

ÁP DỤNG PHƯƠNG PHÁP MẠNG NƠ-RON NHÂN TẠO ĐỂ THIẾT KẾ BỘ CÂN BẰNG HỆ THỐNG MIMO-VLC

LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

Nguyễn Tất Đạt – 1610658
Võ Hoàng Thành – 1613206
Giảng viên hướng dẫn
TS. Phạm Quang Thái



ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN – ĐIỆN TỬ, BỘ MÔN VIỄN THÔNG

Số: _____/BKĐT

Khoa: **Điện – Điện tử**

Bộ môn: **Viễn thông**

NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP

- Họ và tên: Nguyễn Tất Đạt, MSSV: 1610658
Võ Hoàng Thành, MSSV: 1613206
- Ngành: Điện – Điện tử, Chuyên ngành: Kỹ thuật Điện tử – Truyền thông
- Đề tài: Áp dụng phương pháp mạng nơ-ron nhân tạo để thiết kế bộ cân bằng hệ thống MIMO-VLC
- Nhiệm vụ:
 - Nhiệm vụ 1: Thiết lập hệ thống MIMO và lấy dữ liệu
 - Nhiệm vụ 2: Xử lý dữ liệu thu được và đưa vào mạng nơ-ron nhân tạo để xử lý
 - Nhiệm vụ 3: So sánh các kết quả thu được và nhận xét về tính hiệu quả của mạng nơ-ron nhân tạo
- Ngày giao nhiệm vụ: 24-9-2020
- Ngày hoàn thành nhiệm vụ: 11-1-2021
- Họ và tên người hướng dẫn: TS. Phạm Quang Thái
BM Viễn Thông, Khoa Điện – Điện Tử

Phần hướng dẫn:
100%

Nội dung và yêu cầu LVTN đã được thông qua Bộ Môn.

Tp. HCM, Ngày __ tháng __ năm 20__
CHỦ NHIỆM BỘ MÔN

Tp. HCM, Ngày __ tháng __ năm 20__
NGƯỜI HƯỚNG DẪN CHÍNH

PGS. TS. Hà Hoàng Kha

TS. Phạm Quang Thái

PHẦN DÀNH CHO KHOA, BỘ MÔN:

Người duyệt (chấm sơ bộ): _____

Đơn vị: _____

Ngày bảo vệ: _____

Điểm tổng kết: _____

Nơi lưu trữ luận văn: _____

LỜI CẢM ƠN

Đầu tiên, chúng em xin được gửi lời cảm ơn chân thành đến thầy hướng dẫn luận văn của mình TS.Phạm Quang Thái. Trong quá trình thực hiện luận văn, thầy là người đã nhiệt tình hỗ trợ, chỉ dẫn giúp chúng em củng cố kiến thức đồng thời chỉ ra những vấn đề cốt lõi giúp chúng em có định hướng đúng đắn để hoàn thành luận văn.

Tiếp đến, chúng em xin được gửi lời cảm ơn đến các thầy cô đã và đang dạy tại trường Đại Học Bách Khoa Tp.HCM và đặc biệt là các thầy cô ở Bộ Môn Viễn Thông đã giúp chúng em xây dựng được kiến thức nền tảng, là cơ sở để chúng tôi thực hiện được luận văn này.

Cuối cùng chúng em xin gửi lời cảm ơn đến gia đình, bạn bè, đồng nghiệp đã hết sức giúp đỡ, quan tâm, động viên để chúng tôi có điều kiện thuận lợi để thực hiện luận văn này.

Xin chân thành cảm ơn!

Tp. HCM, Ngày 8 tháng 1 năm 2021

Nguyễn Tất Đạt, Võ Hoàng Thành

LỜI CAM ĐOAN

Tôi tên: Nguyễn Tất Đạt (MSSV: 1610658), Võ Hoàng Thành (MSSV: 1613206), là sinh viên chuyên ngành Kỹ thuật Điện tử - Truyền thông, tại Trường Đại học Bách Khoa, Đại học Quốc gia thành phố Hồ Chí Minh. Chúng tôi xin cam đoan những nội dung sau đều là sự thật:

- Công trình nghiên cứu này hoàn toàn do chính chúng tôi thực hiện;
- Các tài liệu và trích dẫn trong luận văn này được tham khảo từ các nguồn thực tế, có uy tín và độ chính xác cao;
- Các số liệu và kết quả của công trình này được chúng tôi tự thực hiện một cách độc lập và trung thực.

Tp. HCM, Ngày 8 tháng 1 năm 2021

Nguyễn Tất Đạt, Võ Hoàng Thành

TÓM TẮT NỘI DUNG

Bài toán nghiên cứu:

Sử dụng kết quả đo đạc thực tế của hệ thống MIMO-VLC sử dụng đèn OLED để xây dựng mạng nơ-ron xác suất nhằm khôi phục tín hiệu sau khi đi qua kênh truyền.

Phương pháp tiếp cận:

Dùng phần mềm Matlab để tạo tín hiệu NRZ điều khiển 2 đèn OLED trong phòng thí nghiệm. Sau khi đi qua hệ thống thực tế tín hiệu thu được sẽ được xử lý qua Matlab để phục vụ cho việc khảo sát mạng nơ-ron xác suất. Thông số tỉ lệ lỗi bit sẽ được đem ra để đánh giá khả năng của mạng nơ-ron trong hệ thống MIMO-VLC.

Kết quả:

Kết luận:

ABSTRACT

Research problem:

Using neural network to enhance the demodulation of MIMO-VLC system .

Research methods:

Using Matlab to create NRZ signal to control 2 OLED in labortary. After transmitting through a real system, received signal is analysed using Matlab to put in a neural network. Bit error rate will be compared to evaluate the efficiency of neural network in MIMO-VLC system.

Result:**Conclusion:**

MỤC LỤC

1	Giới thiệu	1
1.1	Đặt vấn đề	1
1.2	Phạm vi và phương pháp nghiên cứu	1
1.3	Các đóng góp của luận văn	2
2	Cơ sở lý thuyết	3
2.1	Line coding	3
2.1.1	None return to Zero	4
2.1.2	Return to Zero (RZ)	7
2.2	Hệ thống MIMO-VLC	9
2.2.1	Tổng quan về VLC	9
2.2.2	Phân loại kênh truyền	11
2.3	Neural Network	14
2.3.1	Mạng thần kinh học	14
2.3.2	Mạng thần kinh nhân tạo	15
2.3.3	Probabilistic neural network (PNN)	16
2.4	Kết luận chương	21
3	Kết quả nghiên cứu	23
3.1	Phương pháp tiếp cận	23
3.1.1	Sơ đồ khối hệ thống	23
3.1.2	Thí nghiệm 1	24
3.1.3	Thí nghiệm 2	25
4	Kết luận	27
4.1	Tóm tắt và kết luận chung	27
4.2	Hướng phát triển	27
A	Code chương trình	31
A.1	Code tạo tín hiệu NRZ	31
A.2	Code lấy tín hiệu từ oscilloscope	35
A.3	Code trích tín hiệu phát theo tín hiệu thu	36
A.4	Code tính ước lượng SNR, BER	37
A.5	Code tính BER theo PNN	38

DANH SÁCH HÌNH VẼ

Hình 2.1	Các loại mã đường truyền	3
Hình 2.2	Dạng tín hiệu mã Uni-Polar NRZ	5
Hình 2.3	Phổ công suất của tín hiệu NRZ đơn cực	5
Hình 2.4	Dạng tín hiệu mã NRZ	6
Hình 2.5	Phổ của tín hiệu Polar-NRZ	6
Hình 2.6	Dạng tín hiệu Bi-polar NRZ	7
Hình 2.7	Dạng phổ của tín hiệu Bi-polar NRZ	7
Hình 2.8	Dạng tín hiệu của Uni-polar RZ	7
Hình 2.9	Dạng phổ của tín hiệu của Uni-polar RZ	8
Hình 2.10	Dạng tín hiệu của Polar RZ	8
Hình 2.11	Dạng phổ của tín hiệu của Polar RZ	8
Hình 2.12	Dạng phổ của tín hiệu của Bi-polar RZ	9
Hình 2.13	Dạng phổ của tín hiệu của Bi-polar RZ	9
Hình 2.14	Biểu đồ tăng trưởng lưu lượng và số lượng thiết bị	10
Hình 2.15	Vùng ánh sáng khả kiến được sử dụng cho VLC	11
Hình 2.16	Công nghệ VLC trong tương lai	11
Hình 2.17	Mô hình kênh truyền trong hệ thống VLC	12
Hình 2.18	Kênh truyền hệ thống VLC	13
Hình 2.19	Mô hình MIMO trong phòng	14
Hình 2.20	Mô hình của hệ thống giao tiếp ánh sáng MIMO	15
Hình 2.21	Sơ đồ 1 neural sinh học	15
Hình 2.22	Cấu trúc của 1 perceptron	16
Hình 2.23	Cấu trúc của 1 mạng ANN	17
Hình 2.24	Sơ đồ cấu trúc một PNN	17
Hình 2.25	1 chuỗi gồm 500 bit ngẫu nhiên 0 và 1 được tạo từ code Matlab	21
Hình 2.26	Hình ảnh của chuỗi được đưa vào máy phát sóng	21
Hình 2.27	Hình ảnh của chuỗi bit thu được	22
Hình 2.28	Chuỗi bit thu được gồm 500 symbols và 1 symbols được biểu diễn bởi 3 samples	22
Hình 3.1	Sơ đồ khối hệ thống MIMO-VLC	23
Hình 3.2	Đáp ứng tần số của OLED	24
Hình 3.3	Bộ thu quang APD410A của Thorlab	25

DANH SÁCH BẢNG

DANH SÁCH TỪ VIẾT TẮT

ANN Artificial Neural Networks

BER Bit error rate

FOV Field Of View

MIMO Multiple-input Multiple-output

NRZ None Return to Zero

OLED organic light-emitting diode

OWC Optical wireless communication

PDF Probability Distribution Function

PNN Probabilistic neural network

RZ Return to Zero

SNR Signal to Noise Ratio

VLC Visible Light Communication

Chương 1. GIỚI THIỆU

1.1 Đặt vấn đề

Trên thế giới ngày nay, truyền tín hiệu bằng ánh sáng khả kiến Visible Light Communication (VLC) đang rất được quan tâm bởi các nhà nghiên cứu cũng như kỹ sư bởi tính tiện dụng của nó. Tuy chưa được áp dụng rộng rãi, đại trà như công nghệ này rất có tiềm năng trong tương lai. Hàng trăm bài báo, video bạn có thể kiếm được trên internet nói về chủ đề này. Là những sinh viên chúng em cũng muốn tiếp cận những kiến thức mới, công nghệ mới. Việc khảo sát về hoạt động của hệ thống VLC trong phòng thí nghiệm cũng như mô phỏng đã được các anh chị khóa trước trình bày khá rõ ràng, nhưng hệ thống chỉ có 1 đèn và 1 bộ thu. Trong luận văn này, chúng em sẽ làm thí nghiệm về hệ thống VLC sử dụng 2 đèn và 1 bộ thu, mô phỏng gần giống với thực tế. Ngoài ra, việc sử dụng mạng nơ-ron nhân tạo để thiết kế bộ cân bằng cho hệ thống VLC cũng được nghiên cứu khá nhiều nên tại em cũng sử dụng mạng nơ-ron xác suất để khảo sát xem nó có hoạt động tốt trong hệ thống MIMO-VLC không.

Câu hỏi nghiên cứu đặt ra của luận văn là:

- Việc sử dụng mạng nơ-ron xác suất có hiệu quả trong việc làm bộ cân bằng cho hệ thống MIMO-VLC.
- Tốc độ bit cần bao nhiêu thì mạng nơ-ron xác suất sẽ cho ra tỉ lệ lỗi bit không còn đạt ngưỡng.
- Khoảng cách tối đa là bao nhiêu thì mạng nơ-ron sẽ không còn cân bằng chính xác với điều kiện trong phòng thí nghiệm.

Trong chương 2, cơ sở lý thuyết về MIMO-VLC, line coding và neural network sẽ được trình bày. Trong chương 3, các kết quả mô phỏng sẽ được so sánh và phân tích. Cuối cùng, chương 4 đưa ra kết luận chung.

1.2 Phạm vi và phương pháp nghiên cứu

- Sinh viên thực hiện đo đạc trong phòng thí nghiệm ở trường, các thiết bị trong phòng thí nghiệm do thầy hướng dẫn cung cấp. Do kích thước phòng có hạn nên chúng em chỉ khảo sát ở khoảng cách lớn nhất là 2m.
- Khảo sát sự thay đổi của tỉ lệ lỗi bit theo tốc độ bit và sự thay đổi của tỉ lệ lỗi bit theo khoảng cách.
- Tín hiệu thí nghiệm là tín hiệu NRZ được tạo từ code Matlab, sẽ được truyền đi qua hệ thống thực tế.

