ Python, hình 2.8 là khái niệm Python Block trên GRC và mã lập trình có sẵn bên trong khái niệm.



Hình 2.7. Lấy mẫu IQ

và không thể tùy biến được nhiều và sâu cho khồi, vì vậy chỉ nên sử dụng khi kiểm tra nhanh thuật toán. Một cách nữa do nhà phát hành cung cấp chính thức cho các nhà phát triển đó là công cụ gr_modtool.

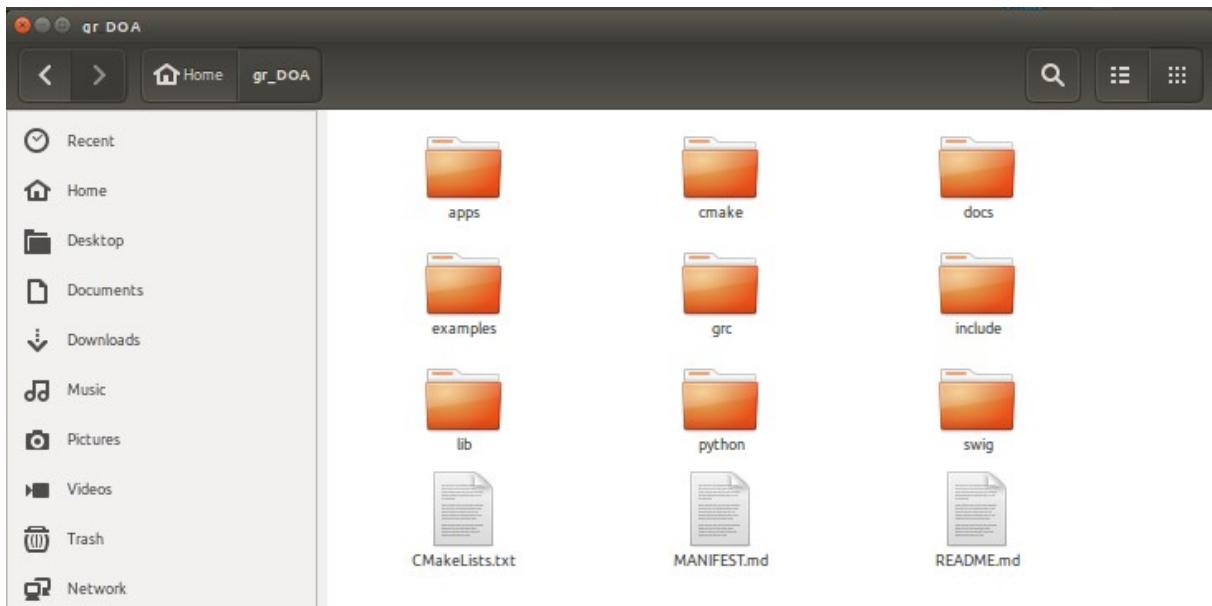


(a) Khối Python Block trên GRC

(b) Lập trình cho khối Python Block

Hình 2.8. Tạo khối với Python Block trên GNU Radio

Công cụ gr_modtool được tích hợp sẵn vào GNU Radio từ năm 2010, sử dụng để tạo sẵn template các khối OTT với chức năng đa dạng cho người dùng, giúp bỏ qua các bước phải lặp đi lặp lại như: tạo các mã soạn sẵn, sửa makefile, ... Hỗ trợ lập trình khối cho GNU Radio với ngôn ngữ C++ và Python với liên kết bằng SWIG tích hợp sẵn, việc cài đặt các khối sau khi lập trình được thực hiện tự động bằng CMAKE, hoàn toàn phù hợp cho cả người mới bắt đầu vào các mục đích nghiên cứu chuyên sâu. Hình 2.9 ví dụ một Module tạo bằng gr_modtool chứa nhiều thư mục và file đã tạo sẵn, một Module có thể chứa nhiều khối OTT.



Hình 2.9. Module tạo bằng gr_modtool

Tiếp theo, cần xác định loại khối cho các mục đích khác nhau, dưới đây là 4 loại khối được GNU Radio cung cấp:

- Synchronous Blocks: Số mẫu đầu vào và đầu ra của khối là bằng nhau (1/1).
- Decimation Blocks: Cứ N mẫu đầu vào thì có 1 mẫu đầu ra (N/1).
- Interpolation Blocks: Cứ 1 mẫu đầu vào thì có M mẫu đầu ra (1/M).
- General Blocks: Tùy chỉnh tỷ lệ số mẫu đầu vào và đầu ra (N/M).

Việc cài đặt khối vào GNU Radio sau khi lập trình xong cũng rất đơn giản do sử dụng CMAKE, sau khi đã cài đặt, các khối OTT sẽ xuất hiện trong hộp thoại của GRC, dễ dàng chọn, tùy chỉnh thông số và kết nối với nhau.

2.3. Triển khai hệ thống xác định hướng sóng đến trên hệ BladeRF

Trên thực tế, việc xây dựng một hệ thống gồm nhiều thiết bị SDR mà ở đây là BladeRF để phục vụ cho việc xác định hướng sóng đến, yêu cầu hệ thu cần phải được đồng bộ hoàn toàn với nhau trước khi sử dụng dữ liệu của SDR để ước lượng DOA.

Trong phần này, để phù hợp với thực tế về số lượng SDR và phần cứng có sẵn, mảng anten thu sẽ chỉ gồm 2 phần tử BladeRF. Các bước đồng bộ dưới đây vẫn áp dụng được cho hệ nhiều phần tử BladeRF, tuy nhiên sẽ cần thêm các phần cứng chuyên dụng: bộ chia bù công suất, bộ chia giữ nguyên độ lệch pha, máy phát xung chuẩn. Chi tiết như sau:

Không giống như dữ liệu lý tưởng khi mô phỏng, dữ liệu thực tế từ hệ SDR sẽ gặp các vấn đề cần phải được xử lý: xung đồng hồ chủ khác nhau; DC_{offset} và cân bằng mẫu IQ cho các tùng thiết bị SDR; việc truyền/nhận dữ liệu từ GNU Radio trên máy tính đến các SDR qua cổng USB thường không được thực hiện đồng thời, dẫn đến hiện tượng các SDR bắt đầu chạy và gửi mẫu về ở các thời gian khác nhau, gây ra trễ mẫu (sample_{offset}) trên dữ liệu mảng thu; sai số trong thời gian lấy mẫu (time sampling); độ lệch pha giữa 2 tín hiệu (phase_{offset}) ở trạng thái ban đầu; cùng với vấn đề sai số (f_{offset}) trong tần số thu của từng SDR. Khi hiệu chỉnh hoàn tất, hệ SDR đồng bộ thì mới sử dụng được dữ liệu của chúng cho thuật toán DOA để ước lượng hướng sóng đến. Bảng 2.3 bên dưới nêu ra các phương pháp được sử dụng để đồng bộ hệ BladeRF trong khóa luận và đưa thêm một số phương pháp khác có thể áp dụng cho các loại SDR khác.

Bảng 2.3. Phương pháp đồng bộ hệ BladeRF được sử dụng trong khóa luận

Vấn đề đồng bộ	Phương pháp giải quyết được sử dụng	Các phương pháp khác
DC_{offset} và cân bằng mẫu IQ	Hiệu chỉnh bằng tinh nắn trong thư viện của BladeRF	
Xung đồng hồ chủ	Chia sẻ xung đồng hồ chủ (CLK) qua chân SMB (System Management Bus) trên BladeRF.	<ul style="list-style-type: none"> * Sử dụng thêm phần cứng: <ul style="list-style-type: none"> – Silicon Labs SI5338 Clock Generator Evaluation Board, sử dụng cùng loại IC SI5338 như trên BladeRF cho phép chia xung tối đa đến 8 thiết bị khác.
$\text{Sample}_{\text{offset}}$	Phân tích tương quan chéo của tín hiệu mảng thu.	<ul style="list-style-type: none"> * Sử dụng thêm phần cứng: <ul style="list-style-type: none"> – Noise Source Card và tương quan chéo của nhiễu Gauss, chuyển đổi qua lại giữa Noise Source Card và RF từ anten qua giao tiếp I2C [12]. – Sử dụng chân MIMO có trên các SDR cao cấp (BladeRF, USRP), trên BladeRF x115; chân J71 pin 4. * Sử dụng phần mềm: <ul style="list-style-type: none"> – Sử dụng một kênh tham chiếu cố định (FM, DVB-T, GSM, ...) được thực hiện trong dự án Multi-RTL; sử dụng kênh BCCH của GSM để đồng bộ [17].
Time sampling	Do đĩa chia sẻ CLK nên bỏ qua công đoạn này.	<ul style="list-style-type: none"> * Sử dụng thêm phần cứng: <ul style="list-style-type: none"> – Sử dụng chân J71 pin 4, tính năng Trigger trong bladeRF – <i>cli</i> Khóa xung lấy mẫu của các BladeRF với nhau.
f_{offset}	Công cụ kalibrate-bladeRF do nhà sản xuất cung cấp.	<ul style="list-style-type: none"> * Sử dụng thêm phần cứng: <ul style="list-style-type: none"> – Cấp tín hiệu chuẩn: 1 xung/giây hoặc 10 MHz ở mức điện áp 1,8 V qua chân J71 pin 1 để hiệu chỉnh VCTCXO.
$\text{Phase}_{\text{offset}}$	Sử dụng thuật toán MUSIC tìm độ lệch pha giữa 2 tín hiệu.	<ul style="list-style-type: none"> * Sử dụng thêm phần cứng: <ul style="list-style-type: none"> – Sử dụng tín hiệu từ máy phát chuẩn như ở [25], cấp đến các SDR, ước lượng độ lệch pha qua công thức: $\phi = \tan^{-1} \left(\frac{\text{Im}(x(k))}{\text{Re}(x(k))} \right)$ <p>với Re và Im lần lượt là phần thực và ảo của mẫu IQ. Sau đó, bù thêm pha cho tín hiệu trễ pha:</p> $\Delta\phi = \phi_1 - \phi_0$ $\hat{x}_1 = \mathbf{x}_1 e^{j\Delta\phi}$

Chi tiết các bước và kết quả đồng bộ bộ hê BladeRF x115 như sau:

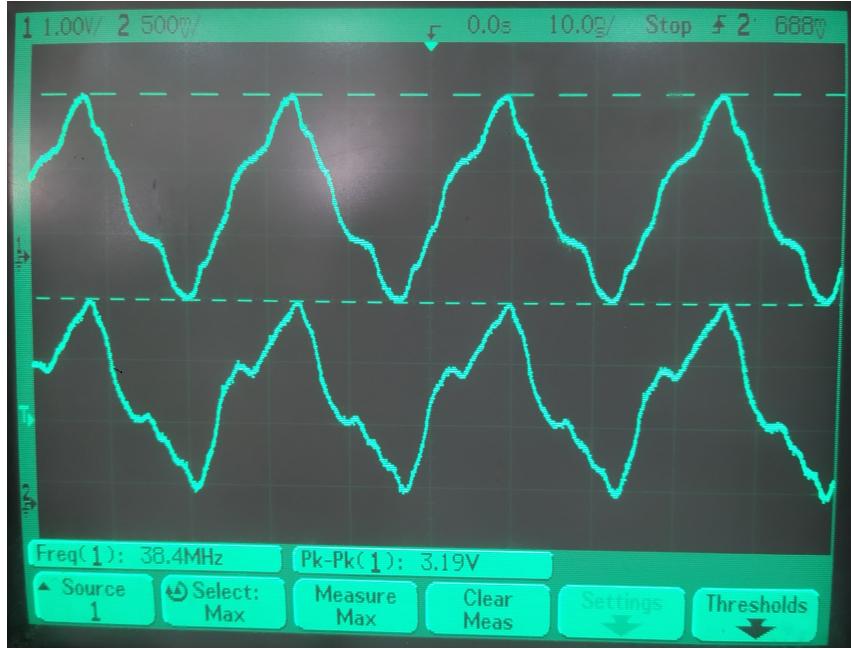
* Đồng bộ xung đồng hồ giữa các thiết bị SDR mảng thu

Việc đầu tiên cần làm với một hệ sử dụng nhiều SDR, hay các hệ truyền thông khác, đó là phải chia được xung đồng hồ chủ trên một thiết bị, chia cho các thiết bị còn lại, đảm bảo chắc chắn rằng chúng chạy trên cùng một xung đồng hồ. Với BladeRF, điều này đã được nhà sản xuất tích hợp sẵn, với cổng SMB (System Management Bus) là chân J62 trên mạch hay *EXT CLK* trong hình 2.6. Hình 2.10 và 2.11 lần lượt là xung đồng hồ tham chiếu của BladeRF chủ, và xung đồng hồ tham chiếu sau khi chia cho một BladeRF khác.

Thông thường, với thiết lập trong *bladeRF-cli* (client điều khiển của BladeRF) cổng SMB cho phép xuất xung tham chiếu (REF CLK) ở tần số 38,4 MHz, điện áp $V_{pp} = 3,9$ V, chỉ đủ để chia cho 1 BladeRF khác, do khi hao hụt công suất mỗi khi kết nối thêm BladeRF, yêu cầu mức điện áp tối thiểu của REF CLK ở đầu vào là 3,3 V. Vì vậy để sử dụng được nhiều BladeRF trong mảng thu hơn, bắt buộc phải cần có thêm bộ chia xung bù công suất, hoặc máy phát xung chuẩn.



Hình 2.10. Xung đồng hồ tham chiếu đầu ra của BladeRF



Hình 2.11. Xung đồng hồ tham chiếu được chia sẻ

* Hiệu chỉnh f_{offset}

f_{offset} hay chính xác hơn là tần số sóng mang offset CFO (Carrier Frequency Offset), luôn tồn tại khi LO (Local Oscillator) chịu ảnh hưởng bởi nhiễu điện, chênh lệnh nhiệt độ, ... Những độ lệch này có thể tạo ra f_{offset} và $\text{phase}_{\text{offset}}$ ngẫu nhiên. Trên các SDR thương mại, các thông số này đã được tính toán, và được bù lại bởi VCTCXO với độ chính xác biểu thị bởi thông số PPM (Parts Per Million). Ví dụ như BladeRF được cân chỉnh ở mức 1 PPM ở nhà máy thì f_{offset} tối đa nếu phát/thu tín hiệu ở tần số sóng mang 923 MHz tương đương:

$$f_{\text{offsetmax}} = \frac{f_c \times \text{PPM}}{10^6} = \frac{923 \times 10^6 \times 1}{10^6} = 923 \text{ Hz} \quad (2.1)$$

923 Hz là khá lớn, để giảm đi sai số này, nhà sản xuất cung cấp công cụ kalibrate-bladeRF [15], cho phép người dùng sử dụng tín hiệu từ 1 kênh FCCH (Frequency Correction Channel) đường xuống từ trạm cơ sở GSM ở gần để hiệu chỉnh lại VCTCXO có thể xuống mức 0,014 PPM, tương đương với việc sai số 923 Hz như trên có thể giảm xuống 10 Hz. Kết quả thực tế như trên hình 2.12 và 2.13, sau khi hiệu chỉnh kết quả thu được sai số chỉ 5 Hz ở 923 MHz và độ lệch $\Delta f_{\text{offset}} = 20 \text{ Hz}$.

