

BỘ GIÁO DỤC VÀ ĐÀO TẠO

BỘ GIAO THÔNG VẬN TẢI

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP.HCM

----- oOo -----

PHẠM VĂN TÂM

**DÙNG MẠNG RBF ĐIỀU KHIỂN
CÁNH TAY ROBOT CẮT MẪU**

CHUYÊN NGÀNH: TỰ ĐỘNG HÓA

MÃ SỐ: 605260

LUẬN VĂN THẠC SĨ KỸ THUẬT

NGƯỜI HƯỚNG DẪN KHOA HỌC:

TS. Nguyễn Thiện Thành

1.099

TRƯỜNG ĐẠI HỌC GTVT TP.HCM
THƯ VIỆN

H.S. TĐHòa 01-12

TP. HCM 06 - 2012

LỜI CAM ĐOAN

Sau thời gian làm việc và nỗ lực của bản thân cùng với sự hướng dẫn tận tình của Thầy TS. Nguyễn Thiện Thành, luận văn đã hoàn thành kế hoạch đề ra.

Luận văn sử dụng bộ điều khiển mô hình nội dùng mạng RBF để điều khiển cánh tay robot cắt mẫu. Kết quả cho thấy đúng theo yêu cầu đề ra.

Tôi xin cam đoan trong quá trình thực hiện luận văn đã làm việc nghiêm túc. Tất cả các mẫu hình, bảng biểu, tài liệu được trích và tham khảo theo đúng qui chế, qui định.

TP.HCM, ngày 08 tháng 06 năm 2012

Người viết cam đoan

Phạm Văn Tâm

LỜI CẢM ƠN

Xin chân thành cảm ơn thầy TS. NGUYỄN THIỆN THÀNH, người đã tận tình hướng dẫn, giúp đỡ tôi trong quá trình thực hiện luận văn này.

Xin cảm ơn quý thầy cô trong Khoa Điện trường đại học Giao Thông Vận Tải TP.HCM, phòng quản lí khoa học đã giúp đỡ tôi trong quá trình học tập và thực hiện luận văn.

Cuối cùng, xin chân thành cảm ơn đến gia đình, các bạn bè thân thiết đã hỗ trợ, động viên và giúp đỡ tôi rất nhiều trong suốt quá trình thực hiện đề tài này.

Đại học Giao Thông Vận Tải Tp. HCM

Người thực hiện

Phạm Văn Tâm

MỤC LỤC

Trang phụ bìa
Lời cam đoan
Lời cảm ơn
Mục lục
Danh mục chữ viết tắt
Danh mục hình vẽ
Lời mở đầu

	Trang
CHƯƠNG I: GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI	
1.1 Đặt vấn đề	1
1.2 Giải quyết vấn đề	2
1.2.1 Điều khiển trực tiếp dùng mạng RBF	2
1.2.2 Điều khiển mô hình nội dùng mạng RBF	2
1.2.3 Các vấn đề cần giải quyết	3
CHƯƠNG II: CƠ SỞ LÝ THUYẾT	
2.1 Lý thuyết mạng nơron.....	4
2.1.1. Mạng neuron nhân tạo	4
2.1.2 Một số loại mạng neuron thông dụng.....	6
2.1.2.1 Mạng Perceptron 1 lớp.....	6
2.1.2.2 Mạng RBF	10
2.2 Ứng dụng mạng RBF trong nhận dạng và điều khiển	14
2.2.1 Nhận dạng dùng mạng RBF	14
2.2.2 Điều khiển dùng mạng RBF	17
2.2.2.1 Điều khiển trực tiếp dùng mạng RBF	17
2.2.2.2 Điều khiển mô hình nội dùng mạng RBF	19