- Mạng nơ-ron nhân tạo được sử dụng là mạng nơ-ron xác suất Probabilistic neural network (PNN), mạng này có sẵn trong Matlab nên rất thuận tiện cho việc khảo sát.

1.3 Các đóng góp của luận văn

Luận văn này có các đóng góp như sau:

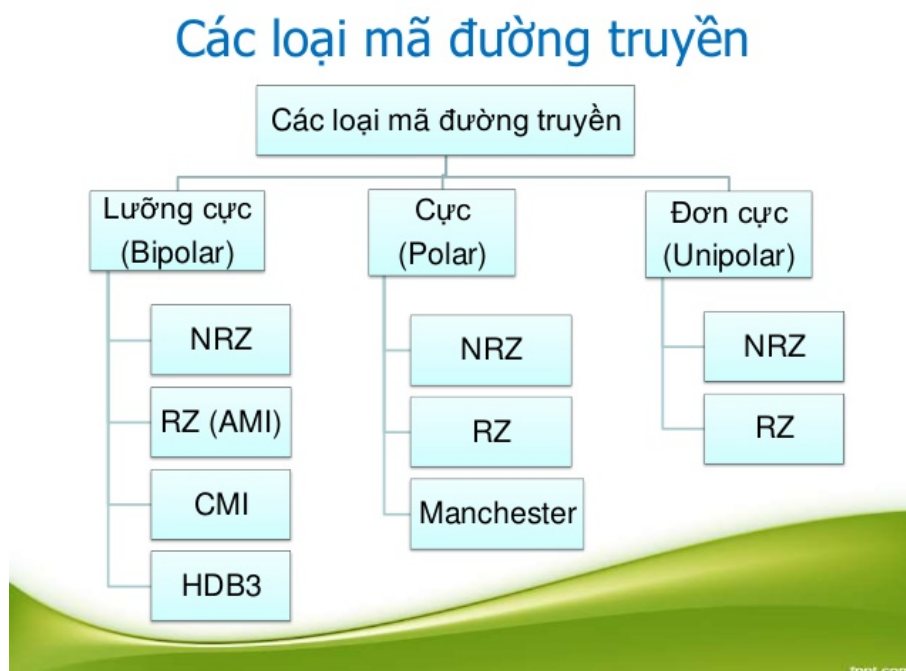
- Hiện thực hoá bộ cân bằng tín hiệu dùng phương pháp mạng nơ-ron nhân tạo, cụ thể là PNN dùng phần mềm Matlab.
- Đưa ra được các số liệu về tỉ lệ lỗi bit sát với lý thuyết. Các số liệu này cũng có thể đem khảo sát ở thực tế.
- So sánh và phân tích Bit error rate (BER) trên các tốc độ bit và khoảng cách.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1 Line coding

Trong viễn thông, mã đường truyền hay còn được gọi là điều chế số dải nền là một mã được chọn để sử dụng cho việc truyền nhận của tín hiệu trong hệ thống, mã đường truyền thường được sử dụng trong truyền thông tin số. Hay nói cách khác, nó dùng để ánh xạ chuỗi dữ liệu của chúng ta thành một dạng tín hiệu mà có thể truyền nhận được.

Mã đường truyền biểu diễn tín hiệu số (chuỗi nhị phân) được truyền đi, bằng một dạng sóng được đặc trưng bởi một đặc tính nào đó của kênh truyền vật lý, hoặc thiết bị thu như là điện áp, dòng điện, photon. Có ba loại chính trong mã hóa đường truyền, đó là Unipolar, Bi-Polar, Polar.



Hình 2.1: Các loại mã đường truyền

Mục đích của mã hoá đường truyền là [4]:

- Tạo ra phổ tín hiệu sao cho phù hợp với kênh truyền.
- Tạo khả năng tách tín hiệu đồng bộ ở bộ thu.
- Tăng tốc độ truyền dẫn.

- Tạo khả năng phát hiện và sửa lỗi.

Từ những mục đích đó, một số yếu tố cũng được đề ra để giúp người thiết kế có thể chọn lựa mã đường truyền phù hợp cho hệ thống, khi chọn lựa mã đường truyền, ta phải lưu tâm đến những yếu tố [4]:

- Thành phần một chiều DC, băng thông của hệ thống.
- Tỷ lệ lỗi, khả năng tự phát hiện lỗi.
- Đơn giản trong việc mã hóa và giải mã.
- Khả năng khôi phục xung clock.

2.1.1 None return to Zero

Trong hệ thống viễn thông, NRZ là mã đường truyền nhị phân, mà trong đó các bit 1 được đại diện bởi một giá trị, thông thường là điện áp dương, trong khi đó các bit 0 được đại diện với một giá trị khác, thông thường là điện áp âm. Bản thân NRZ vốn không có khả năng tự đồng bộ xung clock, nên một số kỹ thuật đồng bộ khác phải được đưa vào để tránh sự trượt bit. Với một tốc độ bit nhất định, NRZ chỉ cần một nửa băng thông dải nền so với mã manchester. Có ba loại mã hóa NRZ: bi-polar, polar, unipolar.

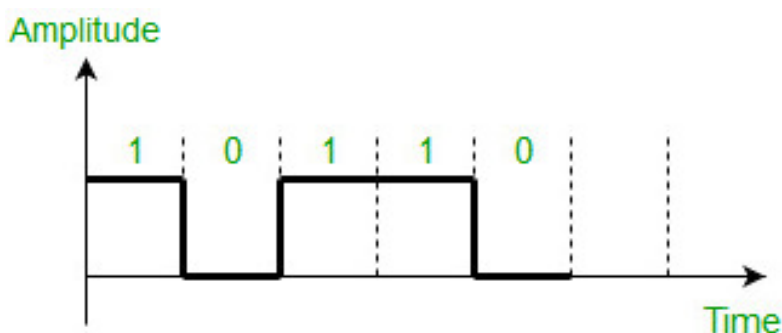
Uni-polar NRZ

Trong hệ thống viễn thông, None Return to Zero (NRZ) là mã đường truyền nhị phân, mà trong đó các bit 1 được đại diện bởi một giá trị, thông thường là điện áp dương, trong khi đó các bit 0 được đại diện với một giá trị khác, thông thường là điện áp âm. Bản thân NRZ vốn không có khả năng tự đồng bộ xung clock, nên một số kỹ thuật đồng bộ khác phải được đưa vào để tránh sự trượt bit. Với một tốc độ bit nhất định, NRZ chỉ cần một nửa băng thông dải nền so với mã manchester. Có ba loại mã hóa NRZ: bi-polar, polar, unipolar.

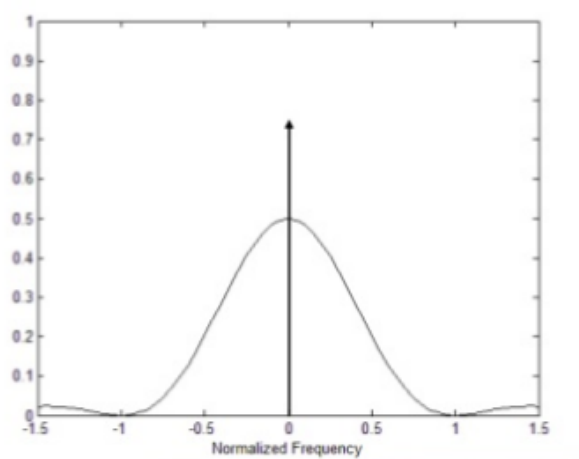
Trong NRZ đơn cực (Unipolar NRZ), bit 1 được đại diện bởi mức điện áp DC trên đường truyền, trong khi đó bit 0 là không có điện áp, hay còn gọi là mức điện áp 0V hoặc là đất. Vì lý do này, mã uni-polar NRZ còn được gọi là mã tắt mở (On-Off keying). Một điểm độc nhất của tín hiệu NRZ đơn cực chính là việc tồn tại mức DC trong quá trình truyền nhận, do đó phổ của tín hiệu tại tần số zero sẽ khác không. Việc này gây ra hai vấn đề chính: thứ nhất, công suất DC được truyền đi dẫn đến sự tiêu hao công suất hơn nhiều so với các phương pháp giải mã mà không có thành phần DC, và thứ hai, việc tồn tại thành phần DC buộc đường truyền phải được DC coupling [4].

Uni-polar NRZ

- Ưu điểm: Đơn giản, tiêu tốn ít băng thông.
- Nhược điểm: Có thành phần DC, không có khả năng sửa lỗi, khó khôi phục xung clock do không có hài tại tần số xung clock (f) nên khó đồng bộ, tồn tại chuỗi bit 0 dài trong dữ liệu làm mất tính đồng bộ.



Hình 2.2: Dạng tín hiệu mã Uni-Polar NRZ



Hình 2.3: Phổ công suất của tín hiệu NRZ đơn cực

Polar NRZ

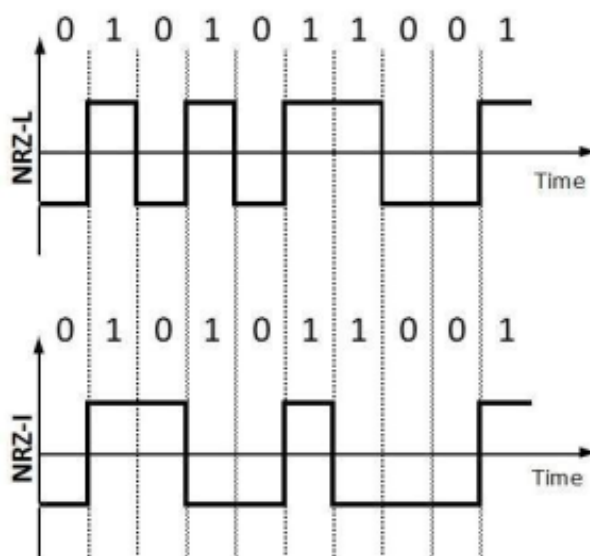
Trong mã hóa NRZ lưỡng cực, bit 1 được đại diện bởi điện áp dương, bit 0 được đại diện bởi điện áp âm. Trong mã này, mức điện áp tín hiệu sẽ được thay đổi từ dương sang âm tại cạnh xuống của chu kỳ xung clock trước. Một ứng dụng phổ biến của mã này chính là chuẩn giao tiếp nối tiếp RS-232, trong đó mức 1 được đặc trưng bởi điện áp trong khoảng -12V đến -5V và mức 0 được đặc trưng bởi điện áp trong khoảng 5V đến 12V. 2.4 mô tả dạng tín hiệu của mã hóa NRZ.

Polar NRZ còn thường được sử dụng dưới hai dạng mã hóa chính, đó chính là NRZ-I và NRZ-L như được mô tả trong hình trên. Trong đó, NRZ-L đơn giản hơn, nhưng cần phân biệt cực tính của tín hiệu, trong khi đó NRZ-I đáng tin cậy hơn, trong môi trường truyền có nhiều tạp âm, phát hiện sự chuyển mức tín hiệu dễ dàng hơn là so sánh tín hiệu với một giá trị ngưỡng như trong NRZ-L.

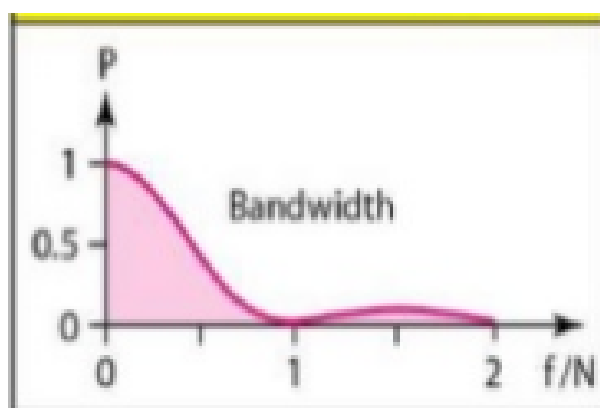
Nhận thấy phổ của tín hiệu NRZ trong trường hợp này giống như trong trường hợp uni-polar NRZ, do thực chất hai trường hợp chỉ là kết quả quá trình DC-offset của nhau.

Nhận xét về mã polar-NRZ:

- Ưu điểm: Đơn giản, dễ thiết kế, ít tiêu tốn băng thông



Hình 2.4: Dạng tín hiệu mã NRZ



Hình 2.5: Phổ của tín hiệu Polar-NRZ

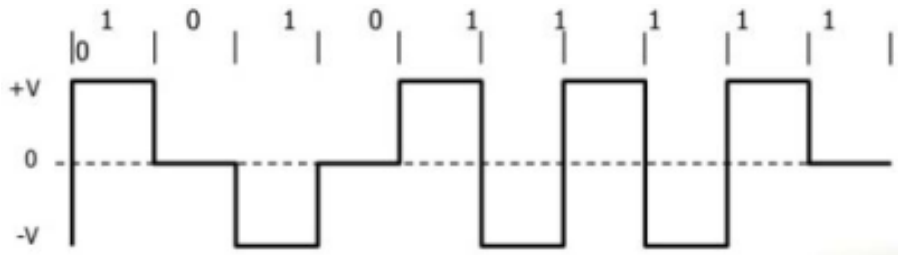
- Nhược điểm: Có thành phần DC, không có khả năng sửa lỗi, khó khôi phục xung clock do không có hài ở tần số f , ít được sử dụng cho việc truyền tín hiệu đi xa.