* Hiệu chỉnh DC_{offset} và cân bằng mău IQ

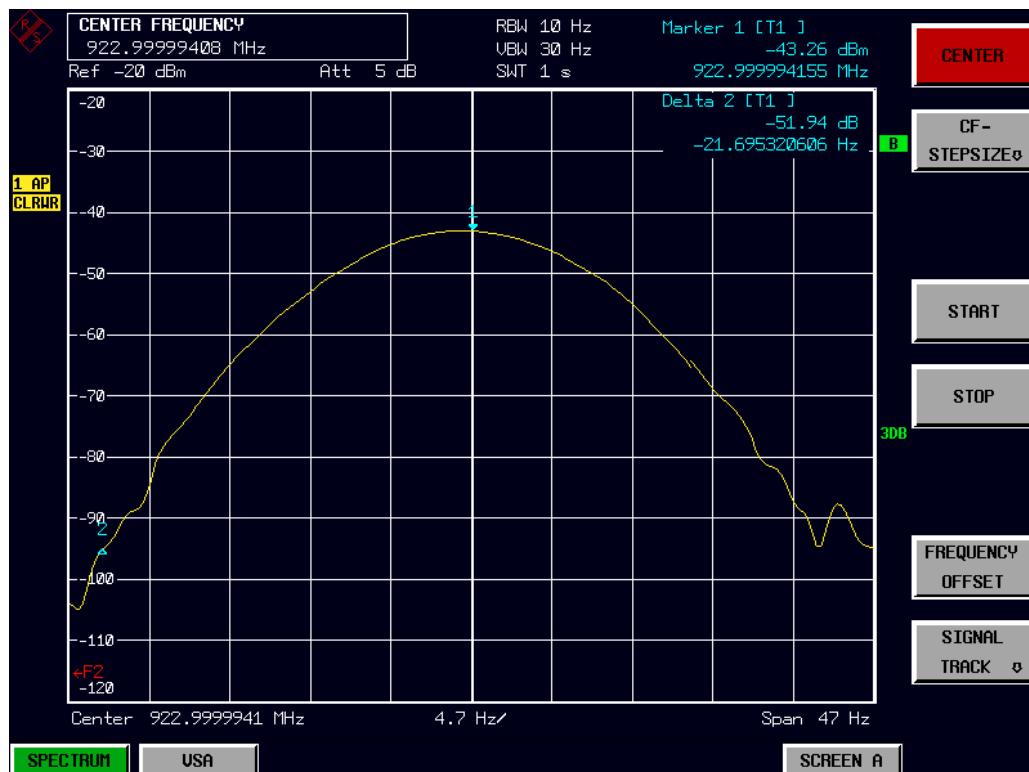
Do được thiết kế như một bộ nhận chuyển đổi trực tiếp (Direct-conversion receiver) hay bộ nhận không có tần số trung gian (zero-IF receiver) nên trong kết quả mău đầu ra

```

haiSon98@HaiSon: ~/kalibrate-bladeRF/src
File Edit View Search Terminal Help
.
.
.
.
.
.
average      [min, max]      (range, stddev)
+ 175Hz      [-31, 229]      (260, 59.182804)
overruns: 0
not found: 0
=====
DAC: 0xa525
.
.
.
.
.
.
average      [min, max]      (range, stddev)
+ 95Hz       [-209, 225]      (434, 151.637741)
overruns: 0
not found: 0
Found lowest offset of 13.639815Hz at 944.800000MHz (0.014437 ppm) using DAC tri
m 0xa520
haiSon98@HaiSon:~/kalibrate-bladeRF/src$ 

```

Hình 2.12. Kết quả cân chỉnh tần số với kalibrate-bladeRF



Hình 2.13. Kết quả thu được trên máy phân tích phổ

từ BladeRF, khi được đưa lên phổ FFT, sẽ luôn có một đỉnh phổ ở tần số trung tâm 0 Hz biểu diễn cho điện áp DC, hiện tại BladeRF chưa có phần cứng để bù vào phần DC này nên chỉ có thể giảm thiểu về mức thấp.

Nhà sản xuất đã tích hợp sẵn việc hiệu chỉnh trong thư viện của BladeRF, chi tiết có tại [14]. Ngoài việc hiệu chỉnh trên phần cứng BladeRF, cần sử dụng thêm khối DC

Blocker trên GNU Radio để loại bỏ hoàn toàn thành phần DC.

* Hiệu chỉnh sample_{offset}

Trong khóa luận này sẽ sử dụng tương quan chéo của tín hiệu thu để ước lượng số mẫu trễ gây ra bởi phần cứng BladeRF và USB. Đây là phương pháp có tính linh hoạt cao, có thể đồng bộ lượng mẫu đầu vào lớn (chục triệu mẫu), tuy nhiên bị giới hạn độ chính xác bởi môi trường truyền.

Tương quan chéo biểu hiện cho sự giống nhau của 2 tín hiệu, trong trường hợp này, có thể được dùng để ước lượng trễ mẫu giữa tín hiệu thu được từ 2 BladeRF bằng việc sử dụng một tín hiệu tham chiếu $x(t)$. Trước hết, công thức toán học của tương quan chéo được ký hiệu $x_1 * x_2$ giữa 2 tín hiệu phức [20]:

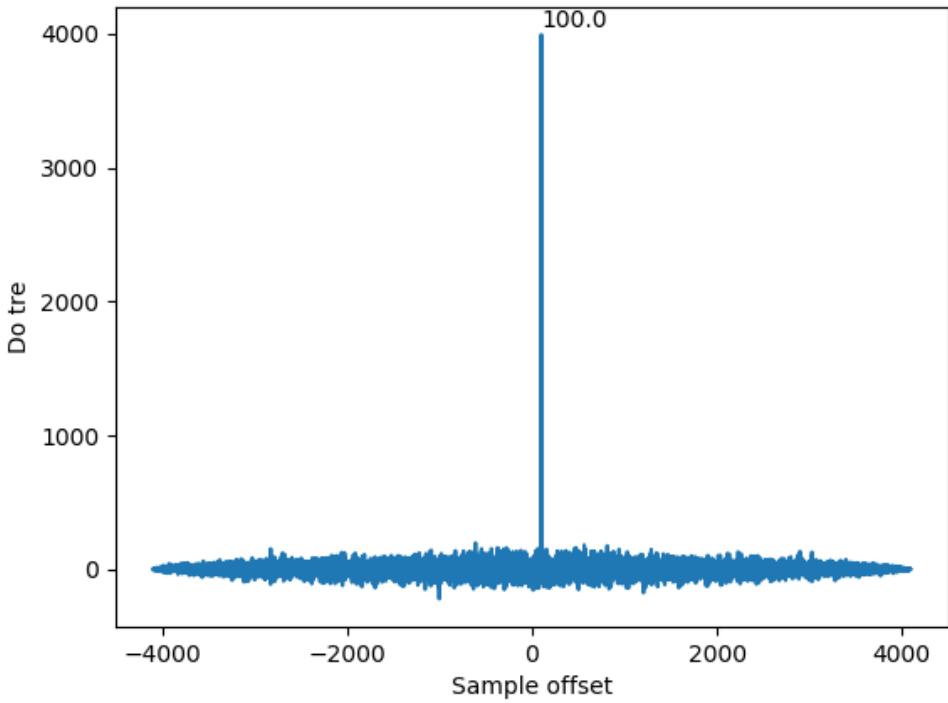
$$\begin{aligned} x(t) &: \text{Tín hiệu cơ sở} \\ x_1(t) &= x(t) \\ x_2(t) &= x(t - T) \\ x_1 * x_2(\tau) &\triangleq \int_{-\infty}^{+\infty} x_1^*(t)x_2(t + \tau)dt \\ T &= \operatorname{argmax}(x_1 * x_2) \end{aligned} \tag{2.2}$$

với $*$ là ký hiệu của liên hợp phức, chỉ có nghĩa khi x_1 là số phức, T là thời gian trễ giữa 2 tín hiệu. Rời rạc công thức tương quan chéo (2.2) để áp dụng cho các mẫu:

$$x_1 * x_2(n) = \sum_{m=-\infty}^{\infty} x_1^*(m).x_2(m + n) \tag{2.3}$$

tương tự thì số mẫu trễ tương ứng là $n = \operatorname{argmax}(x_1 * x_2)$. Trên hình 2.14 là ví dụ về việc sử dụng tương quan chéo của nhiễu Gauss để tìm độ trễ mẫu do phần cứng gây ra, được mô phỏng là 100 mẫu.

Cũng sử dụng tính tương quan của tín hiệu để tìm độ trễ, tuy nhiên thay vì tính toán ngay trên miền thời gian, việc chuyển sang miền tần số sẽ giúp giảm đáng kể lượng tính toán cần thiết do sử dụng khối FFT trên GNU Radio hỗ trợ chia đa luồng xử lý. Sử dụng các phép biến đổi FFT, IFFT để tính toán trễ giữa hai tín hiệu [13]. Áp dụng trên GNU Radio, với các khối có sẵn FFT, IFFT, Multipy Conjugate, dữ liệu đầu vào là tín hiệu được điều chế tần số băng hẹp (Narrow band FM - NBFM) được chia thành 2 nguồn, nguồn đầu tiên bị trễ 1000 mẫu so với nguồn thứ 2, lưu đồ tương ứng trên GNU Radio như hình 2.15(a) và kết quả đầu ra biểu diễn qua QT GUI trên hình 2.15(b) với vị trí đỉnh



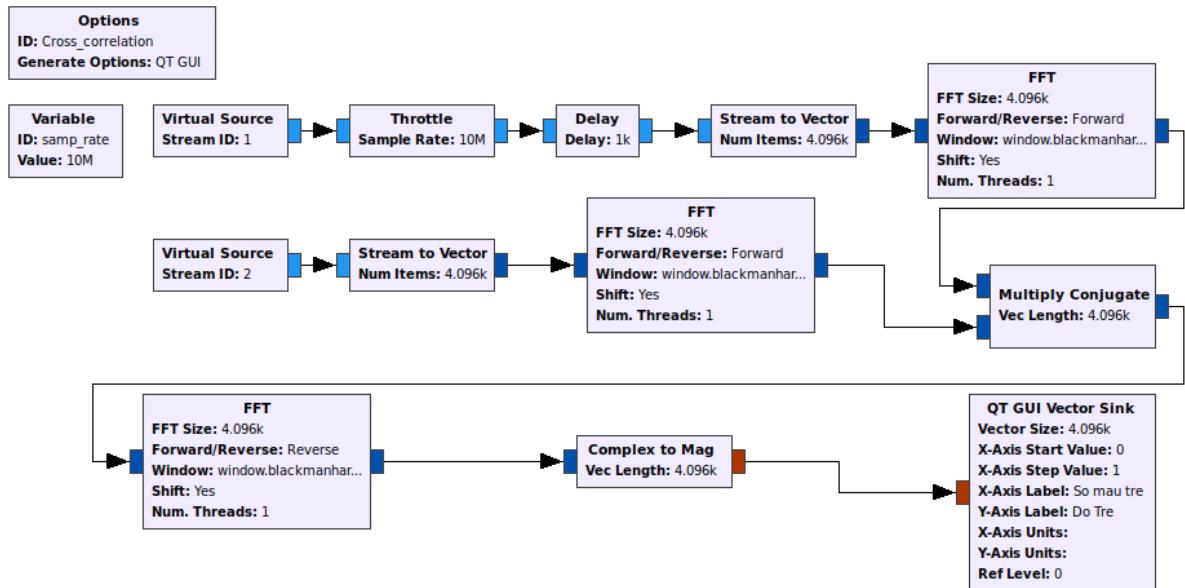
Hình 2.14. Tương quan chéo của 2 nhiễu Gauss

phổ đúng bằng số lượng trễ mô phỏng.

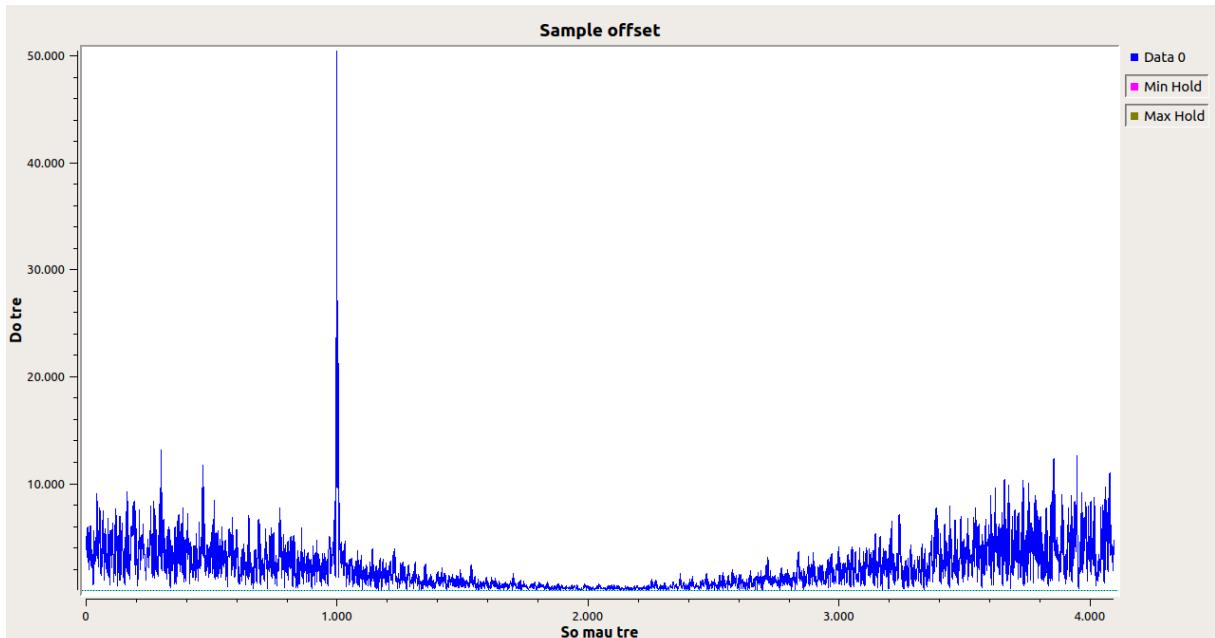
$$\begin{aligned}
 \mathbf{x}_1(t) &\Leftrightarrow \mathbf{x}_1(f) \\
 \mathbf{x}_2(t) &\Leftrightarrow \mathbf{x}_2(f) \\
 \mathbf{x}_1 * \mathbf{x}_2(\tau) &\Leftrightarrow \mathbf{x}_1 * \mathbf{x}_2(f) \\
 \mathbf{x}_1 * \mathbf{x}_2(f) &= \mathbf{x}_1^*(f)\mathbf{x}_2(f)
 \end{aligned} \tag{2.4}$$

Sau khi đã xác định được số mẫu trễ, sử dụng khối Delay, trễ mẫu của khối nguồn đang chạy nhanh đi một lượng $\text{argmax}[\mathbf{x}_1 * \mathbf{x}_2(f)]$. Flow-graph 2.15(a) được viết gọn vào một khối duy nhất Sample Offset như hình 2.16 để tiện cho việc xử lý, $\text{sample}_{\text{offset}}$ được lưu vào một biến và phản hồi lại về đầu vào để căn chỉnh dữ liệu khớp với nhau về mặt thời gian sẵn sàng cho các khối xử lý tiếp theo trong GNU Radio.

Nhược điểm của phương pháp này: nếu tín hiệu đầu vào có tính tương quan cao, kết quả đầu ra sẽ không chính xác tùy thuộc vào sự tương quan đó, mô phỏng cùng với kết quả mô phỏng của vấn đề này được trình bày trong mục 3.1.2 tại bảng 3.1 và 3.2; nếu độ trễ mẫu quá lớn, thông thường lớn hơn 1/2 số mẫu đầu vào khối FFT thì không thể tìm được độ trễ của các nguồn. Vì vậy, số lượng phần tử được đưa vào FFT có thể nâng lên đến hàng trăm nghìn phần tử để chắc chắn rằng thu được độ trễ khớp, nhưng để giảm thiểu tính toán, chỉ cần thực hiện hiệu chỉnh $\text{sample}_{\text{offset}}$ ngay khi hệ thống khởi động và lưu giá trị trễ cố định cho toàn bộ thời gian chương trình chạy. Do một khía cạnh dữ liệu



(a) Flow-graph mô phỏng ước lượng độ trễ giữa các đầu vào



(b) Tương quan chéo của tín hiệu đầu vào

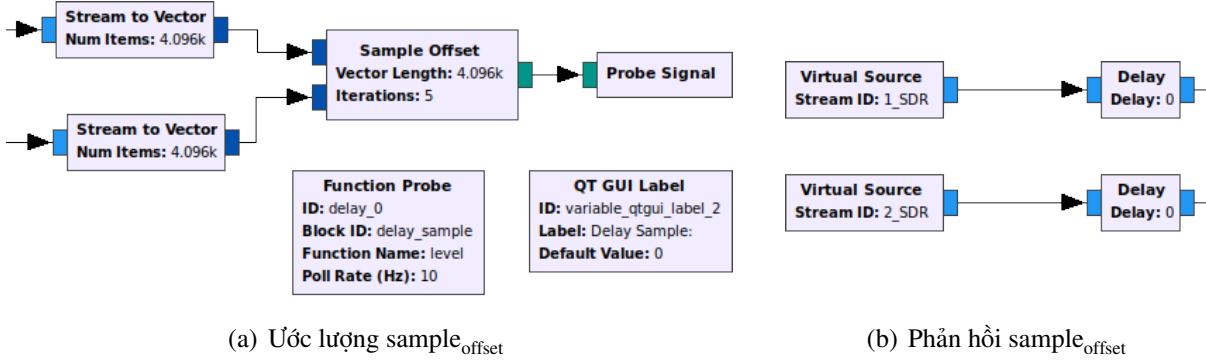
Hình 2.15. Xác định độ trễ mẫu trên GNU Radio

từ SDR đã vào được GNU Radio, lượng trễ này sẽ giữ ở mức ổn định.

* Hiệu chỉnh phase_{offset}

Tiếp đến là căn chỉnh lại pha giữa các SDR, sao cho tất cả pha từ dữ liệu là thẳng hàng trước khi ước lượng DOA, điều này là bắt buộc ngay cả khi cố định xung lấy mẫu qua chân J71-4, do việc hiệu chỉnh sample_{offset} chỉ có độ phân giải nhỏ nhất một mẫu, và sự sai khác nhỏ hơn một mẫu chính là sai khác pha ban đầu.

Sử dụng một tín hiệu tham chiếu bên ngoài đặt ở góc 90° để căn chỉnh pha, do được



Hình 2.16. Uớc lượng và phản hồi hiệu chỉnh $\text{sample}_{\text{offset}}$

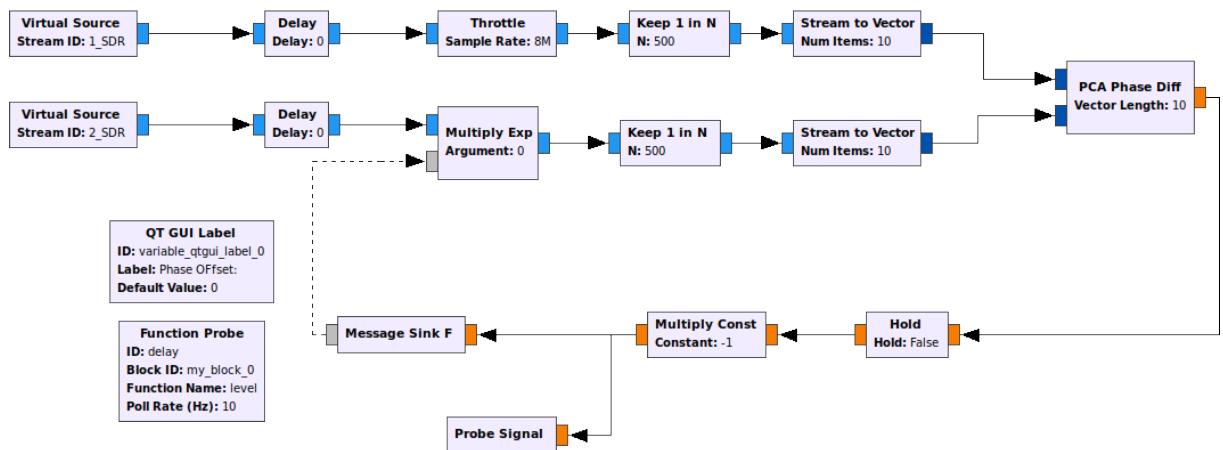
truyền qua kênh truyền với nhiễu, đa đường, suy giảm,... nếu ước lượng sai khác pha trực tiếp: $\Delta\phi = \phi_1 - \phi_0$, sự sai khác pha tính toán ra sẽ phát sinh những sai số ngẫu nhiên khó để căn chỉnh chính xác pha về trạng thái cân bằng. Vì vậy, dưới đây là phương pháp sử dụng ngay thuật toán MUSIC ước lượng DOA để tính toán ra độ lệch pha giữa các tín hiệu.

Căn chỉnh $\text{phase}_{\text{offset}}$ trên GNU Radio, để đảm bảo pha ban đầu trên các SDR như nhau, sử dụng một tín hiệu bên ngoài là nguồn tham chiếu, đặt tín hiệu ở góc 90° . Sau khi hiệu chỉnh được $\text{sample}_{\text{offset}}$ như ở phần trước. Tín hiệu thu được sẽ tồn tại độ lệch pha ($\text{phase}_{\text{offset}}$), các bước chính để căn chỉnh:

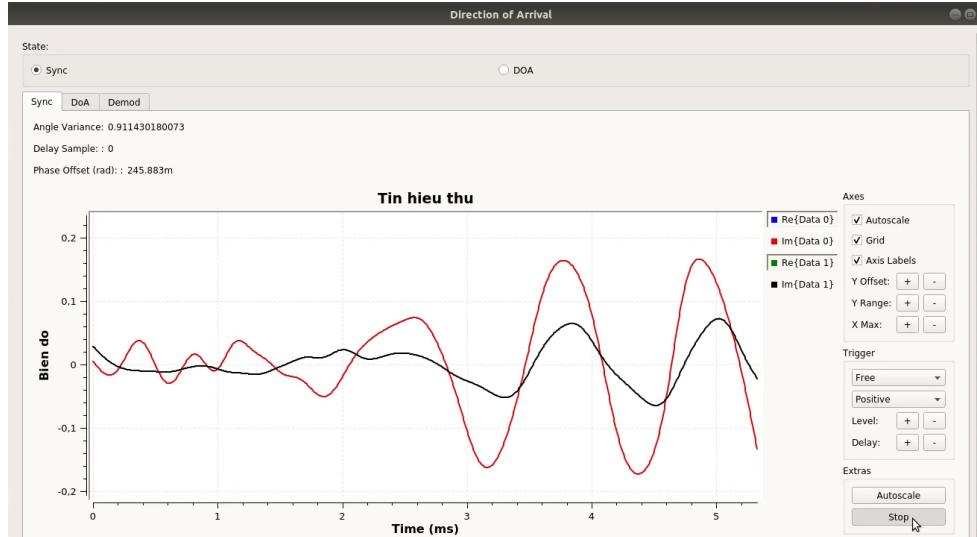
- Tính toán ma trận hiệp phương sai R_x của các tín hiệu.
- Tìm 2 ma trận giá trị riêng (Λ) và vector riêng (E) từ ma trận hiệp phương sai.
- Ước lượng sự sai khác pha của SDR bằng thuật toán MUSIC [24].
- Phản hồi sai khác pha về một khối Multiply Exp (hay $e^{j\Delta\phi}$).
- Căn chỉnh liên tục đến ngưỡng cân bằng khi phương sai giảm xuống thấp.
- Chuyển đổi hệ từ trạng thái đồng bộ sang ước lượng DOA.

Tất cả các bước được gộp thành 3 khối PCA Phase Diff, Hold và Multiply Exp, lưu đồ của hiệu chỉnh như hình 2.17.

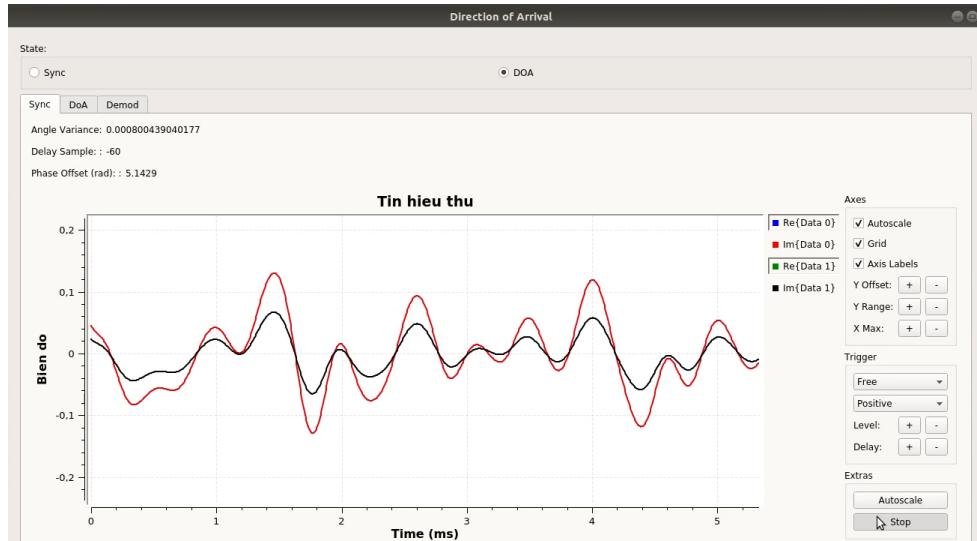
Kết quả thực nghiệm trên hình 2.18, khi phát NBFM ở tần số 923 MHz trên 1 BladeRF, 2 BladeRF thu, tính toán $\text{sample}_{\text{offset}}$ và căn chỉnh pha.



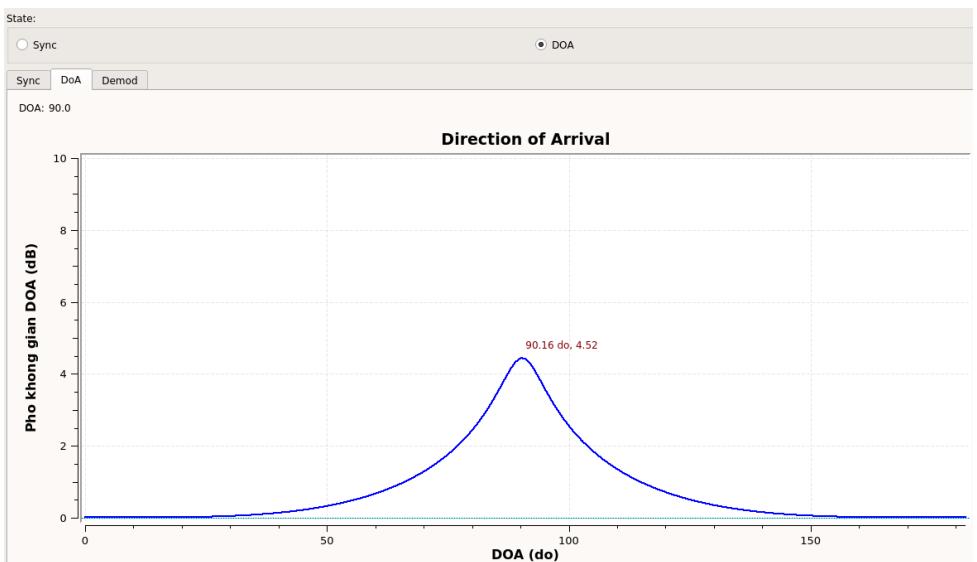
Hình 2.17. Ước lượng và phản hồi phase_{offset}



(a) Tín hiệu ban đầu chưa đồng bộ



(b) Tín hiệu sau khi dịch mẫu và căn chỉnh pha



(c) Góc đầu ra ban đầu khi hệ được đồng bộ

Hình 2.18. Kết quả đồng bộ hệ SDR

CHƯƠNG 3

KẾT QUẢ MÔ PHỎNG, THỰC NGHIỆM HỆ THỐNG

Trong chương này, trình bày việc triển khai thuật toán MUSIC trên GNU Radio với dữ liệu: mô phỏng từ Matlab; giả lập BladeRF từ các khối trên GNU Radio; BladeRF thực. Từ đó rút ra so sánh, đánh giá kết quả thực tế so với mô phỏng.