Bi-polar NRZ

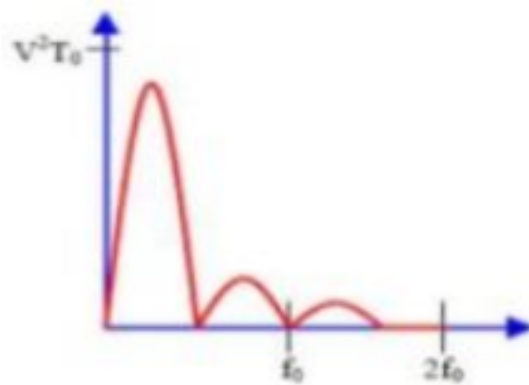
Trong tín hiệu Bi-polar NRZ, bit 0 được đặc trưng bởi mức điện áp 0V, trong khi đó mức 1 được đặc trưng bởi mức điện áp dương hoặc âm, luân phiên nhau.

Nhận xét về mã Bi-polar NRZ:

- Ưu điểm: không có thành phần DC, tiêu tốn ít băng thông hơn so với Unipolar và polar NRZ, có khả năng phát hiện lỗi.
- Nhược điểm: không có thành phần xung clock để đồng bộ hóa.



Hình 2.6: Dạng tín hiệu Bi-polar NRZ



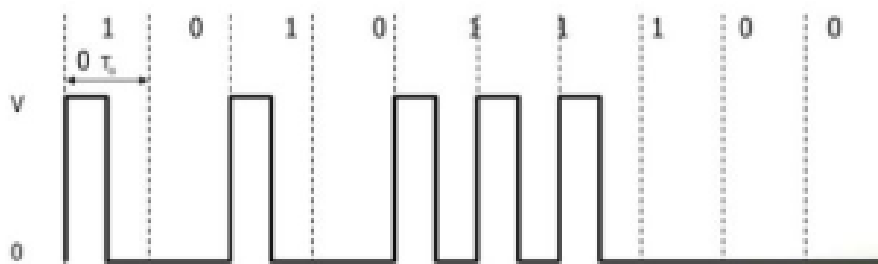
Hình 2.7: Dạng phổ của tín hiệu Bi-polar NRZ

2.1.2 Return to Zero (RZ)

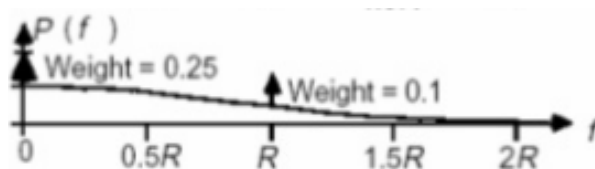
Return to Zero (RZ) là tên gọi của loại mã hóa trong hệ thống viễn thông, mà trong đó mức tín hiệu rơi về vị trí 0V sau mỗi chu kỳ xung. Khác với mã NRZ, RZ là tín hiệu có thể tự đồng bộ (self-clocking), do đó không cần một tín hiệu xung clock riêng biệt gửi cùng với dữ liệu, nhưng bên cạnh đó phải sử dụng gấp đôi băng thông so với cùng một tốc độ dữ liệu trong mã NRZ.

Uni-polar RZ

Cũng giống như uni-polar NRZ, tuy nhiên độ rộng xung chỉ bằng một nửa so với chu kỳ xung. Trong mã này, bit 1 được đại diện bởi dạng sóng có mức điện áp chuyển từ cao sang thấp trong một chu kỳ xung, và bit 0 được đại diện bởi trạng thái nghỉ (mức thấp), dạng tín hiệu được mô tả như 2.8.



Hình 2.8: Dạng tín hiệu của Uni-polar RZ



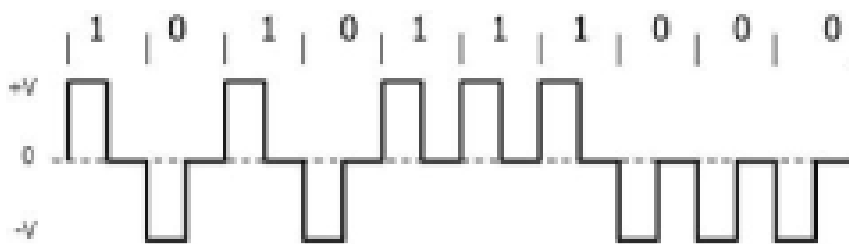
Hình 2.9: Dạng phổ của tín hiệu của Uni-polar RZ

Nhận xét về mã Uni-polar RZ:

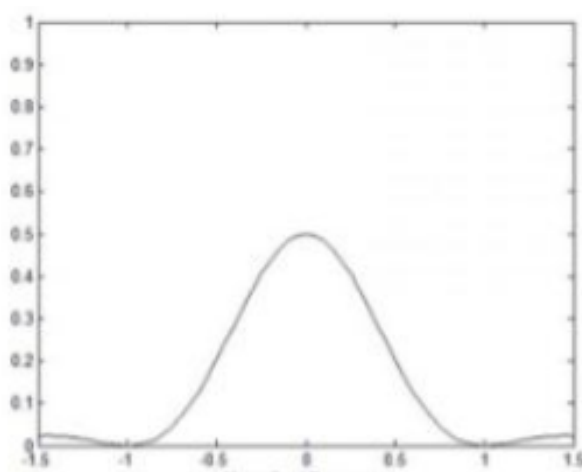
- Ưu điểm: đơn giản, xuất hiện hài tần số f , có thể dùng để khôi phục xung clock.
- Nhược điểm: tồn tại thành phần DC, không có khả năng sửa lỗi khi xuất hiện nhiễu, băng thông sử dụng gấp 2 lần so với unipolar NRZ.

Polar RZ

Trong mã hóa polar RZ, bit 1 được đặc trưng bởi dạng sóng có mức điện áp chuyển từ $+V$ sang 0 trong một chu kỳ xung, và bit 0 được đặc trưng bởi dạng sóng có mức điện áp chuyển từ $-V$ sang 0 trong một chu kỳ xung, dạng sóng này được mô tả trong 2.10 :



Hình 2.10: Dạng tín hiệu của Polar RZ



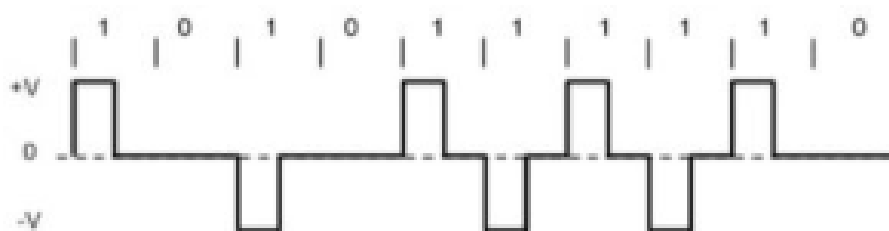
Hình 2.11: Dạng phổ của tín hiệu của Polar RZ

Nhận xét về mã Polar-RZ:

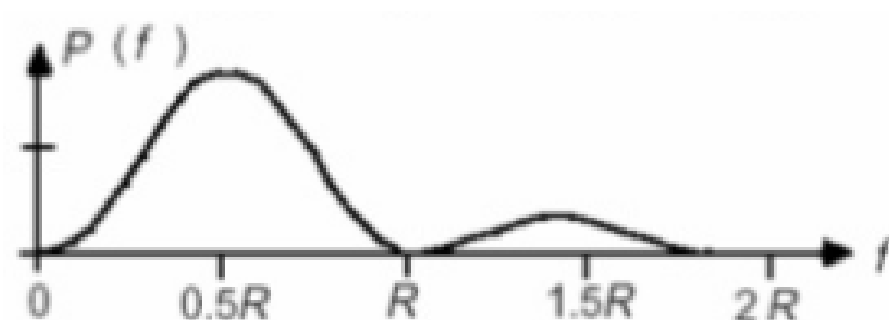
- Ưu điểm: đơn giản, dễ thực hiện.
- Nhược điểm: có thành phần DC, không có khả năng sửa lỗi, không có thành phần clocking để đồng bộ hoá tuy nhiên xung clock có thể được khôi phục bằng cách chắn chỉnh các tín hiệu nhận được, chiếm gấp đôi băng thông so với polar NRZ.

Bi-polar RZ

Trong tín hiệu Bi-polar RZ, bit 0 đặc trưng bởi điện áp $0V$, mức 1 được luân phiên chuyển từ điện áp $\pm V$ xuống 0 trong một chu kỳ xung.



Hình 2.12: Dạng phổ của tín hiệu của Bi-polar RZ



Hình 2.13: Dạng phổ của tín hiệu của Bi-polar RZ

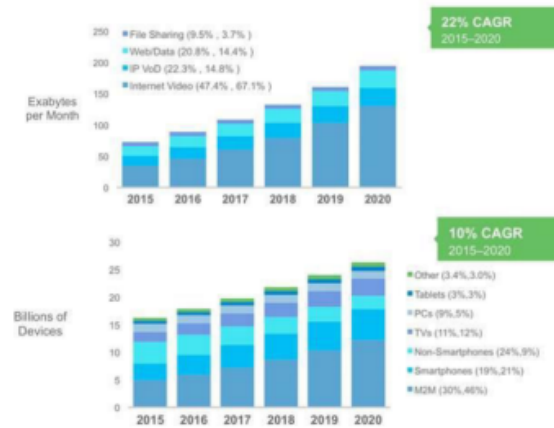
Nhận xét về mã hoá Bi-polar RZ:

- Ưu điểm: không có thành phần DC, tiêu tốn ít băng thông hơn so với Unipolar và polar RZ, có khả năng phát hiện lỗi, clock có thể được tách ra từ cách chắn chỉnh các tín hiệu thu được.
- Nhược điểm: chưa khắc phục được tình trạng dữ liệu có nhiều bit 0 liên tiếp.

2.2 Hệ thống MIMO-VLC

2.2.1 Tổng quan về VLC

Ngày nay kết nối internet đã trở thành nhu cầu cơ bản của con người. Tuy nhiên, không dừng lại ở việc truy cập internet, mà nhu cầu về tốc độ cũng không ngừng tăng cao.



Hình 2.14: Biểu đồ tăng trưởng lưu lượng và số lượng thiết bị

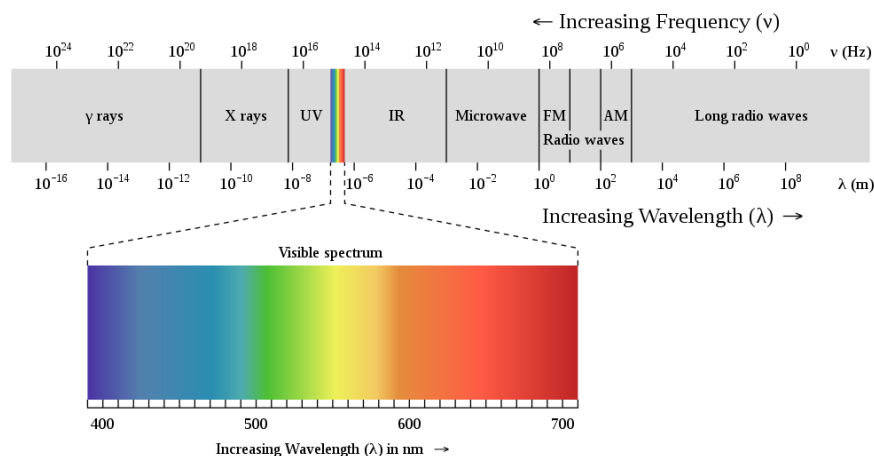
Theo dự báo chỉ số tăng trưởng mạng (VNI) thường niên của cisco lần thứ 10, lưu lượng cũng như số lượng thiết bị truy cập internet tăng với tốc độ chóng mặt trong những năm qua, thậm chí còn được cisco dự đoán sẽ tăng với tốc độ cao hơn trong những năm tới. Một số lượng thiết bị và nhu cầu lưu lượng khổng lồ như vậy đòi hỏi một băng thông cực lớn để truyền tải dữ liệu, công nghệ truyền không dây Wi-fi dần trở nên không đáp ứng được, do đó không thể tránh khỏi việc nghiên cứu và đưa ra một công nghệ mới có thể đáp ứng được nhu cầu to lớn đó. Một trong những công nghệ đã nhận được được sự quan tâm mạnh mẽ là truyền thông không dây quang Optical wireless communication (OWC).