3.1. Mô phỏng hệ thống trên GNU Radio

3.1.1. Mô phỏng với dữ liệu Matlab

Trước hết, sử dụng tín hiệu mô phỏng từ Matlab xuất ra file *.mat* rồi đưa vào GNU Radio và thực hiện ước lượng hướng sóng đến từ dữ liệu, việc này đảm bảo việc lập trình khối hoạt động hiệu quả và giúp đánh giá được sai số lý tưởng của thuật toán MUSIC trước khi mô phỏng tín hiệu BladeRF. Vẫn áp dụng mô hình tín hiệu như ở mục 1.3, các thông số mô phỏng trên bảng 3.1.

Bảng 3.1. Các thông số mô phỏng hệ DOA

Thông số	Giá trị	Chú thích
f	923 Mhz	Tần số.
λ	$c \div f = 0,325$ m	Bước sóng.
M	2	Số phần tử anten mảng thu.
d	$0,5 \times \lambda$	Khoảng cách giữa các phần tử anten mảng thu (ULA).
D	1	Số phần tử nguồn.
angle	$85 \times \pi \div 180$	Góc tới của tín hiệu.
gain	0	Hệ số khuếch đại.
SNR	70 dB	Tỷ lệ tín hiệu tạp âm.
k	$2 \times \pi \div \lambda$	Hệ số sóng.
K	1024	Số mẫu thu thập.

$$\mathbf{s} = \left[20^{\frac{SNR}{10}} \times \text{randn}(1, K) + j \times \text{randn}(1, K) \right] \quad (3.1a)$$

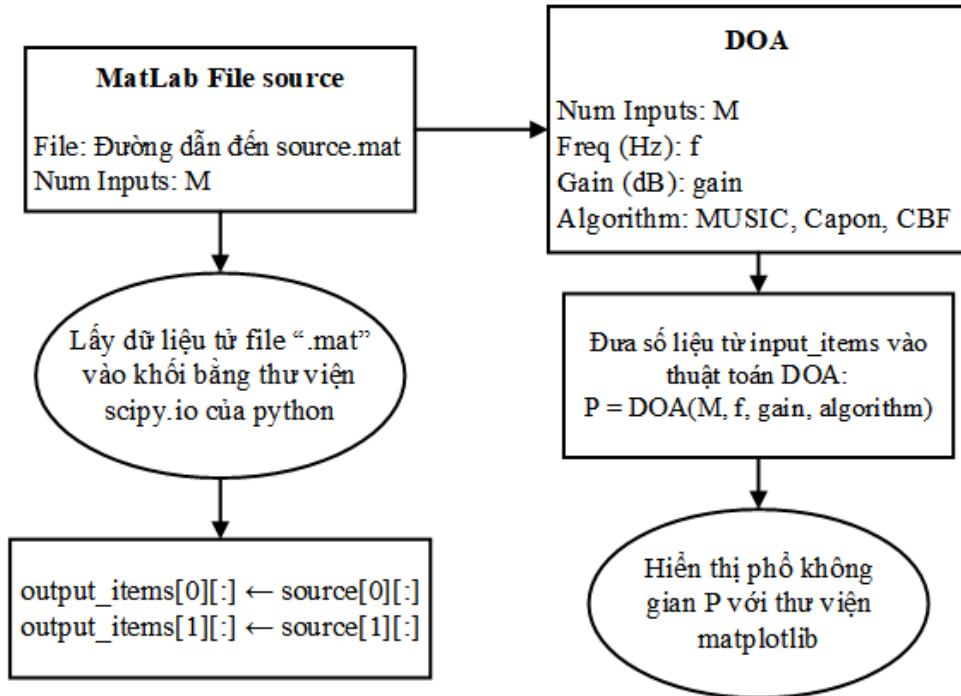
$$\mathbf{A} = \left[10^{\frac{gain}{10}} \times \exp\{j \times k \times (0 : M - 1) \times d \times \cos(\text{angle})\} \right]^T \quad (3.1b)$$

$$\mathbf{n} = \left[\text{randn}(M, K) + j \times \text{randn}(M, K) \right] \quad (3.1c)$$

$$\mathbf{x} = [\mathbf{As} + \mathbf{n}] \quad (3.1d)$$

Sử dụng Matlab tạo mô phỏng dữ liệu đầu vào với những thông số như trên, thu được được ma trận $\mathbf{x}(t) \in \mathbb{C}^{2 \times 5000}$, lưu trữ vào file *source.mat* bằng hàm có sẵn của Matlab. Việc tiếp theo là đưa dữ liệu mô phỏng vào GNU Radio thông qua việc tạo

một khối nguồn cho file *source.mat*. Đối với mục đích này, Synchronous Blocks được sử dụng: một khối loại bỏ đầu vào và thay thế bằng file dữ liệu mô phỏng từ Matlab tạo thành khối nguồn; một khối loại bỏ đầu ra tạo thành khối đích để tính toán và hiển thị giao diện DOA. Sử dụng Python để tiết kiệm thời gian viết khôi, sơ đồ khôi của việc mô phỏng thực hiện như hình 3.1.



Hình 3.1. Sơ đồ làm việc DOA với dữ liệu mô phỏng Matlab

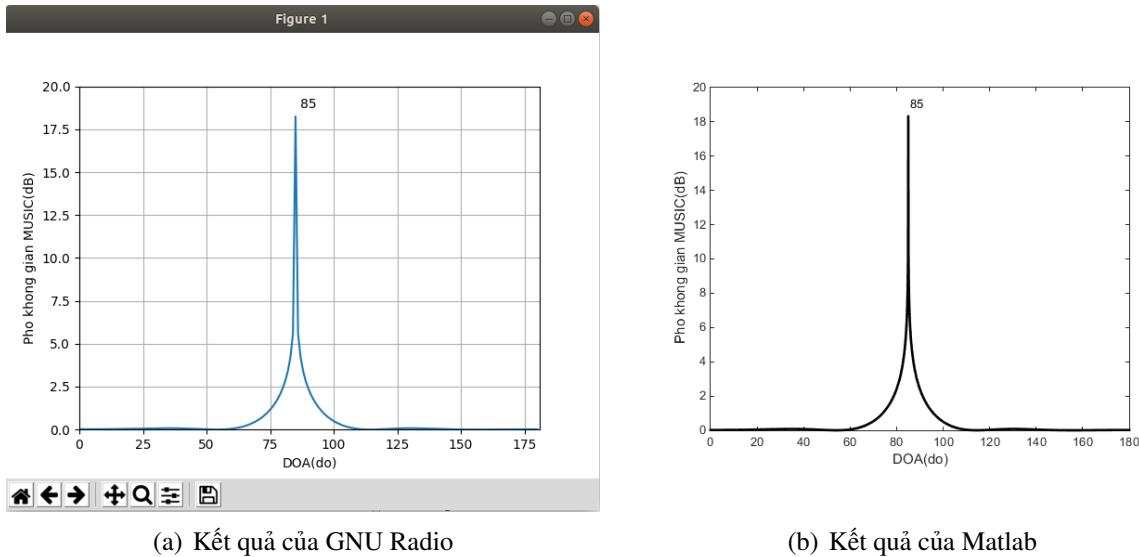
Liên kết hai khôi với nhau và nhập các thông số đầu vào của hệ và chạy hệ thống. Kết quả trên GNU Radio Companion thu được như phụ lục hình A.1. Tiến hành thay đổi góc đầu vào từ file dữ liệu Matlab và chạy hệ thống nhiều lần, mỗi lần sẽ thu được kết quả như hình 3.2.

Do dữ liệu mô phỏng là hoàn toàn khớp với các điều kiện lý tưởng, sai số của thuật toán MUSIC trên GNU Radio giống như trên lý thuyết [1, 18], sử dụng thông số CRB (Cramér–Rao bound) cho một nguồn để đánh giá sai số ước lượng:

$$\text{CRB}(\phi) = \mathbf{J}^{-1} \quad (3.2)$$

với \mathbf{J}^{-1} là ma trận Fisher, được biểu diễn ở phương trình (3.2):

$$\begin{aligned}
 \mathbf{J} &= K \cdot \text{trace} \left\{ \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\partial \phi} \mathbf{R}_x^{-1} \frac{\partial \mathbf{R}_x}{\partial \phi} \right\} \\
 &= \frac{2\text{SNR}^2}{(1 + \text{SNR}|a|^2)^2} [2(\Re(a^H \dot{a}))^2 + (1 + \text{SNR}|a|^2)(|a|^2|\dot{a}|^2 - |a^H \dot{a}|^2)]
 \end{aligned} \quad (3.3)$$

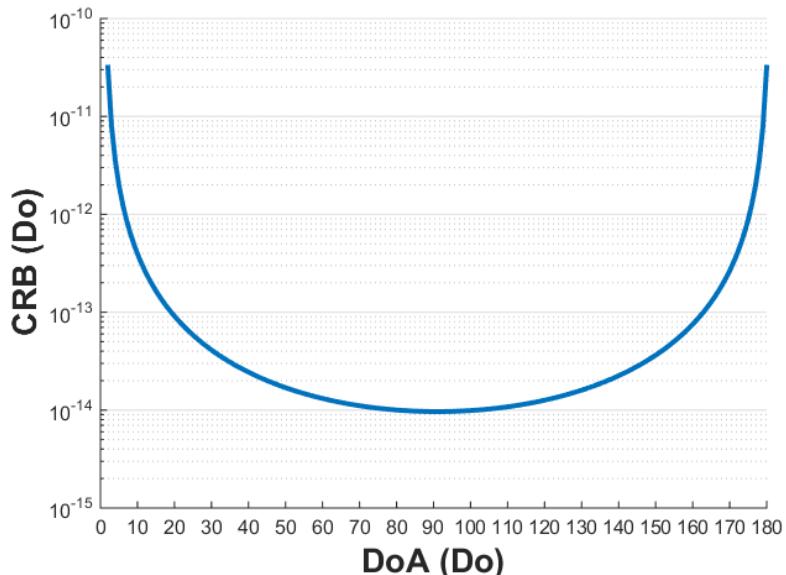


(a) Kết quả của GNU Radio

(b) Kết quả của Matlab

Hình 3.2. Kết quả ước lượng DOA với dữ liệu mô phỏng Matlab

với $\dot{\mathbf{a}}_\phi = \partial \mathbf{a} / \partial \phi$. Hình 3.3 biểu diễn CRB của mảng ULA với các tham số: $M = 2$, SNR = 70 dB, số mẫu $K = 1024$. Như vậy, thấy rằng mảng ULA có lỗi ước lượng nhỏ với các góc ở trục vuông và sai số lớn với các góc ở trục dọc so với mảng anten thu.



Hình 3.3. CRB của mảng anten ULA

3.1.2. Mô phỏng với dữ liệu giả lập BladeRF

Sau khi đã mô phỏng thành công với dữ liệu từ Matlab, tiếp tục mô phỏng dữ liệu BladeRF thực với tín hiệu đầu vào là âm thanh được điều chế NBFM (chi tiết tham số điều chế, phổ, SNR được trình bày ở mục 3.2.1). Vẫn sử dụng thông số mô phỏng như bảng 3.1, sau khi được mã hóa, chia tín hiệu thành 2 luồng đi qua kênh truyền với các thông số như bảng 3.2, các khối mô phỏng tương ứng trên GNU Radio được biểu diễn ở phụ lục hình B.1.

Bảng 3.2. Các thông số mô phỏng tín hiệu BladeRF qua kênh truyền

Mô phỏng	Thông số mô phỏng	Giá trị
Kênh truyền	Giá trị tạp âm	0,01 (1% biên độ tối đa của tín hiệu)
	Gieo hạt của tạm âm	1000
	Số tín hiệu đa đường	8
	Phân bố nhiễu đa đường	Rayleigh
	Gieo hạt nhiễu đa đường	1000
Phần cứng BladeRF	Trễ mẫu	1000 (mẫu)
	Độ lệch pha ban đầu	$45^\circ = 0,78539$ (rad)

Chạy mô phỏng với các tệp âm thanh khác nhau, mỗi tệp chạy 5 lần và ghi lại kết quả thông số đồng bộ và vector giá trị riêng từ hệ DOA, sau đó tính trung bình và đưa vào bảng 3.3.