OWC là ứng viên tiềm năng cho truyền dẫn không dây trong nhà (indoor). Kỹ thuật này sử dụng ánh sáng để truyền dẫn thông tin thay cho sóng điện tử (vốn đã hạn chế về băng thông). Do đó ánh sáng không thể xuyên qua vật thể nên sẽ không gây can nhiễu với các hệ thống khác và mang lại tính bảo mật cao. Hệ thống OWC có các ưu điểm như thiết lập nhanh, tốc độ truyền nhanh hơn so với các công nghệ truyền thông không dây hiện nay như Bluetooth, Wifi. Hơn nữa, việc sử dụng những linh kiện đơn giản với giá thành thấp như sử dụng organic light-emitting diode (OLED) ở phía phát và photodetector ở phía thu, khiến cho việc sử dụng vào thực tế sẽ dễ dàng hơn. Với sự phát triển của công nghệ vật liệu, đặc biệt là sự phát triển không ngừng của OLED đã hướng sự quan tâm về việc nghiên cứu cũng như ứng dụng công nghệ truyền thông không dây sử dụng ánh sáng khả kiến VLC. Đây được xem là một giải pháp khả thi cho truyền thông quang không dây trong tương lai [1].

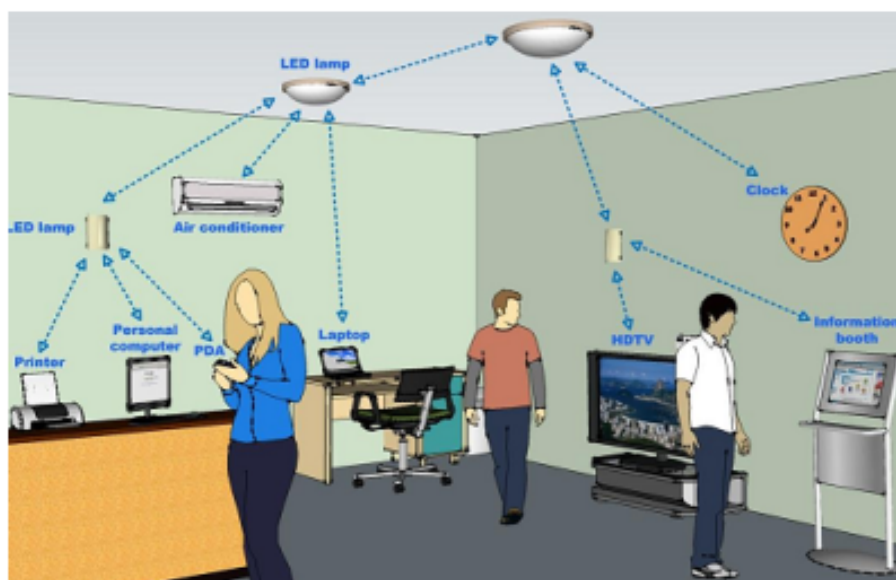
Giao tiếp ánh sáng khả kiến (VLC) là một biến thể truyền thông dữ liệu sử dụng ánh sáng nhìn thấy được giữa 400 và 800 THz (780–375 nm). VLC là một tập hợp con của các công nghệ truyền thông không dây quang học.

Công nghệ này sử dụng đèn huỳnh quang (đèn thông thường, không phải thiết bị liên lạc đặc biệt) để truyền tín hiệu ở tốc độ 10 kbit/s hoặc đèn LED lên đến 500 Mbit/s trong khoảng cách ngắn.

VLC có thể được sử dụng như một phương tiện truyền thông cho điện toán phổ biến, vì các thiết bị mang ánh sáng (như đèn trong nhà / ngoài trời, TV, biển báo giao thông, màn hình thương mại và đèn pha / đèn hậu xe hơi) được sử dụng ở mọi nơi. Sử dụng ánh sáng có thể nhìn thấy được cũng ít nguy hiểm hơn đối với các ứng dụng công suất cao vì con người có thể cảm nhận được nó và hành động để bảo vệ mắt khỏi bị hư hại [2].



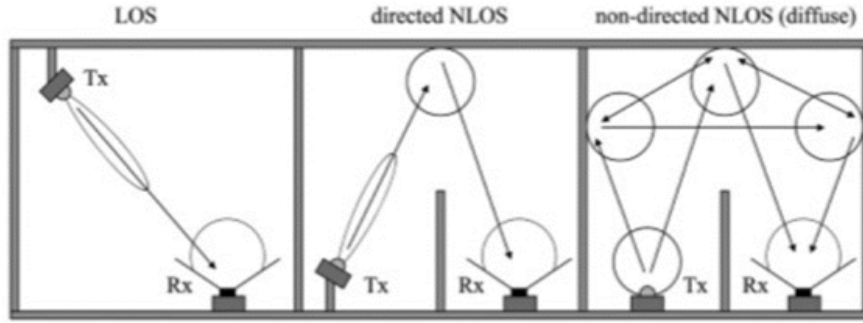
Hình 2.15: Vùng ánh sáng khả kiến được sử dụng cho VLC



Hình 2.16: Công nghệ VLC trong tương lai

2.2.2 Phân loại kênh truyền

Cấu hình đường truyền cho hệ thống VLC dựa vào mức độ định hướng giữa bộ phát và bộ thu, do đó được chia thành ba loại: Trực tiếp (LOS-Line Of Sight), không trực tiếp (NLOS-directed Non Line Of Sight) và Lai ghép (non-directed NLOS). Đường truyền trực tiếp (LOS) từ bộ phát và bộ thu có công suất cao nhất vì nó chịu suy hao nhỏ nhất từ ảnh hưởng từ môi trường. Đối với đường truyền không trực tiếp, các thiết bị di động dễ dàng nhận được tín hiệu ngay cả khi đang di chuyển nhưng công suất tín hiệu không cao do tín hiệu bị phân tán và chịu ảnh hưởng từ các nguồn sáng khác từ môi trường. Cấu hình lai ghép mức độ định hướng giữa bộ phát-thu có sự khác biệt, công suất nhận được cao hơn công suất của bộ không trực tiếp do độ trung ánh sáng của bộ phát, nhưng nhỏ hơn truyền trực tiếp và vẫn bị ảnh hưởng bởi các nguồn sáng do độ mở của bộ nhận lớn.



Hình 2.17: Mô hình kênh truyền trong hệ thống VLC

Các tham số hiệu năng của kênh truyền

a) Tỷ số tín hiệu trên nhiễu SNR (Signal-to-Noise Ratio)

Hàm mật độ công suất nhiễu có thể được xác định như sau:

$$N_0 \cong N_{Shot} = 2qyP_n \sim 10^{-22} \left(\frac{A^2}{Hz} \right) \quad (2.1)$$

Trong đó:

y là hệ số đáp ứng

P_n là năng lượng ánh sáng trung bình

Từ đó, với mỗi tốc độ bit R_b nào đó và P_r là công suất quang nhận được, ta có tỉ lệ lỗi bit SNR được xác định như sau:

$$SNR = \frac{y^2 P_r^2}{R_b N_0} \quad (2.2)$$

b) Dung năng kênh

Theo định lý Shannon, dung năng kênh C (tốc độ truyền dữ liệu tối đa ứng với một giá trị SNR cho trước) được xác định bởi:

$$C = B \log_2(1 + SNR) \quad (2.3)$$

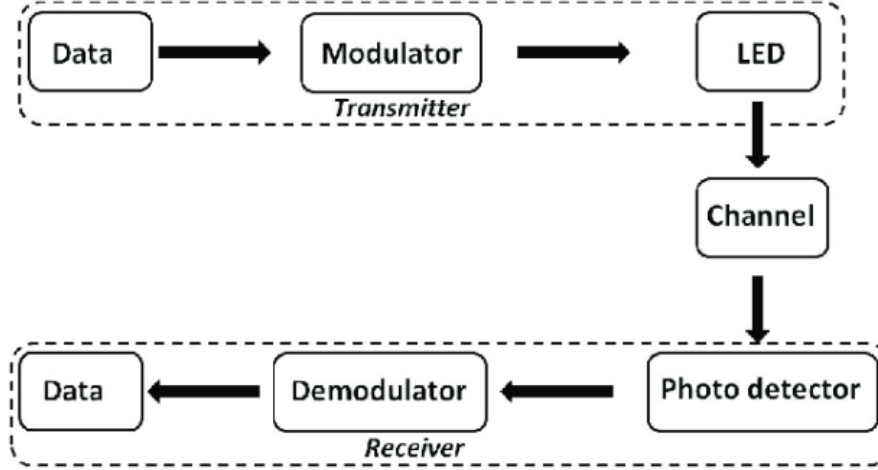
c) Tỷ lệ lỗi bit BER (Bit Error Rate)

$$BER = \frac{Errorbits}{bits} \quad (2.4)$$

Mức ngưỡng BER của hệ thống là 0.0038.

Những kiến thức trên là từ hệ thống thông tin vô tuyến nói chung, do đó hệ thống VLC có đôi chút khác biệt. Do hệ thống VLC sử dụng chủ yếu ở môi trường trong nhà, kênh truyền ít biến động, nên trong bản báo cáo này ta xét mô hình LOS cho hệ thống. (2.5) mô tả suy hao kênh truyền trong kịch bản LOS-VLC (Line of sight visible light communication). Trong công thức đó, khoảng cách giữa bên thu với bên phát là D và độ mở của bộ thu là r . Góc hợp bởi đường thẳng nối giữa phía phát và phía thu với trục máy thu là α và với trục máy phát là β . ω_r là góc quan sát của máy thu, và A_r là diện tích thu.

$$A_r \cos(\alpha) = D^2 \omega_r \quad (2.5)$$



Hình 2.18: Kênh truyền hệ thống VLC

Do đây là hệ thống VLC, nên yếu tố ánh sáng cũng phải được xem xét. Luồng ánh sáng hấp thu được ở phía thu và suy hao kênh truyền ánh sáng lần lượt được tính theo (2.6)

$$F_r = \log_s(\beta)\omega_r \quad (2.6)$$

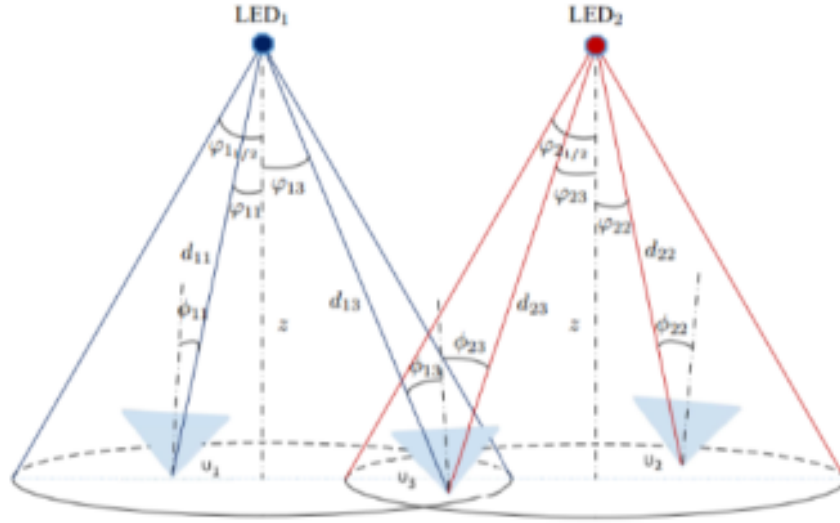
$$L_L = \frac{F_r}{F_s} = \frac{g_s(\beta)A_r \cos(\alpha)}{D^2 \int_0^{\Theta_{max}} 2\pi g_s(\Theta) \sin(\theta) d\theta} \quad (2.7)$$

Trong những phương trình trên, $g_s(\beta)$ là hàm phân bố ánh sáng chuẩn hoá, được cung cấp bởi nhà phân phối thiết bị. Đơn giản hoá (2.7), suy hao kênh truyền LOS được tính toán với bậc lambert $m = \frac{\ln(2)}{\ln(\cos(\alpha/2))}$ và góc nửa công suất của phía thu $\alpha(1/2)$. Trong (2.7), β đại diện cho góc Field Of View (FOV) của máy thu, A_r là diện tích hấp thụ của bộ thu. Do đó, hệ số kênh truyền phụ thuộc vào vị trí của máy thu so với máy phát. Như vậy, suy hao kênh truyền tổng quát của hệ thống VLC được tính theo công thức

$$L_L = \frac{(m+1)A_r}{2\pi D^2} \cos(\alpha) \cos^m(\beta) \quad (2.8)$$

Hệ thống MIMO quang không dây

Cách tiếp cận dựa trên việc sử dụng kỹ thuật Multiple-input Multiple-output (MIMO), liên quan đến truyền thông không dây, có nghĩa là dữ liệu được truyền dưới dạng nhiều hơn một luồng, như vậy cách mà nhiều con đường truyền được tạo thành đồng thời, giữa nhiều anten phát và nhiều anten thu. Tất nhiên, khi chúng ta nói về việc thực hiện truyền MIMO trong lĩnh vực quang học, chúng ta đang suy nghĩ về nhiều bộ phát và dò ánh sáng thay vì ăng-ten vô tuyến. Tuy nhiên, ý tưởng chính vẫn được duy trì. Nếu chúng ta nói về hệ thống VLC trong nhà, nơi có đèn LED được sử dụng làm nguồn sáng, thường được cấu thành từ nhiều đèn LED riêng biệt. Về mặt lý thuyết, có thể tách các bộ truyền dữ liệu riêng lẻ ở cả mức độ toàn bộ bóng đèn, nếu nhiều hơn một bóng đèn được lắp đặt trong cùng một nơi, cũng như số lượng của đèn LED riêng lẻ hoặc các nhóm bên



Hình 2.19: Mô hình MIMO trong phòng

trong một bóng đèn. Bất kể trường hợp nào trong thực tế, sự hoạt động của hệ thống MIMO-VLC có thể được mô tả bằng một mô hình toán học, giải thích bằng đồ thị được mô tả ở 2.20. Trong hệ thống truyền dẫn như vậy, các tín hiệu được truyền đồng thời bởi N máy phát và M máy thu, trong đó ($M \geq N$) được mô tả bởi phương trình sau [6]:

$$X = HY + n \quad (2.9)$$

Trong đó X là vectơ của tín hiệu truyền, Y là vectơ tín hiệu nhận, H là ma trận của kênh truyền chuẩn.