Bảng 3.3. Kết quả mô phỏng đồng bộ

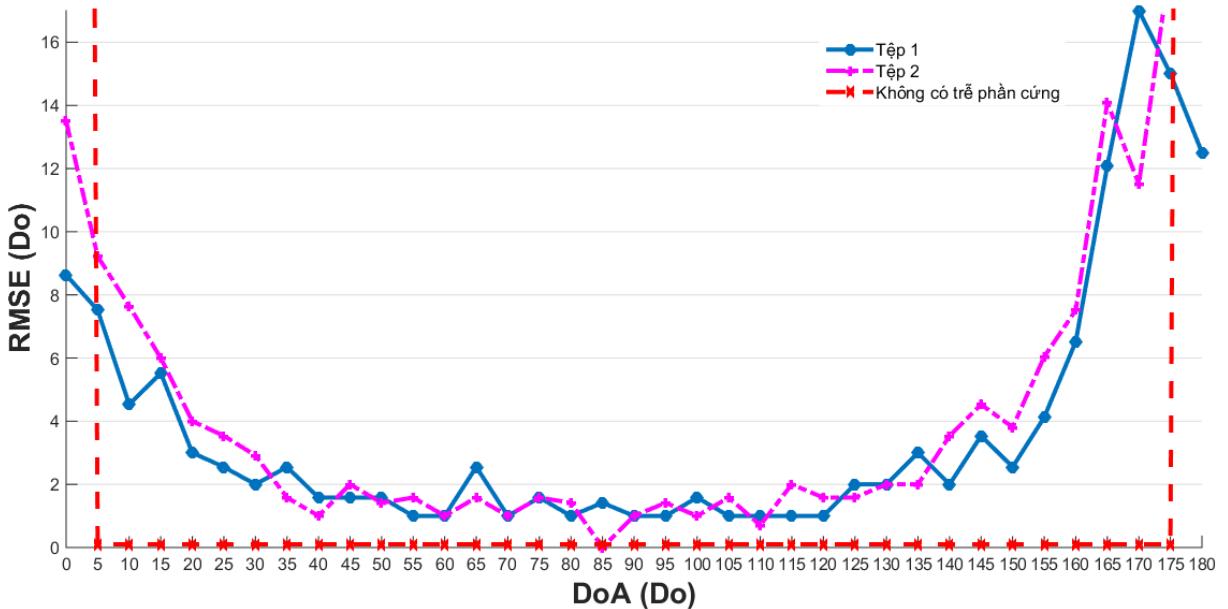
Nguồn tín hiệu	Độ trễ (mẫu)	Độ lệch pha (rad)	Vector giá trị riêng
Tệp âm thanh 1	994,6	0,72986	$0,00048259 \quad 0$ $0 \quad 0,04719653$
Tệp âm thanh 2	991,25	0,72999	$2,03512e-5 \quad 0$ $0 \quad 0,0863451$
Tệp âm thanh 1 bỏ qua phần cứng			$1,241364-5 \quad 0$ $0 \quad 0,0034562$

Từ bảng 3.3, nhận thấy việc ước lượng độ trễ bằng tương quan chéo của tín hiệu NBFM cho kết quả chưa hoàn toàn chính xác (< 10 mẫu), kéo theo đó sẽ là sai số khi ước lượng độ lệch pha ở mức 0,056 (rad). Thông số đánh giá còn lại là ma trận vector giá trị riêng, biểu hiện sự phân biệt giữa tín hiệu và tạp âm, với tỷ lệ trung bình 10^2 đến 10^3 , tín hiệu NBFM đủ để xuất hiện đỉnh phổ trên phổ không gian MUSIC.

Từ kết quả mô phỏng đồng bộ chưa hoàn toàn chính xác khi sử dụng NBFM, cần phải mô phỏng ước lượng hướng sóng đến dựa trên những kết quả đồng bộ phía trên để có đánh giá chính xác nhất. Kết quả sẽ được tính lỗi căn trung bình bình phương RMSE của góc thu được (ϕ) so với góc mô phỏng ($\bar{\phi}$) như trên công thức (3.3) với D là số lần thu thập kết quả, sai số được biểu diễn trên hình 3.4.

$$\text{RMSE} = \sqrt{\frac{1}{D} \sum_{i=1}^D (\phi_i - \bar{\phi}_i)^2} \quad (3.4)$$

Như vậy so với CRB hay mô phỏng hệ thống không chịu ảnh hưởng bởi trễ phần cứng gây ra bởi BladeRF thì dạng đồ thị của RMSE khi sử dụng NBFM làm tín hiệu nguồn phát vẫn giữ được đúng hình dạng là sai số nhỏ ở những góc gần 90° và tăng dần



Hình 3.4. RMSE: Mô phỏng dữ liệu BladeRF qua kênh truyền

ở hai biên phổ không gian. Tuy nhiên với các tệp âm thanh nguồn khác nhau, việc đồng bộ hệ BladeRF sẽ cho kết quả khác nhau và dẫn đến sai số góc ước lượng cũng là khác nhau nhưng không quá nhiều.

Kết luận lại, dù vẫn còn sai số trong việc đồng bộ hệ BladeRF tuy nhiên sai số vẫn đủ nhỏ để tiếp tục sử dụng tín hiệu NBFM thực hiện hệ DOA trên hệ thống BladeRF thực.

3.2. Thực nghiệm hệ thống

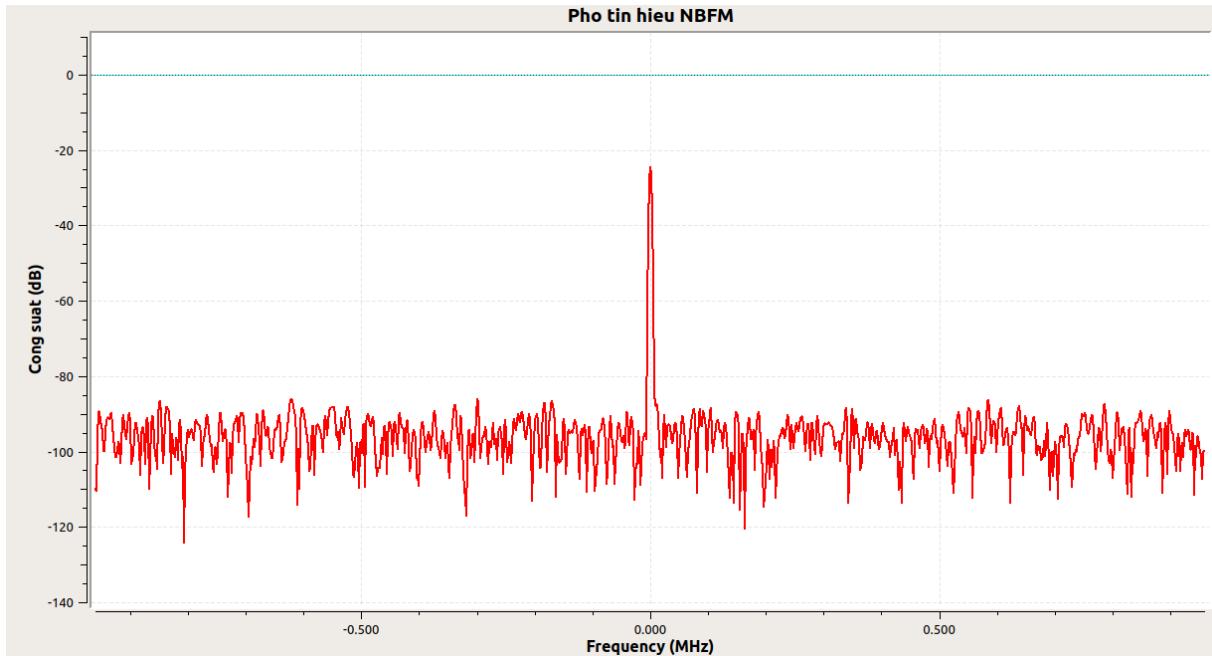
Với tiền đề là việc mô phỏng thành công được dữ liệu BladeRF trên GNU Radio, tiếp theo là thực nghiệm hệ thống DOA trên hệ BladeRF thực, trong phần này sẽ trình bày điều kiện và kết quả thực nghiệm, cuối cùng đưa ra nhận xét từ kết quả thu được.

3.2.1. Tín hiệu nguồn phát và thiết lập hệ thu

Dữ liệu thực tế được dùng cho hệ tìm hướng sóng đến rất đa dạng như: tín hiệu FM của các đài phát thanh, tín hiệu chuẩn DVB-T của đài truyền hình, GSM hay WiFi,... Trong khóa luận này, tín hiệu NBFM được lựa chọn để làm tín hiệu nguồn phát, vừa để đồng bộ hệ SDR thu, vừa để làm tín hiệu nguồn cho việc xác định hướng sóng đến. Lý do lựa chọn NBFM, vì đây là điều chế tín hiệu băng hẹp phù hợp với mô hình tín hiệu ban đầu của thuật toán.

NBFM: Điều chế NBFM trên GNU Radio với file nguồn là file âm thanh nén ở chuẩn WAV (Waveform Audio File Format). Flow-graph phụ lục hình C.1 là sơ đồ điều chế tiêu chuẩn NBFM cho tệp âm thanh WAV, với tín hiệu đầu ra băng hẹp 5,3 kHz (có phổ như hình 3.5, SNR trên 70 dB) và chỉ cần một bộ lọc thông thấp 5 kHz và khối giải

điều chế NBFM tiêu chuẩn là đủ để giải điều chế.



Hình 3.5. Phổ của tín hiệu tín hiệu NBFM

Đối với hệ thu, các bước đồng bộ và ước lượng DOA trình bày ở Chương 2 được kết hợp với nhau trên GNU Radio. Trước hết thực hiện các thiết lập trên *bladeRF-cli* như chia sẻ xung đồng hồ qua cổng SMB, hiệu chỉnh CFO của BladeRF. Sau đó sử dụng flow-graph trên phụ lục hình D.1 phía dưới để đồng bộ và ước lượng DOA.

3.2.2. Tình huống thực nghiệm

Bố trí hệ thu phát sử dụng BladeRF để triển khai việc ước lượng hướng sóng đến, trên hình 3.6 là hệ thống được cài đặt hoàn chỉnh, gồm 1 BladeRF phát và 2 BladeRF thu, cả 3 anten được sử dụng đều là anten không định hướng VERT2450. Cấu hình phần cứng và phần mềm của 2 máy tính thực hiện việc phát và thu trong khóa luận được nêu ra trong bảng 3.4. Tần số được chọn là 923 MHz, là tần số nằm trong khoảng giữa của đường lên và xuống trong GSM900.

Bảng 3.4. Thông số máy tính

Thông số	Bên phát	Bên thu
CPU	Intel Core I5-4800MQ	Intel Core I5-4210M
RAM	8 GB	8 GB
Hệ điều hành	Ubuntu 16.04 LTS	Ubuntu 18.04 LTS
Phiên bản GNU Radio	3.7.11	3.7.11

Điều kiện thực nghiệm là phòng 204, nhà G2, Đại học Công Nghệ - Đại học Quốc Gia Hà Nội, tiến hành thực nghiệm 350 lần, xác định hướng sóng đến từ các góc khác



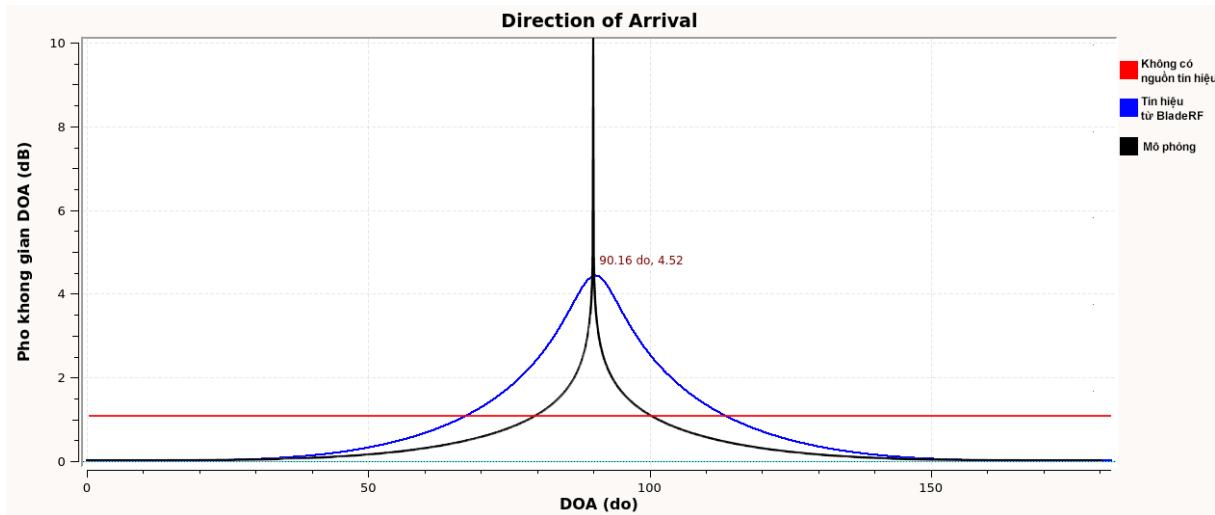
Hình 3.6. Sơ đồ bố trí hệ DOA thực

nhau, để giảm thiểu sai số chủ quan độ phân giải khi thực nghiệm ở mức 5° , bố trí hệ thống ở 3 vị trí khác nhau để đưa ra ảnh hưởng của môi trường đến độ chính xác của hệ DOA.

3.2.3. Kết quả thực nghiệm

* Phổ không gian

Hình 3.7 biểu diễn kết phổ không gian của thuật toán MUSIC trên GNU Radio trong trường hợp: có tín hiệu đến ở góc 90° , không có tín hiệu nào đến và tín hiệu mô phỏng từ Matlab.

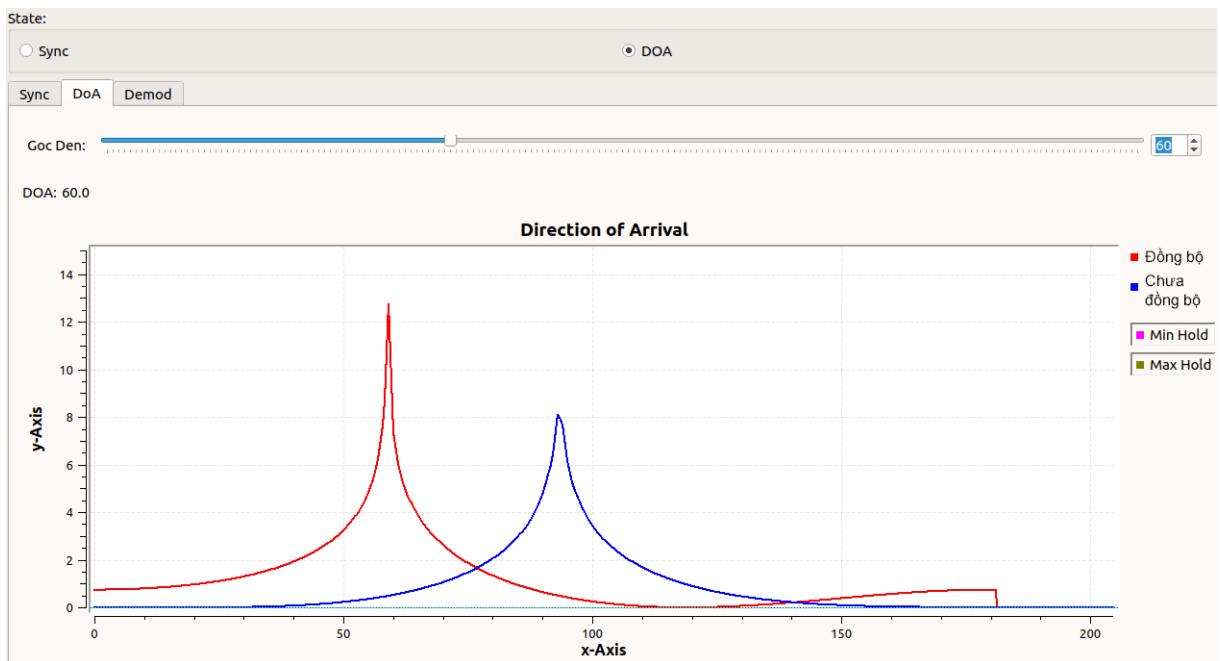


Hình 3.7. So sánh phổ không gian trên thực tế và mô phỏng

Hình dạng phổ không gian khi thực nghiệm là giống với kết quả khi mô phỏng, tuy nhiên về đỉnh phổ thì lại thấp hơn so với mô phỏng và độ rộng của đỉnh phổ cũng lớn hơn mô phỏng, điều này xảy ra do tín hiệu thực có biên độ chênh lệch với nhau lớn, đây là vấn đề do phần cứng, hệ số khuếch đại, việc khắc phục sẽ được thực hiện trong tương lai. Còn khi không có tín hiệu đến, phổ DOA đã cho kết quả đúng khi không xuất hiện đỉnh phổ nào.