Ma trận này được hình thành bởi độ truyền giữa mỗi máy phát và mỗi máy thu, và có thể được ước tính bằng độ lợi giữa máy thu thứ i và máy phát thứ j . Bằng cách nghịch đảo ma trận H , ta có thể phục hồi lại tín hiệu ban đầu, theo công thức:

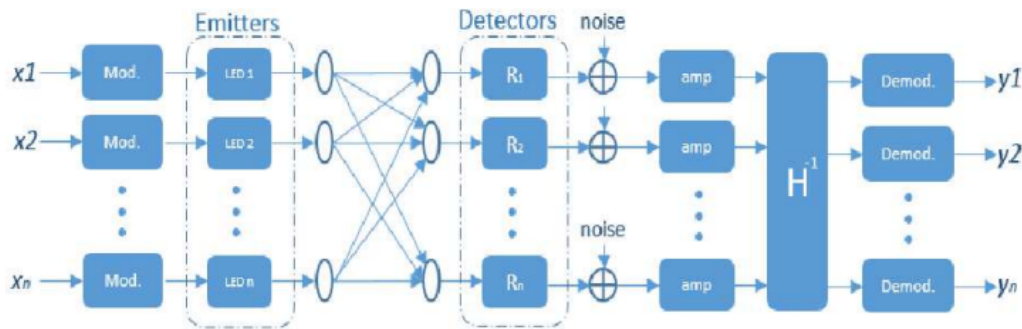
$$Y = H^{-1}X + n \quad (2.10)$$

Hiển nhiên, ma trận H có thể không phải là ma trận 1 chiều và điều kiện của nó quyết định Signal to Noise Ratio (SNR). Ma trận H^{-1} có thể được tìm thấy bởi 1 số thuật toán tối ưu hoá hoặc được tính toán ở giai đoạn khởi tạo hệ thống. Trong thực tế, phương pháp cuối cùng là thường được sử dụng. Bởi vì quá trình phục hồi tín hiệu ở bên thu, kiến thức về ma trận H^{-1} là cần thiết, hệ thống hoạt động tốt hơn ở 2 pha. Ở pha đầu tiên, ma trận H được định nghĩa, thông thường 1 số tín hiệu pilot được truyền. Dựa vào đó, ma trận H^{-1} được tính. Ở giai đoạn 2, tín hiệu chứa thông tin được truyền. Ma trận H^{-1} được sử dụng trong quá trình khôi phục tín hiệu.

2.3 Neural Network

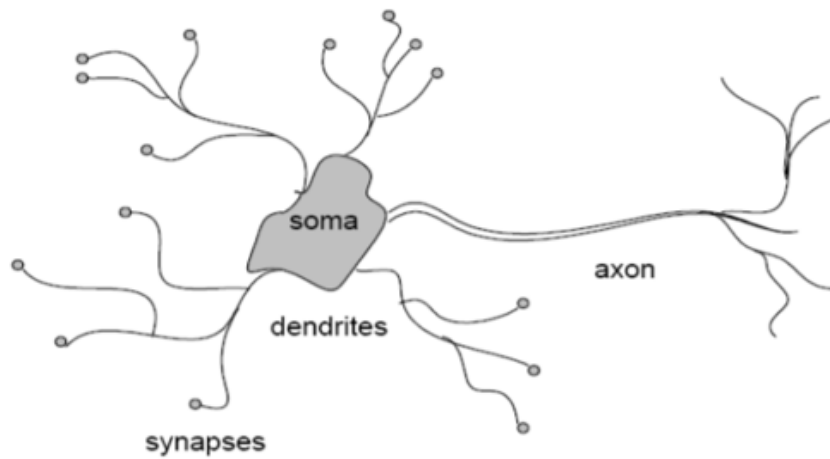
2.3.1 Mạng thần kinh học

Bộ não con người là một hệ thống xử lý thông tin phức hợp, phi tuyến và song song có khả năng học, ghi nhớ, tổng quát hóa và xử lý lỗi. Những khả năng có được của não bộ



Hình 2.20: Mô hình của hệ thống giao tiếp ánh sáng MIMO

là nhờ nó được cấu thành từ một mạng thần kinh chứa rất nhiều tế bào neural sinh học và con số đó được ước tính khoảng 100 tỷ tế bào.



Hình 2.21: Sơ đồ 1 neural sinh học

Mỗi neural nhận xung thần kinh (thông tin) truyền từ neural đứng trước nó, truyền dọc theo Axon và truyền ra neural sau thông qua Synapse. Xung động thần kinh truyền từ neural này sang neural khác trong mạng thần kinh tạo nên mọi chức năng thần kinh của não bộ.

Con người với ước muốn nhân tạo hóa mạng thần kinh sinh học nên đã tìm cách mô phỏng cách não bộ hoạt động bằng máy tính và các thuật toán để phục vụ nhiều mục đích trong đời sống.

2.3.2 Mạng thần kinh nhân tạo

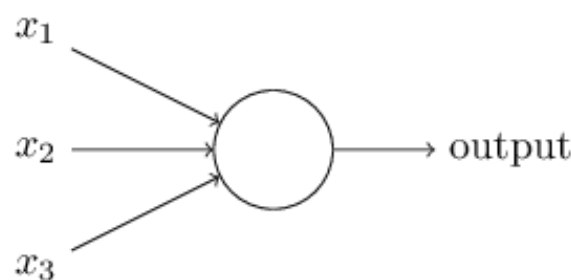
Mạng thần kinh nhân tạo (Artificial neural network – ANN) là mô hình xử lý thông tin được mô phỏng dựa trên hoạt động của hệ thống thần kinh sinh vật, bao gồm số lượng lớn các neural được gắn kết thông qua các trọng số liên kết để xử lý thông tin. ANN giống như não bộ con người, được học bởi kinh nghiệm (thông qua huấn luyện), có khả năng lưu giữ những kinh nghiệm hiểu biết (tri thức) và sử dụng những tri thức đó trong việc dự đoán những dữ liệu chưa biết.

Người ta phân loại mạng neural dựa vào các kiểu kết nối của các neural và dựa vào số lớp neural trong mạng:

- Phân loại dựa theo kiểu kết nối các neural: dựa theo kiểu kết nối, ta có mạng neural truyền thẳng (feedforward Neural Network) và mạng hồi quy (recurrent NN). Trong mạng truyền thẳng, các kết nối đi theo một hướng nhất định, không tạo thành chu trình. Ngược lại, các mạng hồi quy cho phép các kết nối neural tạo thành chu trình, với đỉnh là các neural và cung là các kết nối giữa chúng. Các neural nhận tín hiệu vào gọi là neural vào, các neural đưa thông tin ra gọi là neural ra, các neural còn lại gọi là neural ẩn.
- Phân loại theo số lớp neural: các neural trong mạng có thể được tổ chức thành các lớp theo nguyên tắc neural ở lớp này, chỉ được nối với các neural ở lớp khác, không cho phép kết nối giữa các neural trên cùng một lớp, hoặc từ neural lớp dưới lên neural lớp trên. Lớp nhận tín hiệu vào gọi là lớp vào, lớp đưa thông tin ra gọi là lớp ra, các lớp ở giữa gọi là lớp ẩn. Thông thường lớp vào không tham gia quá trình tính toán của mạng neural, nên khi tính số lớp người ta không kể lớp vào.

Kiến thức cơ bản của mạng thần kinh nhân tạo

Perceptron: Tên gọi của các neuron trong mạng thần kinh nhân tạo (ANN). Nó có chức năng nhận đầu vào và xuất đầu ra.



Hình 2.22: Cấu trúc của 1 perceptron

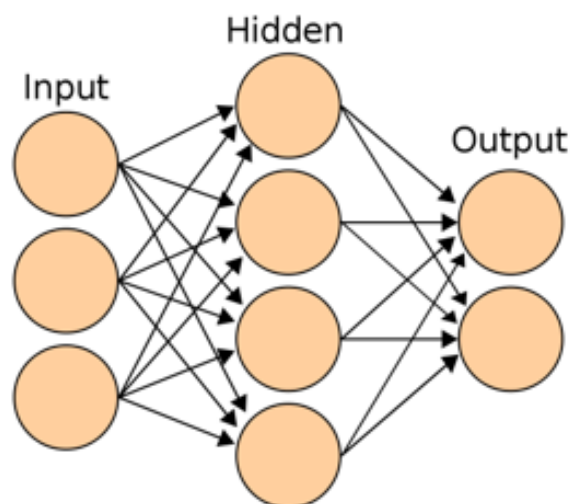
Một ANN cấu thành từ nhiều lớp khác nhau, trong đó gồm:

- Lớp vào (Input layer): có chức năng nhận dữ liệu đầu vào.
- Lớp ẩn (Hidden layer): nhận dữ liệu từ các lớp trước nó, sau đó xử lý và lan truyền dữ liệu đến các lớp tiếp theo. Trong một ANN thường có nhiều lớp ẩn, số lượng sẽ phụ thuộc vào người thiết kế mạng.
- Lớp ra (Output layer): tổng hợp thông tin nhận được và cho ra kết quả tính toán của mạng.

2.3.3 Probabilistic neural network (PNN)

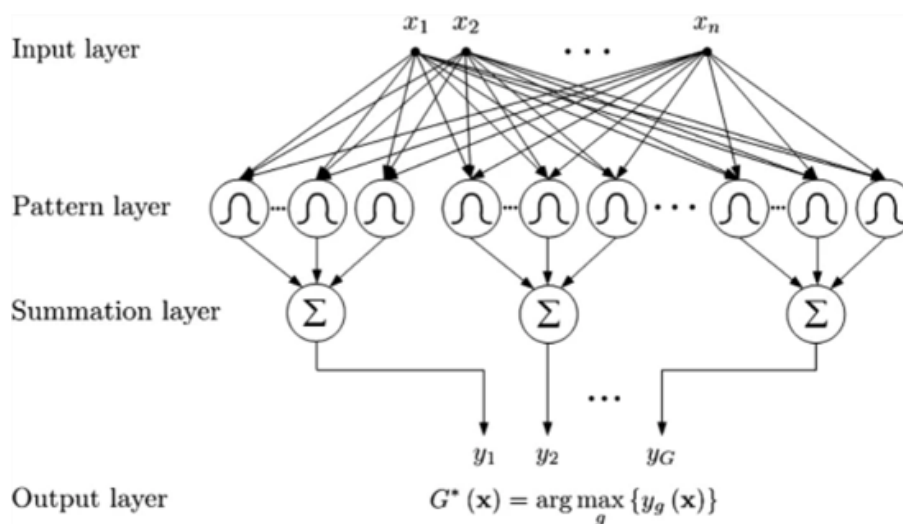
Kiến thức cơ bản về PNN

Mạng nơ-ron xác suất PNN là một mạng nơ-ron truyền thẳng (các kết nối giữa các nơ-ron không tạo thành 1 chu trình), được sử dụng nhiều trong các bài toán phân lớp và nhận



Hình 2.23: Cấu trúc của 1 mạng ANN

điển đặc điểm. Trong thuật toán PNN, hàm mật độ xác suất mẹ Probability Distribution Function (PDF) của từng lớp được tính gần bằng bằng cửa sổ parzen và một hàm phi tham số. PDF của mỗi lớp sau đó được sử dụng để tính xác suất lớp của dữ liệu đầu vào mới cùng với quy tắc Bayes được áp dụng để phân bổ lớp có xác suất cao nhất cho lớp dữ liệu đầu vào mới. Loại Artificial Neural Networks (ANN) này được bắt nguồn từ mạng Bayes và một thuật toán thống kê được gọi là phân tích phân biệt Kernel Fisher, được giới thiệu bởi D.F. Specht năm 1966 [5].