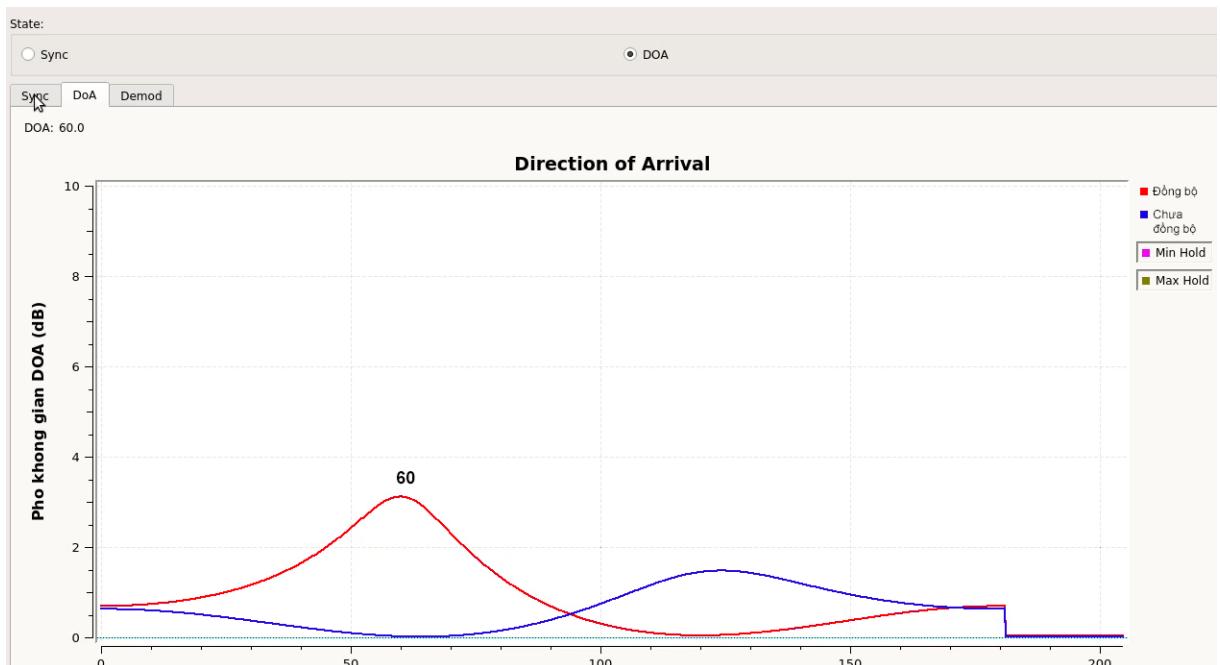
* Kết quả đồng bộ

Kết quả mô phỏng khi hệ thống DOA đồng bộ và không đồng bộ như hình 3.8, kết cho thấy sau khi được đồng bộ về lượng mâu trễ và pha sai lệch giữa các tín hiệu từ BladeRF, kết quả phổ DOA thu được sẽ giữ ở một đỉnh phổ ổn định và chính xác, còn khi hệ BladeRF chưa đồng bộ đỉnh phổ liên tục nhảy giữa các vị trí khác nhau, không thể xác định hướng sóng đến.



Hình 3.8. Kết quả DOA mô phỏng khi hệ SDR đồng bộ

Qua kiểm nghiệm thực tế, kết quả cũng thu được như trên mô phỏng tuy nhiên độ rộng của đỉnh phổ thì lớn hơn mô phỏng nhưng kết quả ước lượng vẫn chính xác khi hệ BladeRF được đồng bộ. Đối với hệ BladeRF chưa được đồng bộ, góc ước lượng được luôn thay đổi và không thể xác định hướng sóng đến.



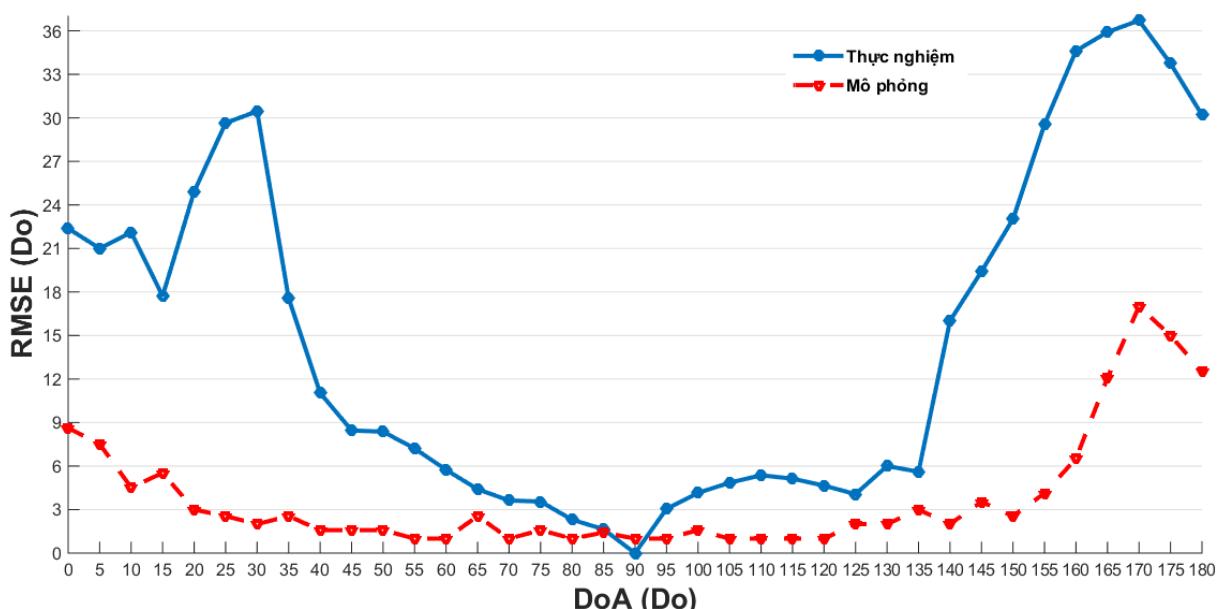
Hình 3.9. Kết quả DOA thực nghiệm khi hệ SDR đồng bộ

* Khảo sát lỗi ước lượng của hệ thống

- Kết quả ước lượng theo độ $\phi = 0 \rightarrow 180$

Hình 3.10 là RMSE của quá trình thực nghiệm ước lượng DOA trên BladeRF thực so với kết quả mô phỏng dữ liệu BladeRF. Nhận thấy, dạng biểu đồ RMSE thu được từ thực nghiệm là tương đồng với mô phỏng và CRB được ước lượng ở hình 3.3, sai số ở các góc gần 90° nhỏ, và tăng dần ở các góc ở hai biên của phổ không gian.

Ở khoảng góc từ 60° đến 135° , sai số dưới 6° , và do chỉ sử dụng 2 phần tử anten mảng thu nên mức sai số này là chấp nhận được.



Hình 3.10. RMSE: DOA thực nghiệm và mô phỏng

- Kết quả ước lượng theo vị trí

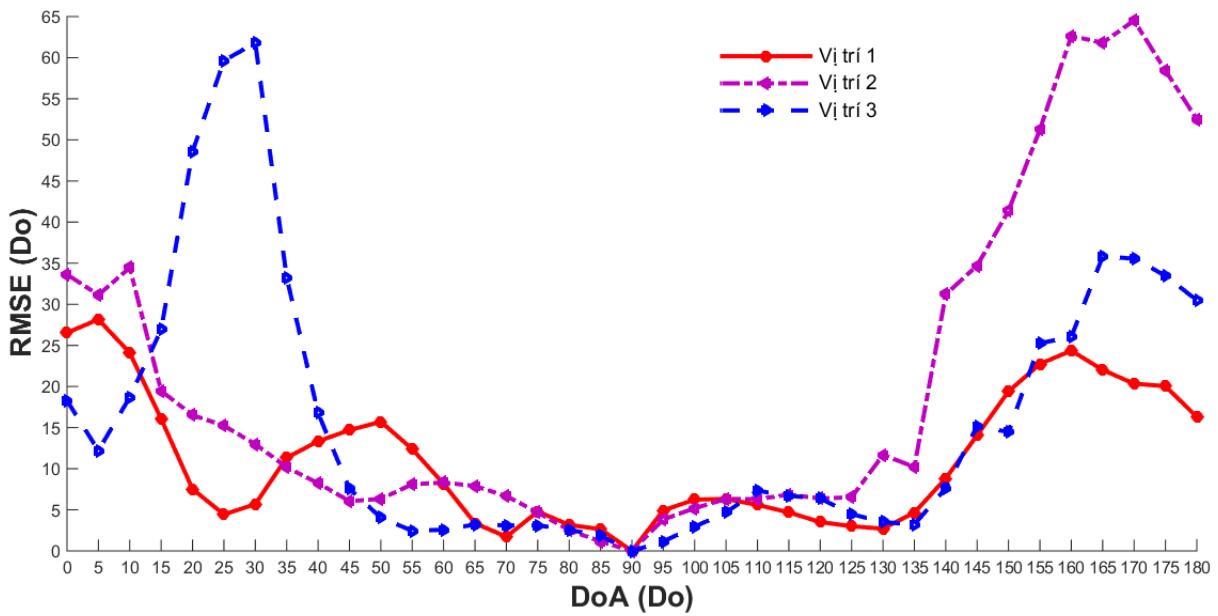
Thực nghiệm thêm để xem xét vị trí đặt hệ DOA hay kênh truyền có ảnh hưởng đến kết quả ước lượng hướng sóng đến hay không, đặt hệ ở 3 vị trí khác nhau, kết quả RMSE thu được như hình 3.11.

Từ kết quả RMSE thu được, có thể nhận thấy khoảng góc từ 60° đến 135° vẫn cho sai số thấp tương đương nhau ở cả 3 vị trí. Ở các góc ngoài khoảng trên, kết quả rất dễ bị ảnh hưởng bởi nhiều đa đường, gây ra sai số nhảy vọt ở một số khoảng góc có nhiều vật cản.

Kết luận hệ ước lượng hướng sóng đến, vẫn hoạt động ở các điều kiện kênh truyền khác nhau.

- Kết quả ước lượng với dữ liệu khác nhau

Tuy theo lý thuyết, thuật toán MUSIC áp dụng cho mô hình tín hiệu băng hẹp,



Hình 3.11. RMSE: DOA ở từng vị trí thực nghiệm

nhưng trong khóa luận này, sẽ mở rộng thêm và sử dụng tín hiệu băng rộng cho hệ DOA sử dụng thuật toán MUSIC.

Tín hiệu băng rộng được sử dụng là DVB-T, tín hiệu chuẩn thu truyền hình số mặt đất tiêu chuẩn ở Việt Nam. Việc điều chế, thông số, phổ DVB-T trên GNU Radio như sau:

DVB-T: Điều chế chuẩn DVB-T trên GNU Radio với file nguồn là file video nén ở chuẩn TS (Transport Stream). Flow-graph ở phụ lục hình E.1 truyền video ở băng thông 8 MHz, chòm sao QPSK, tốc độ mã 7/8, khoảng bảo vệ 1/32, tốc độ lấy mẫu 10 MHz, tần số 923 MHz. Chi tiết hơn về việc sử dụng BladeRF truyền nhận DVB-T có tại [4].

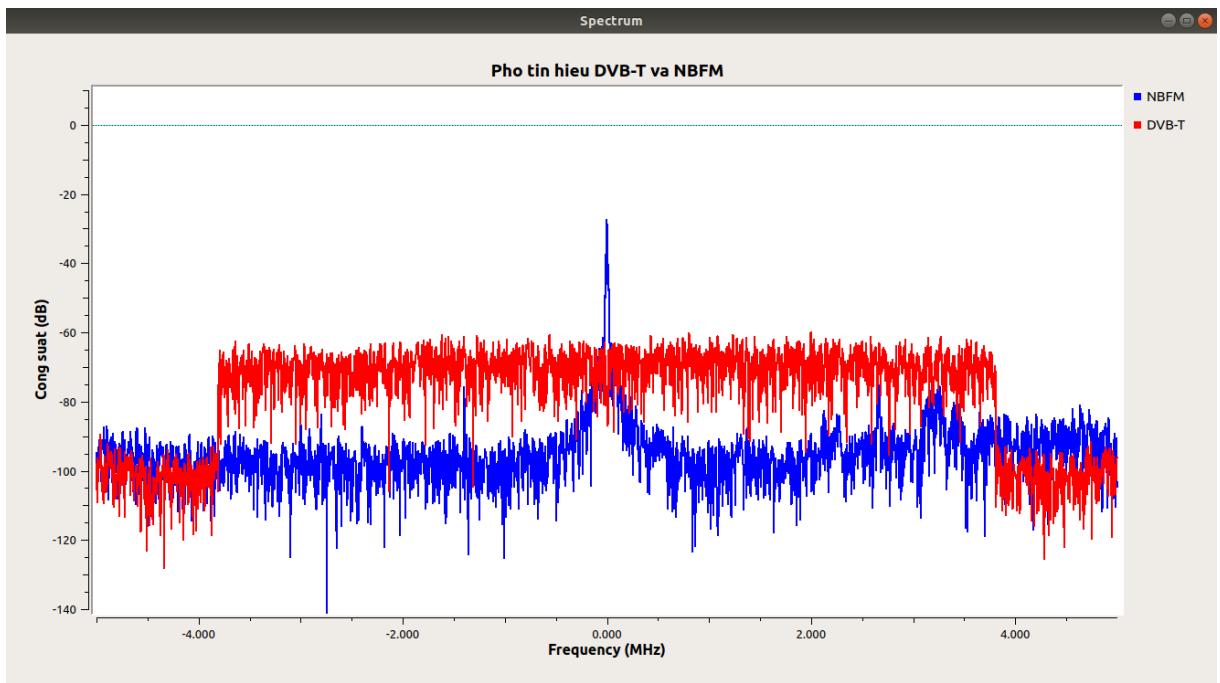
Phổ tín hiệu DVB-T được truyền, và thu lại như hình 3.12, nhận thấy do sử dụng OFDM nên phổ của DVB-T bị trải đều trên dải tần số, dẫn đến SNR ở mức 37,5 dB so với trên 70 dB của NBFM.