Hình 2.24: Sơ đồ cấu trúc một PNN

PNN có 4 lớp như sau: Input layer (Lớp đầu vào), Pattern layer (Lớp mẫu), Summation layer (Lớp tổng hợp), Output layer (Lớp đầu ra).

a) Input layer (Lớp đầu vào):

Mỗi neural trong lớp này đại diện cho một biến cần dự đoán x_1, x_2, \dots, x_n , nó nhận giá trị của biến dự đoán sau đó cung cấp cho từng neural trong lớp ẩn.

b) Pattern layer (Lớp mẫu):

Số neural trong lớp này ứng với số trường hợp trong tập dữ liệu huấn luyện – các vector đặc trưng và nhân của nó (giả sử có G nhân trong PNN).

Mỗi nút trong lớp này chứa tương ứng với một hàm Gauss cho mỗi vector đặc trưng mẫu. Hàm Gauss này tập trung vào vector đặc trưng liên kết của nó.

Đầu ra của mỗi nút được tính theo công thức sau:

$$f_i = \frac{1}{\sqrt{(2\pi\sigma_g^2)^N}} e^{-\frac{\|x_i - x^{(p)}\|^2}{2\sigma_g^2}} \quad (2.11)$$

Trong đó:

- σ_g : là độ lệch chuẩn của hàm gauss ứng với mỗi nhân nó có thể bằng một nửa khoảng cách từ mẫu đến vector mẫu khác gần nhất.
- x_i : là vector đầu vào cần xác định nhân.
- N : là kích thước vector mẫu đầu vào x_i
- p : là số lượng mẫu đặc trưng cho một nhân
- $\|x_i - x_{(p)}\|$: là khoảng cách Euclide (căn bậc hai của tổng bình phương) giữa x_i và $x_{(p)}$

c) Summation layer (Lớp tổng hợp):

Ta có G neural trong lớp này ứng với số lượng nhân trong mạng, mỗi nút cung cấp giá trị cho lớp đầu ra y_g , bằng cách tính tổng các giá trị đầu vào nhận được từ các neural lớp mẫu của từng nhân tương ứng. Các giá trị y_g này ta gọi là trọng số của phiếu bầu cho nhân mục tiêu ứng với x_i .

d) Output layer (Lớp đầu ra):

Nhiệm vụ lớp này so sánh các phiếu bầu có trọng số được nhận từ lớp tổng hợp, nhân có trọng số phiếu bầu lớn nhất được dùng để dự đoán loại mục tiêu cho đầu vào x_i .

Cửa sổ Parzen

1 trong những thuật toán phi tham số để ước tính PDF là "cửa sổ Parzen". Để tính toán PDF $p(x)$ tại điểm x, yêu cầu xác định số mẫu N_h với khoảng $[x - h, x + h]$ và chia cho tổng số vectơ mẫu M và $2h$. Công thức ước lượng PDF tại x

$$\hat{p}(x) = \frac{N_h(x)}{2hM} \quad (2.12)$$

Với hàm hỗ trợ k_h , ta có thể chọn

$$K_h = \begin{cases} 0,5 & \text{nếu } |m| \leq |h| \\ 0 & \text{nếu } |m| > |h| \end{cases}$$

Từ (2.12) chúng ta có

$$\hat{p}(x) = \frac{1}{hM} \sum_{i=1}^M K\left(\frac{x - m}{h}\right) \quad (2.13)$$

Với tổng thứ i bằng không khi m_i không thuộc khoảng $[x - h, x + h]$, điều này dẫn đến

$$\gamma(x, m) = \frac{1}{h} K\left(\frac{x - m}{h}\right) \quad (2.14)$$

Nếu như \hat{p} được xem như hàm liên quan đến số lượng mẫu, chúng ta suy ra

$$\hat{p}(x) = \hat{p}(x, M) \quad (2.15)$$

\hat{p} không sai lệch khi $M \rightarrow \infty$, nếu $h = h(M)$

Trong thực tế, chỉ xảy ra với những số hữu hạn. Lựa chọn cho thông số h là cực kỳ quan trọng, vì thế được khuyến khích bắt đầu từ ước lượng ban đầu của h rồi sau đó thay đổi từ từ để giảm thiểu sai số phân lớp sai. 1 cách lý thuyết, số M lớn là cần thiết. Nhưng trong thực tế, 1 số lượng lớn các điểm làm tăng sự phức tạp tính toán 1 cách không cần thiết.

Sự lựa chọn thông thường của hàm $K(m)$ là

$$K(m) = (2\pi)^{1/2} e^{-\frac{m^2}{2}} \quad (2.16)$$

$$K(m) = \frac{1}{\pi(1 + m^2)} \quad (2.17)$$

$$K(m) = \begin{cases} 1 - |m| & \text{nếu } |m| \leq 1 \\ 0 & \text{nếu } |m| > 1 \end{cases}$$

Thuật toán Bayes cho bài toán phân lớp

1 quy chuẩn được chấp nhận cho các quy tắc lựa chọn hoặc chiến lược được dùng để giải bài toán phân lớp là chúng làm theo cách mà giảm thiểu "rủi ro mong đợi". Chiến lược như vậy gọi là "chiến lược Bayes và có thể được ứng dụng trong các vấn đề chứa nhiều thể loại.

Xem xét 1 tình huống 2 loại trong đó trạng thái của θ được biết là θ_A hoặc θ_B . Nếu như bài toán mong muốn xác định $\theta = \theta_A$ hoặc $\theta = \theta_B$ dựa trên 1 tập các chỉ số đo được biểu hiện bởi 1 vector có ρ chiều $X^t = [X_1..X_2..X_3]$, quy tắc Bayes trở thành [3]

$$\begin{aligned} d(X) &= \theta_A, \text{ nếu } h_A I_A f_A(X) > h_B I_B f_B(X) \\ d(X) &= \theta_B, \text{ nếu } h_A I_A f_A(X) < h_B I_B f_B(X) \end{aligned} \quad (2.18)$$

Trong đó $f_A(X)$ và $f_B(X)$ là các hàm mật độ xác suất PDF cho loại A và loại B. I_A là hàm mất mát với quyết định $d(X) = \theta_B$ khi $\theta = \theta_A$; I_B là hàm mất mát với quyết định $d(X) = \theta_A$ khi $\theta = \theta_B$ (Sự mất mát liên quan đến quyết định đúng sẽ bằng 0). h_A là xác suất giả thuyết của dữ liệu quan sát thuộc loại A và $h_B = 1 - h_A$ là xác suất giả thiết khi $\theta = \theta_B$.

Do đó đường biên giới giữa các vùng trong đó quyết định Bayes $d(X) = \theta_A$ và vùng trong đó $d(X) = \theta_B$ được chỉ bởi

$$f_A(X) = K f_B(X) \quad (2.19)$$

Trong đó

$$K = h_B I_B / h_A I_A \quad (2.20)$$

Thông thường, 2 loại quyết định được cho bởi (2.19) có thể phức tạp tùy ý, khi không có 1 giới hạn cho mật độ ngoại trừ các điều kiện mà hàm mật độ xác suất phải tuân theo, ví dụ như chúng có thể không âm, tích hợp và các tích phân của chúng trong mọi không gian đều bằng nhau. Các quy luật quyết định tương tự có thể được dùng trong mọi bài toán nhiều loại.

Chìa khoá để sử dụng (2.19) là khả năng dự đoán PDF dựa trên các tập training. Thông thường xác suất giả thuyết có thể được biết hay được dự đoán chính xác và hàm mất mát yêu cầu sự đánh giá chủ quan. Tuy nhiên, nếu mật độ xác suất của mẫu thuộc loại bị phân chia là không biết, và tất cả được cho bởi 1 tập training mẫu, khi đó mẫu này chỉ cung cấp bằng chứng cho mật độ xác suất quan trọng chưa biết.

Ưu điểm và nhược điểm của PNN

Mạng nơ-ron xác suất có nhiều ưu và nhược điểm so với mạng nơ-ron nhiều lớp truyền thống

a) Ưu điểm:

- PNN nhanh hơn sơ với mạng nơ-ron nhiều lớp truyền thống.
- PNN cho kết quả chính xác hơn so với mạng nơ-ron nhiều lớp truyền thống.
- PNN tương đối nhạy cảm với những ngoại lệ.
- PNN tạo ra các điểm xác suất mục tiêu dự đoán chính xác.
- PNN tiệm cận phân loại tối ưu Bayes.

b) Nhược điểm:

- PNN chậm hơn mạng nơ-ron nhiều lớp truyền thống ở phân loại các trường hợp mới.
- PNN yêu cầu nhiều bộ nhớ để lưu trữ mô hình

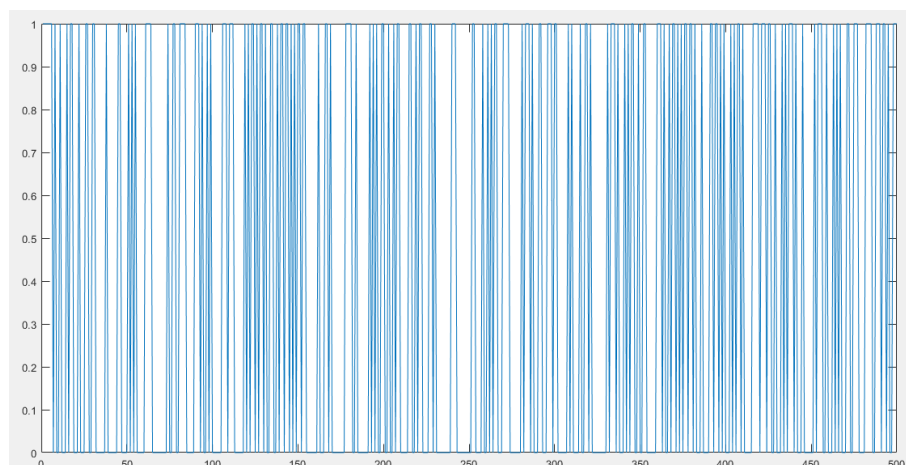
Cách ứng dụng mạng nơ-ron xác suất vào đề tài luận văn

Trong đề tài luận văn này, tín hiệu được sử dụng là tín hiệu NRZ 2 mức 0 và 1, sau khi đi qua hệ thống sẽ bị biến đổi thành tín hiệu có các mức khác nhau. Bài toán đặt ra ở bộ cân bằng là làm thế nào để biến đổi các mức tín hiệu khác nhau về thành tín hiệu ban đầu có 2 mức 0 và 1. Bài toán này tương tự như bài toán phân lớp nên chúng em quyết định chọn mạng nơ-ron xác suất để giải quyết vấn đề. Ngoài ra còn 1 điểm nữa là mô hình PNN là 1 hàm có sẵn trong matlab mà còn rất tối ưu nên rất thuận tiện cho việc xử lý chuỗi tín hiệu lên đến hàng ngàn bit.

Sau khi đi qua hệ thống thực tế, tín hiệu sẽ bị suy hao + méo dạng như hình 2.27. Khi phân tích, ta sẽ nhận được 1 chuỗi gồm 500 symbols nhưng mỗi symbols sẽ được biểu diễn bởi các 3 mức khác nhau như hình 2.28. Bằng cách sử dụng PNN, ta sẽ biến đổi 3 samples đây về thành 1 mức 0 hoặc 1 cho giống với chuỗi truyền đi. Như vậy đây là bài toán phân lớp 1500 điểm về thành 2 lớp 0 hoặc 1.

transmitted_symbol																
1x500 double																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	0	0	0	0	0	1	0	1	1	0	1	1	1	0	1	

Hình 2.25: 1 chuỗi gồm 500 bit ngẫu nhiên 0 và 1 được tạo từ code Matlab



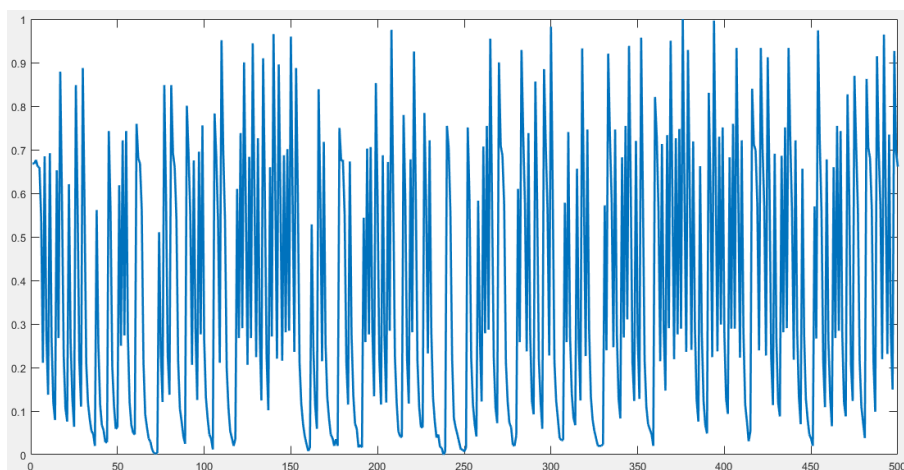
Hình 2.26: Hình ảnh của chuỗi được đưa vào máy phát sóng

2.4 Kết luận chương

Trong đề tài, tín hiệu NRZ được sử dụng 2 mức 0 và 1 được sử dụng vì ít tiêu tốn băng thông với lại đơn giản, dễ dàng quan sát khi có lỗi xuất hiện khi chuỗi tín hiệu phát đi là 1 chuỗi dài.