Qua mô phỏng BladeRF trên GNU Radio, kết quả đồng bộ từ tín hiệu DVB-T thu được như bảng 3.5. Với tính tương quan thấp hơn của NBFM, kết quả ước lượng sai số phần cứng BladeRF từ tín hiệu DVB-T là hoàn toàn khớp với các thông số mô phỏng.

Bảng 3.5. Kết quả mô phỏng đồng bộ bằng tín hiệu DVB-T

Loại tín hiệu	Độ trễ (mẫu)	Độ lệch pha (rad)	Vector giá trị riêng
DVB-T	1000	0,78539	1,4544877e-11 0 0 9,0942479e-04

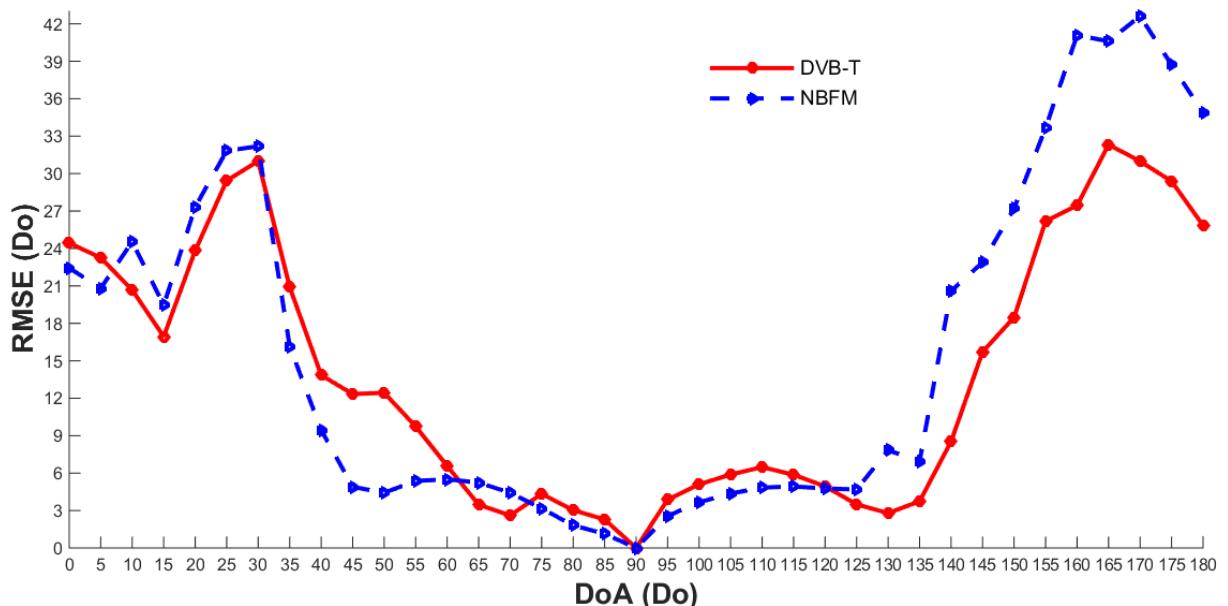
Tuy nhiên qua thực nghiệm sử dụng tín hiệu băng rộng DVB-T cho đồng bộ và xác định hướng sóng đến, nhận thấy nếu thu toàn bộ 8 MHz băng thông của DVB-T sử dụng



Hình 3.12. Phổ tín hiệu DVB-T và NBFM thu được

cho đồng bộ hệ BladeRF thì không thể thu được kết quả như mô phỏng. Vì vậy thay vì sử dụng toàn bộ dải băng rộng của tín hiệu DVB-T, tác giả khóa luận chỉ thu một phần của phổ DVB-T là 2 MHz, sau đó sử dụng thêm bộ lọc thông thấp 5 kHz như NBFM để lấy một phần nhỏ của tín hiệu DVB-T ban đầu, điều này khiến việc giải điều chế tín hiệu đồng thời như NBFM là không thể, tuy nhiên lại cho kết quả tốt khi xác định hướng sóng đến.

Kết quả thực nghiệm trên hình 3.13 khi thực nghiệm DOA tại một vị trí chỉ thay đổi tín hiệu bên phát, cho thấy dù ở khoảng 135° đến 180° sai số của DVB-T là nhỏ hơn NBFM nhưng nhìn chung thì trong khoảng góc quan trọng nhất là xung quanh 90° và hình dạng của đồ thị sai số của hai loại tín hiệu này vẫn là giống nhau và giữ được hình dạng như CRB.



Hình 3.13. RMSE: DOA với tín hiệu NBFM và DVB-T

Kết luận lại, sử dụng tín hiệu băng rộng cho hệ DOA sử dụng thuật toán MUSIC hoàn toàn khả thi, kết quả thu được vẫn tốt giống như tín hiệu băng hẹp NBFM, tuy nhiên để tối ưu hơn hệ thống và tận dụng hết ưu thế của điều chế OFDM trong tín hiệu DVB-T thì tác giả cần phải tiếp tục nghiên cứu và phát triển thêm.

KẾT LUẬN VÀ HƯỚNG NGHIÊN CỨU TIẾP THEO

1. Kết luận của khóa luận tốt nghiệp

Dựa trên các phân tích, mô phỏng, thực thi thuật toán đã triển khai ở các phần trên, chúng tôi có một số kết luận sau:

- Nếu chỉ sử dụng 2 BladeRF như trong khóa luận, việc ước lượng góc vẫn là khả thi khi ở trong khoảng từ 60° đến 135° với sai số dưới 6° . Tuy nhiên để tăng độ chính xác của thuật toán, việc tăng thêm số phần tử trong mảng thu là điều bắt buộc.

- Hệ thống có thể hoạt động ở các điều kiện kênh truyền khác nhau.

- Hệ thống vẫn hoạt động chính xác với tín hiệu băng rộng và cho ra kết quả tương đồng với tín hiệu băng hẹp.

Kết quả của hệ thống vẫn còn phụ thuộc vào loại nguồn tín hiệu dùng để đồng bộ, và điều kiện kênh truyền đặc biệt là nhiều đa đường. Đây cũng chính là hạn chế của thuật toán MUSIC, việc chỉ dựa trên sự khác biệt về pha của các tín hiệu đèn giúp hệ thống hoạt động linh hoạt với các loại điều chế, chuẩn tín hiệu khác nhau, tuy nhiên lại giảm đi sự chính xác khi tín hiệu có sự tương quan lớn gây ra bởi nhiều đa đường khi đặt hệ thống trong không gian hẹp.

2. Hướng nghiên cứu tiếp theo

Trong khóa luận, các quá trình thực nghiệm diễn ra trong thời gian chưa đủ dài để nhiệt độ làm ảnh hưởng đến hệ BladeRF thu, vì vậy cần tiếp tục nghiên cứu thêm về thời gian hoạt động trước khi cần đồng bộ lại hệ thống.

Việc chuyển đổi giữa trạng thái đồng bộ và ước lượng hướng sóng đến trong khóa luận còn thủ công, có thể thêm điều kiện để hệ thống tự động chuyển sang trạng thái DOA.

Do điều kiện phần cứng hiện tại hệ thu chỉ gồm 2 thiết bị BladeRF, tuy nhiên việc tăng thêm số phần tử mảng thu là cần thiết để tăng độ phân giải của hệ thống. Kéo theo đó là tăng được thêm số lượng nguồn sóng đến để phân tích việc có nhiều tín hiệu đến ảnh hưởng đến sự chính xác và ổn định của hệ thống.

Đồng bộ biên độ giữa tất cả các phần tử mảng thu giúp đẩy đỉnh phổ không gian hội tụ tại một đỉnh cao nhất qua việc thay đổi hệ số khuếch đại, xử lý tín hiệu bằng phần mềm,...

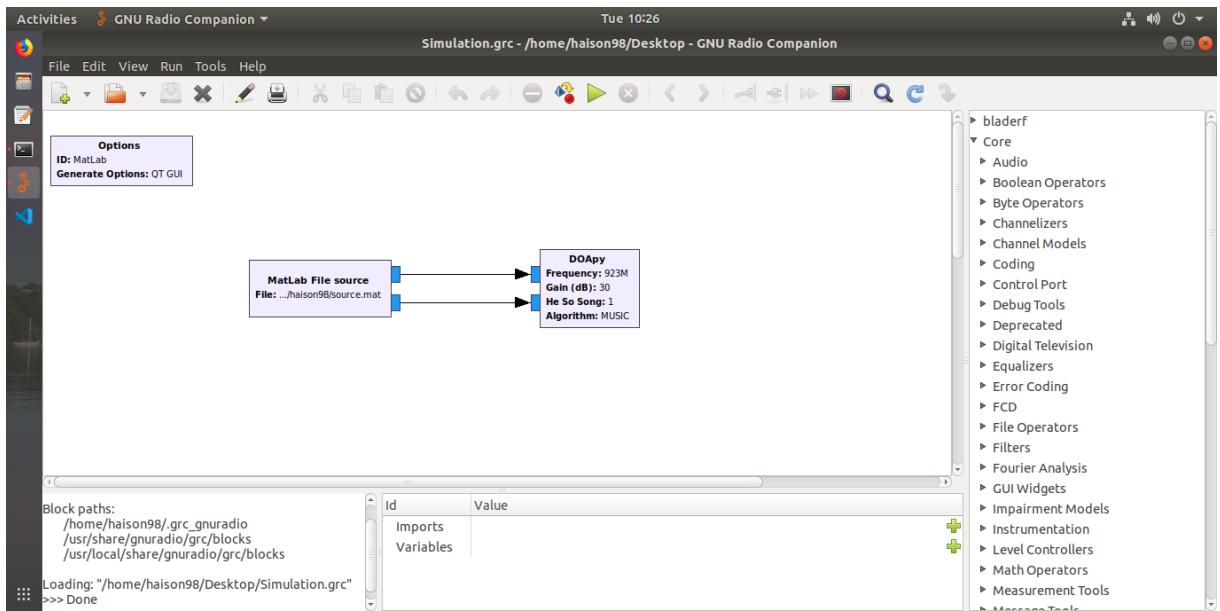
Cuối cùng do BladeRF hỗ trợ khả năng lập trình lại FPGA và có sẵn SPI Flash trong phần cứng, có thể nghiên cứu đưa toàn bộ hệ thống vào phần cứng thay vì sử dụng thêm máy tính để xử lý bên ngoài.

Tất cả khóa luận được cập nhật lên github dưới dạng mã nguồn mở:

https://github.com/DoHaiSon/gr-DoA_BlaeRF

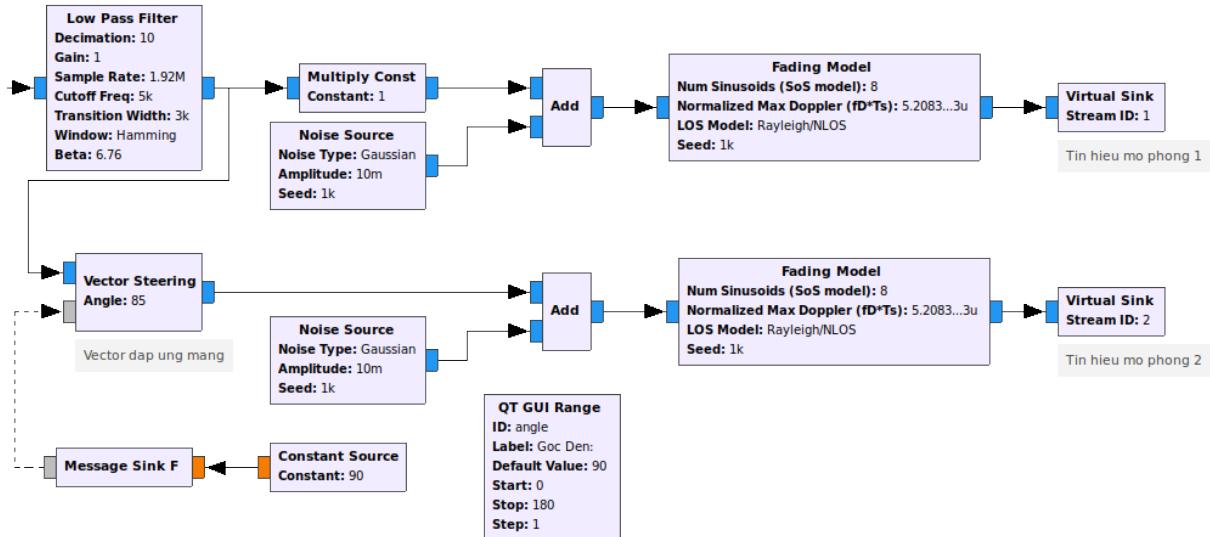
PHỤ LỤC

A). Flow-graph mô phỏng DOA với dữ liệu Matlab trên GNU Radio.