Tín hiệu thu được từ hệ thống thực tế sẽ bị nhiễu phi tuyến + đường truyền không ổn định vì thế phương pháp mạng nơ-ron nhân tạo được sử dụng với các lý do kể trên.

Ở chương kết tiếp, các thông số cụ thể của hệ thống MIMO-VLC cũng như các bước thực hiện thí nghiệm, kết quả và đánh giá phương pháp neural network sẽ được trình bày.



Hình 2.27: Hình ảnh của chuỗi bit thu được

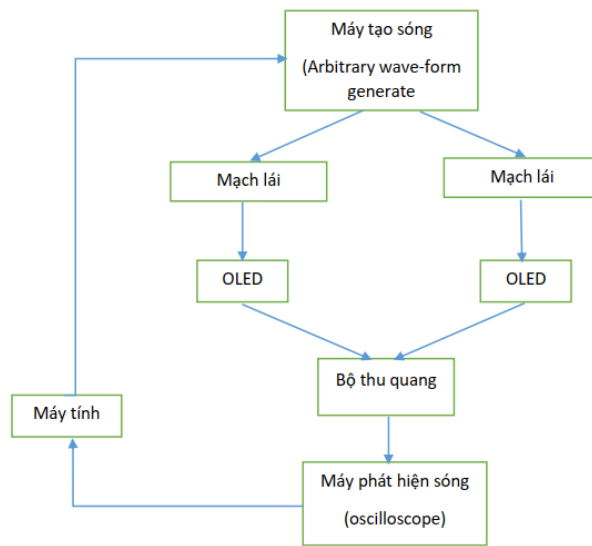
extracted_received_symbol																
3x500 double																
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	6.8667	6.9667	6.8667	6.8333	6.8667	6.6000	3.2000	6.5333	4.1333	2.0667	5.2667	3.9333	2.0667	1.2333	4.6667	3.9333
2	6.9667	6.8667	7	6.8000	6.9333	5.9333	2.3667	8.0667	3.0667	1.7333	10.4333	2.9333	1.7333	1.3333	9.9333	2.9333
3	6.8000	6.8000	7	6.8000	6.6000	4.6000	2.2000	6.5000	2.4000	1.9000	5.6000	2.4000	1.4000	1.5000	5.6000	2.5000

Hình 2.28: Chuỗi bit thu được gồm 500 symbols và 1 symbols được biểu diễn bởi 3 samples

Chương 3. KẾT QUẢ NGHIÊN CỨU

3.1 Phương pháp tiếp cận

3.1.1 Sơ đồ khối hệ thống



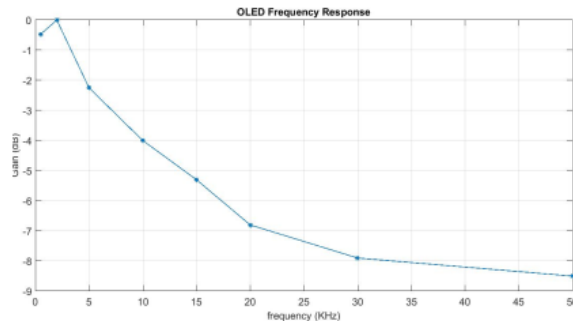
Hình 3.1: Sơ đồ khối hệ thống MIMO-VLC

a) TRANSMITTER

Trong thập kỷ qua, ta đã chứng kiến sự sử dụng rộng rãi của diode phát quang (LED) trong rất nhiều ứng dụng trong đời sống. LED đã chứng minh mình là một công nghệ mạnh mẽ cho lĩnh vực chiếu sáng, nhưng giờ đây, ngoài khả năng chiếu sáng, khoảng băng tần khả kiến của LED còn có thể sử dụng cho truyền nhận dữ liệu không dây, mà tại đó LED chính là nguồn phát dữ liệu. Bên cạnh LED, đèn LED hữu cơ (OLED) cũng đã là cơ sở cho nhiều ứng dụng chiếu sáng và trình chiếu, đặc biệt là trong điện thoại di động hoặc các thiết bị cầm tay khác. Việc này là nhờ vào công năng tuyệt vời của nó, như là kích thước nhỏ gọn, nhẹ, hình dáng có thể tùy chỉnh, tiêu thụ ít công suất, chi phí sản xuất thấp.

Cả LED và OLED đều một lựa chọn rất được các nhà nghiên cứu quan tâm để dùng làm nguồn sáng cho công nghệ VLC. Tuy nhiên, OLED lại có băng thông điều chế thấp hơn rất nhiều so với đèn LED thông thường. Không giống như LED thông thường, OLED là tập hợp các phân tử hữu cơ, do đó độ linh động của nó thấp hơn nhiều so với các thiết bị silicon khác, kết quả là sẽ giới hạn băng thông điều chế của nó. Băng thông điều chế của OLED còn phụ thuộc vào nhiều thứ, như là quá trình sản xuất, kích cỡ,...nhưng băng thông thông thường của chúng chỉ dừng lại ở vài trăm KHz.

Hình [3.1] mô tả đáp ứng tần số của OLED Lumiable sáng trắng tiêu chuẩn (do Philip sản xuất), có diện tích mặt phát quang là $4\pi\text{cm}^2$. Trong chiếu sáng và trang trí trên thị trường, đây là một trong số ít các OLED được bày bán. Các phép đo cho thấy băng thông điều chế của OLED này là khoảng 7kHz.



Hình 3.2: Đáp ứng tần số của OLED

b) RECEIVER

Photodiode là một diode bán dẫn thực hiện biến đổi photon thành dòng điện theo hiệu ứng quang điện, các photon có thể thuộc vùng phổ của ánh sáng khả kiến, hồng ngoại hay tử ngoại,... Photodiode sử dụng trong khảo sát là APD410A do thorlab sản xuất, có thể thu được ánh sáng có bước sóng từ 400-1000 nm, băng thông 10MHz.

Các đặc tính cơ bản của bộ thu quang photodiode:

- Độ nhạy cao trong khoảng bước sóng hoạt động.
- Thời gian đáp ứng nhanh, để thỏa mãn được các hệ thống có tốc độ cao.
- Kết nối hiệu quả với cáp quang.
- Đặc tính ít thay đổi theo nhiệt độ và thời gian sử dụng.
- Tương thích với các thiết bị điện.
- Chi phí thấp, độ tin tưởng cao.

3.1.2 Thí nghiệm 1

Ở thí nghiệm đầu tiên, chúng em sẽ khảo sát và so sánh tỉ lệ lỗi bit ở các tốc độ bit khác nhau, sử dụng đồng thời 2 phương pháp ước lượng truyền thống và probabilistic neural network.

Các bước tiến hành thí nghiệm:

Bước 1: Tạo code tín hiệu NRZ từ máy tính bằng phần mềm matlab. Tín hiệu được tạo sẽ có chiều dài chuỗi bit, tốc độ lấy mẫu và biên độ ứng với các biến trong code.

Bước 2: Đưa tín hiệu vào máy tạo sóng. Có 2 điều cần lưu ý là phải giảm điện áp lái trên nguồn DC về 0 trước khi bấm nút "Output" để phát tín hiệu và ghi chú lại thời gian T trên đồ thị tín hiệu trên màn hình máy tạo sóng.



Hình 3.3: Bộ thu quang APD410A của Thorlab

Bước 3: Kết nối dây từ nguồn DC, máy phát sóng và đèn OLED với mạch lái. Chú ý để mạch lái trên vật dụng bằng nhựa hoặc gỗ để tránh chạm mạch.

Bước 4: Bật nút "OUTPUT" trên máy tạo sóng, tăng điện áp nguồn lên dần dần. Quan sát thông số công suất trên màn hình DC và dừng tăng điện áp khi công suất đến ngưỡng 0.2 W-0.3 W. Nếu để công suất quá lớn sẽ làm cháy OLED do mạch lái OLED tạo các gai lớn.

Bước 5: Tín hiệu sau khi được thu qua photodiode sẽ hiển thị lên màn hình oscilloscope, lúc này ta cần chỉnh trục thời gian $T'=2T$ để chắc chắn tín hiệu thu được chuỗi tín hiệu đầy đủ không bị mất.

Bước 6: Sử dụng matlab để xử lý tín hiệu thu được và sau đó cho qua bộ neural network để khảo sát.

Bước 7: Lặp lại các bước như trên với các tốc độ bit khác nhau. Cách làm tổng quát sẽ là từ lý thuyết ta sẽ khảo sát ở tốc độ bit từ 50k đến 60k để tìm xem tỉ lệ lỗi bit sau khi đã xử lý qua neural network sẽ đạt ngưỡng ở tốc độ nào, sau đó ta sẽ tăng tốc độ hoặc giảm tốc độ rồi lấy dữ liệu để so sánh.

3.1.3 Thí nghiệm 2

Ở thí nghiệm này, ta sẽ so sánh tỉ lệ lỗi bit đã qua xử lý neural network ở các khoảng cách khác nhau với cùng 1 tốc độ bit là 50kb/s. Các bước tiến hành như thí nghiệm 1 chỉ khác là ta sẽ giữ nguyên tốc độ bit và thay đổi khoảng cách.

Chương 4. KẾT LUẬN

4.1 Tóm tắt và kết luận chung

Qua các số liệu cũng như các kết quả đã được đưa ra chương 3, chúng em có các kết luận sau:

4.2 Hướng phát triển

Một số hướng phát triển của đề tài:

- nâng cấp mạch lái đèn led để hạn chế sự méo dạng cũng như xung cao để tránh làm hư đèn.
- Sử dụng tín hiệu lorentz sẽ tăng được tốc độ bit lên gấp 2-3 lần
- Tăng khoảng cách lên xa hơn để phù hợp với thực tế khác
- Áp dụng các phương pháp neural network khác như generalized regression neural network, convolutionary neural network,...

TÀI LIỆU THAM KHẢO

- [1] Đặng Khánh Toàn, *Thiết kế bộ cân bằng cho hệ thống VLC sử dụng mạng nơ-ron tích chập*, Luận văn kỹ sư, Trường Đại Học Bách Khoa TP.HCM, tháng 9, 2020.
- [2] Giới thiệu về giao thức ánh sáng khả kiến,
<https://bkaii.com.vn/tin-tuc/225-gioi-thieu-ve-giao-thuc-anh-sang-kha-kien>
- [3] Donald F. Specht, *Probabilistic Neural Networks*, Lockheed Missiles and Space Company, Inc, 1990
- [4] Nguyễn Ngọc Anh Khoa. *Mô phỏng hệ thống MIMO VLC sử dụng NOMA-OFDM/FBMC*. Luận văn kỹ sư, Trường Đại Học Bách Khoa TP.HCM, tháng 12, 2017.
- [5] Probabilistic Neural Network by Wikipedia,
https://en.wikipedia.org/wiki/Probabilistic_neural_network
- [6] M. Kowalczyk. *2x2 MIMO VLC Optical Transmission System Based on LEDs in a Double Role*. Warsaw University of Technology, Institute of Telecommunications, Warsaw, Poland.

Phụ lục A. CODE CHƯƠNG TRÌNH

A.1 Code tạo tín hiệu NRZ

```
1 %% example: creating m-PAM symbol for arbitrary singal generator
2 str = input('Mau_tao_vao_=', 's')
3 number_of_PAM_level = 2; % 2: NRZ; 4, 8, 16, ... m: m-PAM
4 bit_rate = 10000; % bit per second, use to calculate the
   frequency of the arbitrary singal generator, change if needed
5 number_of_transmitted_symbol = 400;
6 % number of transmitted m-PAMsymbols and pilot symbols, change
   if needed
7
8 symbol_rate = bit_rate / log2(number_of_PAM_level); % symbol per
   second, replated to number of bits per symbol
9 sample_rate = 1 * symbol_rate; % sample per symbol, does not
   implemented in this code
10
11 pilot_symbol = [1 0 1 0 0 0 1 0 1]; % pilot sequence to find the
   received symbols on the oscilloscope, change if needed
12
13 [transmitted_symbol, transmitted_bit] = PAM_symbol_creation(
   number_of_PAM_level, number_of_transmitted_symbol,
   pilot_symbol);
14
15 generator_period = length(transmitted_symbol) * 1/ sample_rate;
   % signal generator period in second
16 generator_frequency = 1 / generator_period; % Adjust the signal
   generator frequency to this number
17
18 visaAddress = 'USB::0x0699::0x0356::C020110::INSTR'; % Use VISA
   'Instrument Manager - Properties' to find the USB port
   address
19
20 generator_import(transmitted_symbol, visaAddress, sample_rate,
   generator_period); % import the transmitted symbol to the
   signal generator
21
22 file_name = sprintf('Transmitted_PAM_data_%s', str);
23 save(file_name); % save all variables for demodulation process
```

```

24
25 % file_name = sprintf('Transmitted_PAM_symbol_%s.csv', datestr(
    now, 'mm-dd-yyyy_HH-MM'));
26 % writematrix(transmitted_symbol, file_name); % save transmitted
    symbols in .csv file to import to ArbExpress program
27
28 %% other functions
29
30 function [transmitted_symbol, transmitted_bit] =
    PAM_symbol_creation(number_of_PAM_level,
        number_of_transmitted_symbol, pilot_symbol)
31 % This function create a sequence consists of pilot symbols and
    random PAM
32 % symbols.
33 % Input:
34 % number_of_PAM_level: number of m-PAM levels ,
    bit_per_PAM_symbol = log2(PAM_level)
35 % number_of_transmitted_symbol: total number of transmitted
    symbols, including pilot symbols and data symbols
36 % pilot_symbol: pilot symbols, select by the user
37 % Output:
38 % transmitted_symbol: transmitted m-PAM symbols, normalized
    from 0 to 1.
39 % transmitted_bit: matrix of trasmitted bits for BER
    calculation, size = [number of transmitted symbol, bit per
    symbol]
40
41     number_of_data_symbol = number_of_transmitted_symbol - size(
        pilot_symbol,2);
42
43     random_PAM_symbol = randi(number_of_PAM_level,[1 ,
        number_of_data_symbol]) -1;
44
45     transmitted_bit = de2bi(random_PAM_symbol);
46
47     % random m-PAM symbols, with magnitude scaled to 0-1
48     random_PAM_symbol_normalized = random_PAM_symbol / (
        number_of_PAM_level-1);
49
50     pos = strfind(random_PAM_symbol_normalized, pilot_symbol);
51
52     % replace data similar to pilot to avoid synchromization
    error
53     pilot_replace = 1 - pilot_symbol;
54     if isempty(pos) == 0
55         for index_pilot = 1:length(pos)
56             tem_pos = pos(index_pilot);
57             random_PAM_symbol_normalized(tem_pos:tem_pos+length(

```

```

        pilot)-1) = pilot_replace;
58     end
59     end
60
61     transmitted_symbol = [pilot_symbol
        random_PAM_symbol_normalized];
62
63 end
64
65 %%
66 function generator_import(transmitted_symbol, visaAddress,
        sample_rate, generator_period)
67 % This function import a waveform into the signal generator
68 % Input:
69 % transmitted_symbol: sample of waveform to be imported
70 % visaAddress: TekVISA address of the signal generator,
71 % sample_rate: number of sample per second
72 % generator_period: length in second of the whole waveform
73 % Output:
74
75 %     visaAddress = 'USB::0x0699::0x0356::C020110::INSTR'; % Use
        VISA 'Instrument Manager - Properties' to find the USB port
        address
76
77     timeVec = 0:1/sample_rate:generator_period;
78     timeVec = timeVec(1:end-1);
79
80     tem_1 = min(transmitted_symbol);
81     tem_2 = max(transmitted_symbol);
82     tem_4 = transmitted_symbol - (tem_2 - tem_1)/2;
83     normalize_transmitted_symbol = tem_4./max(tem_4);
84
85     waveform = normalize_transmitted_symbol; %
        transmitted_symbol is a vector (1, samples of symbol
        sequence)
86     waveform = round((waveform + 1.0)*8191); % Convert the
        double values integer values between 0 and 16382 (14 bit
        vertical resolution)
87     waveformLength = length(waveform); % Encode variable '
        waveform' into binary waveform data for AFG.
88
89     binblock = zeros(2 * waveformLength, 1);
90     binblock(2:2:end) = bitand(waveform, 255);
91     binblock(1:2:end) = bitshift(waveform, -8);
92     binblock = binblock';
93
94     bytes = num2str(length(binblock)); % Build binary block
        header

```

```

95     header = [ '#', num2str(length(bytes)) bytes ];
96
97     instrreset; % Clear MATLAB workspace of any previous
           instrument connections
98
99     myFgen = visa('TEK', visaAddress); % Create a VISA object
100
101     myFgen.OutputBufferSize = length(binblock) * 8; %
           OutputBufferSize = number of data points * datatype size
           in bytes
102
103     myFgen.ByteOrder = 'littleEndian'; % Set ByteOrder to match
           the requirement of the instrument
104
105     fopen(myFgen); % Open the connection to the function
           generator
106
107     % fprintf: write text format data, fwrite: write binary
           format data
108     fprintf(myFgen, '*RST'); % Reset the function generator to a
           know state
109     fprintf(myFgen, '*CLS;');
110
111     fprintf(myFgen, [ 'DATA:DEF_EMEM, ' num2str(length(timeVec))
           '; ']); % Resets the contents of edit memory and define
           the length of signal 1001
112     fwrite(myFgen, [ ':TRACE_EMEM, ' header binblock '; '], 'uint8
           '); % Transfer the custom waveform from MATLAB to edit
           memory of instrument
113     fprintf(myFgen, 'SOUR1:FUNC_EMEM'); % Associate the waveform
           in edit memory to channel 1
114     fprintf(myFgen, 'SOUR2:FUNC_EMEM');
115
116     fprintf(myFgen, 'SOUR1:FREQ:FIXed_25Hz'); % Set the output
           frequency to 1 Hz to avoid further upconverted by AFG
117     fprintf(myFgen, 'SOUR2:FREQ:FIXed_25Hz');
118     fprintf(myFgen, 'source1:voltage:level:immediate:low_0V');
           %Set channel 2 low level at 0V
119     fprintf(myFgen, 'source1:voltage:level:immediate:high_3V'); %
           Set channel 2 high level at 3V
120     fprintf(myFgen, 'source2:voltage:level:immediate:low_0V');
121     fprintf(myFgen, 'source2:voltage:level:immediate:high_3V');
122     % fprintf(myFgen, ':OUTP1 ON'); % Turn on Channel 1 output
123
124     fclose(myFgen); % Clean up – close the connection and clear
           the object
125     clear myFgen;
126

```

127 **end**

A.2 Code lấy tín hiệu từ oscilloscope

```

1  %TEST2 Code for communicating with an instrument.
2  %
3  %   This is the machine generated representation of an
   instrument control
4  %   session using a device object. The instrument control
   session comprises
5  %   all the steps you are likely to take when communicating with
   your
6  %   instrument. These steps are:
7  %
8  %       1. Create a device object
9  %       2. Connect to the instrument
10 %       3. Configure properties
11 %       4. Invoke functions
12 %       5. Disconnect from the instrument
13 %
14 %   To run the instrument control session, type the name of the
   file,
15 %   TEST2, at the MATLAB command prompt.
16 %
17 %   The file, TEST2.M must be on your MATLAB PATH. For
   additional information
18 %   on setting your MATLAB PATH, type 'help addpath' at the
   MATLAB command
19 %   prompt.
20 %
21 %   Example:
22 %       TEST2;
23 %
24 %   See also ICDEVICE.
25 %
26
27 %   Creation time: 11-Apr-2019 11:39:48
28 for i=10000:10001
29
30 % Create a VISA-USB object.
31 % check the USB address using VISA program: click on VISA icon
   -> Instrument manager
32 interfaceObj = instrfind('Type', 'visa-usb', 'RsrcName', 'USB::0
   x0699::0x03A6::C032080::INSTR', 'Tag', '');
33
34 % interfaceObj = instrfind('Type', 'visa-usb', 'RsrcName', 'USB
   ::0x0699::0x0356::C020110::INSTR', 'Tag', '');
35

```

```

36
37 % Create the VISA-USB object if it does not exist
38 % otherwise use the object that was found.
39 if isempty(interfaceObj)
40 %     interfaceObj = visa('TEK', 'USB::0x0699::0x0356::C020110::
        INSTR');
41
42     interfaceObj = visa('TEK', 'USB::0x0699::0x03A6::C032080::
        INSTR');
43
44 else
45     fclose(interfaceObj);
46     interfaceObj = interfaceObj(1);
47 end
48
49 % Create a device object.
50 deviceObj = icdevice('tektronix_tds2024.mdd', interfaceObj);
51
52 % Connect device object to hardware.
53 connect(deviceObj);
54
55 % Execute device object function(s).
56 groupObj = get(deviceObj, 'Waveform');
57 [X,Y,YUNIT,XUNIT] = invoke(groupObj, 'readwaveform', 'Channel1')
    ;
58
59 % Delete objects.
60 delete([deviceObj interfaceObj]);
61
62 %file_name = sprintf('oscilloscope_%s', datestr(now, 'mm-dd-
        yyyy_HH-MM'));
63 file_name = sprintf('./dataset/oscilloscope_%s', str);
64 save(file_name, 'X', 'Y', 'YUNIT', 'XUNIT');
65
66 sampling= repmat(X,1,1);
67 to_column= sampling(:);
68 filename = ['Test', num2str(i), '.csv'];
69 % csvwrite(filename, to_column);
70
71 end

```

A.3 Code trích tín hiệu phát theo tín hiệu thu

```

1 oscilloscope_file_name = sprintf('./dataset/oscilloscope_%s.mat',
    ,str) ; % received signal from oscilloscope in Matlab format
2 load(oscilloscope_file_name); % import: symbol_rate,
    transmitted_symbol
3 time_vector = Y.';

```



```

4 voltage_vector = X.'; % extract voltage values
5 no_of_transmitted_symbol = length(transmitted_symbol);
6 number_of_symbol = round(symbol_rate * (max(time_vector)-min(
    time_vector))); % number of symbols inside received vector
7 sample_per_symbol = round(size(time_vector, 1) /
    number_of_symbol); % number of samples per symbol
8 eq = number_of_symbol*sample_per_symbol;
9 x=1:1:2500;
10 xq = 1:2500/eq:2500;
11 voltage_vector = interp1(x,voltage_vector',xq)';
12 transmitted_vector = kron(transmitted_symbol, ones(1,
    sample_per_symbol))'; % create transmitted vector with the
    same samples per symbol
13 transmitted_vector = (transmitted_vector-1)*-1;
14 transmitted_vector_length = size(transmitted_vector,1);
15 received_vector_length = size(voltage_vector,1);
16 length_diff = received_vector_length - transmitted_vector_length
    ; % received vector is at least equal to transmitted vector
17 correlation_value = zeros(length_diff+1,1);
18 for index_shifting = 1 : length_diff+1
19     shifted_received_vector = voltage_vector(index_shifting :
        index_shifting + transmitted_vector_length -1);
20     tem_1 = corrcoef(transmitted_vector, shifted_received_vector
        );
21     correlation_value(index_shifting, 1) = tem_1(1,2);
22 end
23 [~, sequence_index] = max(correlation_value); % index of the
    first symbol in the transmitted sequence inside the recieved
    vector
24 % extracting received symbol, save as a maxtrix
    extracted_received_symbol, where each column is a symbol
25 extracted_received_sequence = voltage_vector(sequence_index :
    sequence_index + no_of_transmitted_symbol * sample_per_symbol
    - 1);
26 extracted_received_symbol = reshape(extracted_received_sequence ,
    [sample_per_symbol , no_of_transmitted_symbol]);
27 file_name = sprintf('./dataset/extracted_recieved_symbol_%s',str
    );
28 save(file_name);

```

A.4 Code tính ước lượng SNR, BER

```

1 recieved_file_name = sprintf('./dataset/
    extracted_recieved_symbol_40000_500_1.mat');
2 load(recieved_file_name, 'extracted_received_symbol');
3 x1=extracted_received_symbol;
4 transmitted_file_name = sprintf('./dataset/
    Transmitted_PAM_data_40000_500_1.mat');

```

```

5  load(transmitted_file_name,'transmitted_symbol');
6  a1=transmitted_symbol;
7  low = x1(:, a1 == 1);
8  high = x1(:, a1 == 0 );
9  [row,~] = size(low);
10 nlow = reshape(low,[1 row*length(low)]);
11 nhigh = reshape(high,[1 row*length(high)]);
12 ber=1/2*erfc(sqrt(SNR/2))
13 x=mean(extracted_received_symbol);
14 a=(x-min(x))/(max(x)-min(x));
15 plot(a,'LineWidth',2,...
16 'MarkerSize',10,...
17 'MarkerEdgeColor','b')
18 hold on
19 %plot((transmitted_symbol-1)*-1)

```

A.5 Code tính BER theo PNN

```

1  datafull = readtable('datafull40000_500.csv');
2  datafull=table2array(datafull);
3  data = datafull(:,2:end);
4  target = datafull(:,1);
5  target=target'+1;
6  data=data';
7  notrain=0.8*size(data,2);
8  pos=randperm(size(data,2));
9  dtrain=data(:,pos(1:notrain));
10 dtest = data(:,pos(notrain+1:end));
11 ttrain = target(:,pos(1:notrain));
12 ttest = target(:,pos(notrain+1:end));
13 ttrainc = ind2vec(ttrain);
14 net = newpnn(dtrain,ttrainc,0.01);
15 Y = sim(net,dtest);
16 ttestc = vec2ind(Y);
17 er = ttestc - ttest;
18 so_er = length(find(er~=0));
19 ber=so_er/length(ttestc)

```