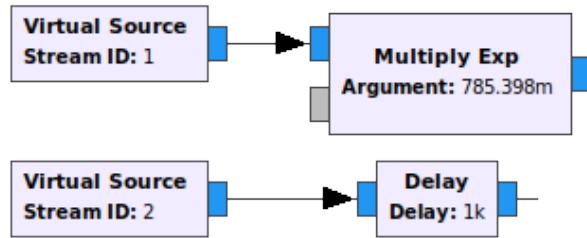


Hình A.1. Flow-graph ước lượng DOA với dữ liệu mô phỏng Matlab trên GNU Radio

B). Mô phỏng dữ liệu thu từ BladeRF, vừa chịu ảnh hưởng từ kênh truyền, vừa chịu ảnh hưởng do sai số phần cứng BladeRF gây ra.



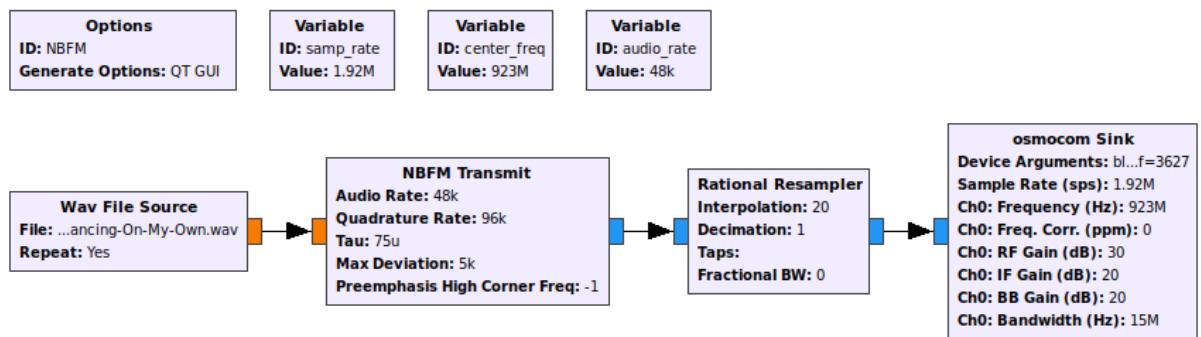
(a) Mô phỏng tạp âm và nhiễu đa đường



(b) Mô phỏng sample_{offset} và phase_{offset}

Hình B.1. Mô phỏng dữ liệu BladeRF qua kênh truyền

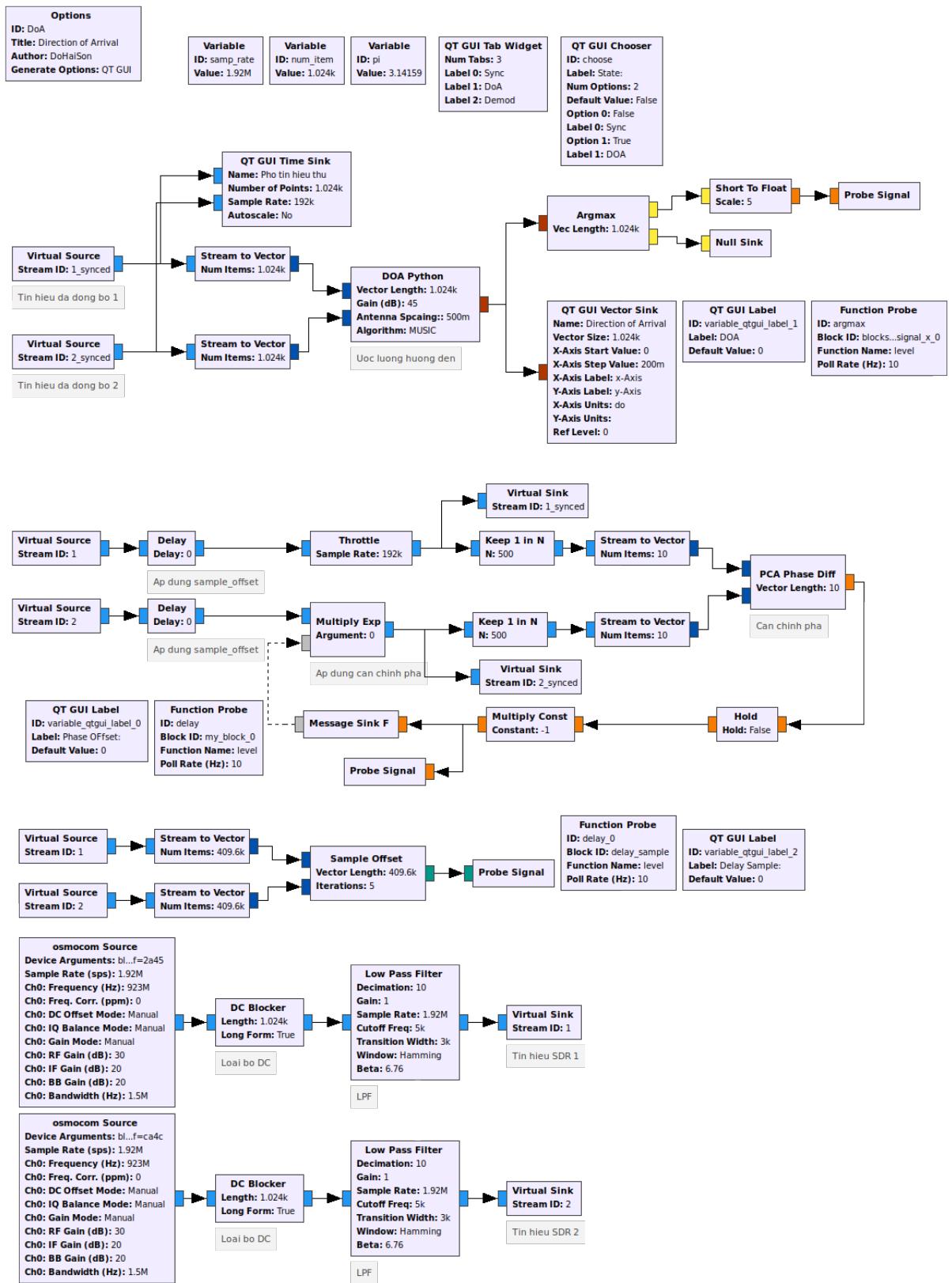
C). Flow-graph điều chế tín hiệu NBFM:



Hình C.1. Flow-graph truyền tín hiệu NBFM

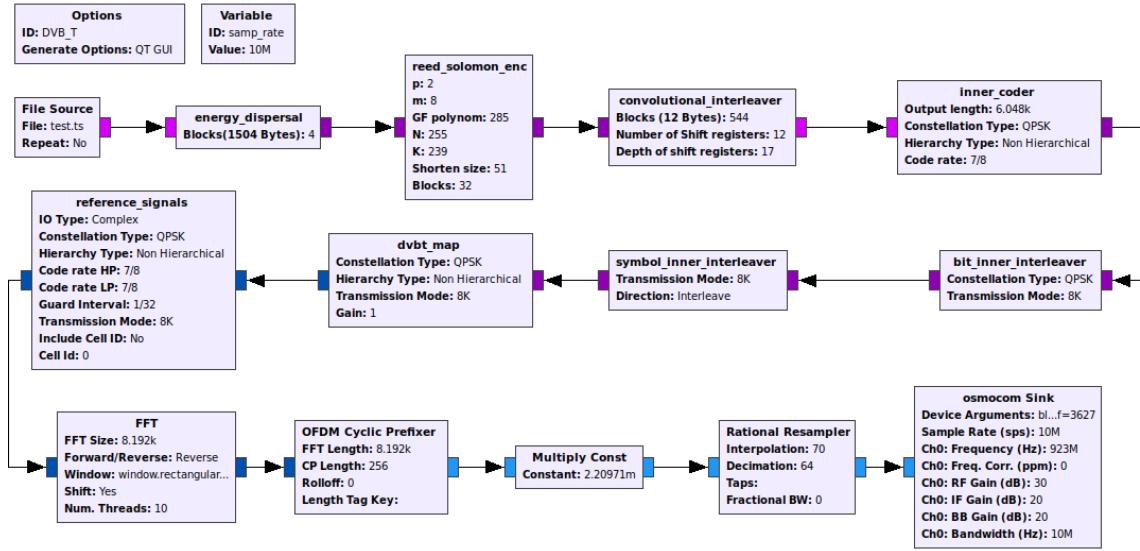
D). Flow-graph của hệ DOA trên BladeRF, các khối tương ứng với chức năng chính:

- Osmosdr Source: khối nguồn lấy dữ liệu trực tiếp từ các SDR, được phát triển dưới dạng mã nguồn mở, hỗ trợ rất nhiều loại SDR: RTL-SDR, USRP, BladeRF, HackRF, LimeSDR, ... Có sẵn hầu hết các thông số đầu vào như tần số thu, tốc độ lấy mẫu, băng thông, khuếch đại,... tất cả đều trực quan và dễ dàng chỉnh sửa.
- DC Blocker: loại bỏ thành phần một chiều trong tín hiệu.
- Low Pass Filter: bộ lọc thông thấp, dễ dàng chuyển đổi f_{cut} tương ứng với trạng thái đồng bộ hoặc ước lượng DOA của hệ thống.
- Virtual Sink/Source: các khối chuyển tiếp dữ liệu, giúp thu gọn và chia luồng dữ liệu để tiện xử lý.
- Sample_{offset}: ước lượng mẫu offset do trễ đầu vào.
- PCA Phase Diff: tính toán độ lệch pha giữa 2 tín hiệu cho bước đồng bộ SDR.
- Probe Signal/Fucion Probe: lưu giá trị đầu vào dưới dạng biến làm thông số cho các khối khác.
- DOA/DOA Python: khối ước lượng DOA được viết trên ngôn ngữ tương ứng C++/Python.
- QT GUI: các khối hiển thị giao diện người dùng.



Hình D.1. Flow-graph ước lượng DOA

E). Flow-graph điều chế tín hiệu truyền hình chuẩn DVB-T trên GNU Radio.



Hình E.1. Flow-graph truyền tín hiệu DVB-T

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Trần Thị Thúy Quỳnh , *Nghiên cứu nâng cao hiệu năng của hệ tìm phương sử dụng anten không tần pha trong môi trường các nguồn tín hiệu tương quan*, Luận án tiến sĩ, 2015, tr. 19–23.

Tiếng Anh

- [2] Andrew Feickert, “The Joint Tactical Radio System (JTRS) and the Army’s Future Combat System (FCS): Issues for Congress”, *Congressional Research Service: Report*, 2005, pp. 1.
- [3] B. Bloessl, M. Segata, C. Sommer and F. Dressler, “An IEEE802.11a/g/p OFDM receiver for GNU radio”, *SRIF 2013 - Proceedings of the 2nd, 2013 ACM SIGCOMM Workshop on Software Radio Implementation Forum*, 2013, pp. 9–15.
- [4] BogdanDIA , “DVB-T implementation in gnuradio”, 2015.
[Online]. Available: <https://github.com/BogdanDIA/gr-dvbt>
- [5] C. Bergstrom, S. Chuprun, S. Gifford, and G. Maalouli , “Software defined radio (SDR) special military applications”, *MILCOM 2002. Proceedings*, vol. 1, 2002, pp. 383–388.
- [6] E. Bellini and A. Tosi, “A directive system of wireless telegraphy”, *Proceedings of the Physical Society of London*, vol. 21, no. 1, 1907, pp. 305–328.
- [7] I. Gomez-Miguel, A. Garcia-Saavedra, Paul D. Sutton, P. Serrano, C. Cano and Douglas J. Leith, “srsLTE: An open-source platform for LTE evolution and experimentation”, *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2016, pp. 25–32.
- [8] I. Ziskind and M. Wax, “Maximum Likelihood Localization of Multiple Sources by Alternating Projection”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 10, 1988, pp. 1553–1560.
- [9] J. Capon, “High-resolution frequency-wavenumber spectrum analysis”, *Proceedings of the IEEE*, vol. 57, no. 8, 1969, pp. 1408–1418.
- [10] Joseph C. Liberti and Theodore S. Rappaport, *Smart antennas for wireless communications: IS-95 and third generation CDMA applications*, Prentice Hall PTR, 1999, pp. 134–134.
- [11] Kenneth van Rijsbergen, “The effectiveness of a homemade IMSI catcher build with YateBTS and a BladeRF”, *Semanticscholar*, 2016.

- [12] KerberosSDR: “Demo software for direction finding and passive radar”, 2019.
[Online]. Available: <https://github.com/rtlsdrblog/kerberossdr>
- [13] M. Nentwig, “Delay estimation by FFT”, *DSPRelated*, 2016.
- [14] Nuand/bladeRF, “DC offset and IQ Imbalance Correction”.
[Online]. Available: <https://github.com/Nuand/bladeRF/wiki/DC-offset-and-IQ-Imbalance-Correction>
- [15] Nuand/bladeRF: “kalibrate-bladeRF”, 2010.
[Online]. Available: <https://github.com/Nuand/kalibrate-bladeRF>
- [16] P. Kavitha, K. Meera Mohan, R. Surya, R. Gandhiraj and K.P. Soman, “Implementation of CDMA in GNU radio”, *Procedia Computer Science*, vol. 46, 2015, pp. 981–988.
- [17] P. Krysik, “Multi-channel receiver with use of RTL-SDR dongles”, 2016.
[Online]. Available: <https://github.com/ptrkrysik/multi-rtl>
- [18] P. Stoica and A. Nehorai, “MUSIC, Maximum Likelihood, and Cramer-Rao Bound”, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, 1989.
- [19] PyBOMBS: “Python Build Overlay Managed Bundle System”.
[Online]. Available: <https://github.com/gnuradio/pybombs>
- [20] Ronald N. Bracewell, *The Fourier Transform and Its Applications*, McGraw-Hill, 1978, pp. 46 and 243.
- [21] R. I. Lackey and D. W. Upmal, “Speakeasy: The Military Software Radio”, *IEEE Communications Magazine*, vol. 33, no. 5, 1995, pp. 56–61.
- [22] R. Keen, *Direction and Position Finding by Wireless*, 1922.
- [23] R. Schmidt, “Multiple emitter location and signal parameter estimation”, *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, 1986, pp. 276–280.
- [24] S. Whiting, D. Sorensen, T. K. Moon and J. H. Gunther, “Time and frequency corrections in a distributed radio network”, *2017 51st Asilomar Conference on Signals, Systems, and Computers*, 2017, pp. 1123–1126.
- [25] Srikanth Pagadarai, Travis Collins and Alexander M. Wyglinski, “Phase Synchronization Capability of TwinRX Daughterboards and DoA Estimation”, *Ettus Research*, 2016, pp. 1–12.