CHƯƠNG III: KHẢO SÁT ĐỐI TƯỢNG CÁNH TAY ROBOT 2 KHỚP NỐI

3.1 Khảo sát mô hình toán động lực học của cánh tay robot 2 khớp nối.....	20
3.2 Khảo sát đáp ứng bước của đối tượng	23
3.2.1 Sơ đồ simulink	23
3.2.2 Đáp ứng bước của đối tượng	23

CHƯƠNG IV: THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN CÁNH TAY ROBOT 2 KHỚP NỐI DÙNG MẠNG RBF

4.1 Thiết kế hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD.....	25
4.1.1 Sơ đồ khối.....	25
4.1.2 Cấu trúc bộ điều khiển neuron PD dùng mạng RBF	25
4.1.3 Huấn luyện mạng RBF	26
4.1.4 Mô phỏng bộ điều khiển neural PD theo mô hình toán.....	30
4.2 Thiết kế mạng RBF nhận dạng mô hình cánh tay robot 2 khớp nối	33
4.2.1 Cấu trúc mạng.....	33
4.2.2 Huấn luyện mạng RBF	33
4.2.3 Mô phỏng mạng nhận dạng	37
4.3 Thiết kế hệ thống điều khiển mô hình nội cánh tay Robot 2 khớp nối dùng mạng RBF	39
4.3.1 Sơ đồ khối.....	39
4.3.2 Mô phỏng bộ điều khiển mô hình nội theo mô hình toán	40

CHƯƠNG V: KẾT QUẢ MÔ PHỎNG

5.1 Giao diện vẽ tín hiệu đặt	46
5.2 Kết quả mô phỏng của hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD.....	47
5.2.1 Sơ đồ simulink	47
5.2.2 Kết quả mô phỏng.....	47
5.3 Kết quả mô phỏng của hệ thống điều khiển mô hình nội cánh tay robot 2 khớp nối dùng mạng RBF	52

5.3.1 Sơ đồ simulink	52
5.3.2 Kết quả mô phỏng khi không có nhiễu tác động ở ngõ ra	52
5.3.3 Kết quả mô phỏng khi có nhiễu tác động ở ngõ ra.....	63

CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

6.1 Kết luận.....	66
6.2 Hướng phát triển đề tài	66

TÀI LIỆU THAM KHẢO

PHỤ LỤC

DANH MỤC CHỮ VIẾT TẮT

Adaline: *Adaptic linear element*

LGU: *Linear graded unit*

LTU: *Linear threshold unit*

PD: *Proportional derivative*

PI: *Proportional integral*

PID: *Proportional integral derivative*

RBF: *Radial Basis Function Networks*

DANH MỤC HÌNH VẼ

Trang

+ Hình 1.1 Sơ đồ khối hệ thống điều khiển trực tiếp dùng mạng RBF	2
+ Hình 1.2 Sơ đồ khối hệ thống điều khiển mô hình nội hệ phi tuyến dùng mạng RBF	2
+ Hình 2.1 Cấu trúc của 1 neuron.....	4
+ Hình 2.2 Cấu trúc của 1 neuron nhân tạo	4
+ Hình 2.3 Cấu trúc mạng Perceptron 1 lớp	7
+ Hình 2.4 Cấu trúc mạng Adaline.....	7
+ Hình 2.5 Cấu trúc mạng RBF	10
+ Hình 2.6 Cấu trúc mạng nhận dạng RBF	15
+ Hình 2.7 Sơ đồ simulink thu thập dữ liệu.....	16
+ Hình 2.8 Kết quả nhận dạng cổng Not	16
+ Hình 2.9 Điều khiển trực tiếp dùng mạng neuron	17
+ Hình 2.10 Hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD	18
+ Hình 2.11 Sơ đồ khối hệ thống điều khiển mô hình nội dùng mạng RBF	19
+ Hình 3.1 Mô tả cánh tay robot Scara.....	20
+ Hình 3.2 Cánh tay robot 2 khớp nối	20
+ Hình 3.3 Sơ đồ simulink mô phỏng cánh tay robot 2 khớp nối	23
+ Hình 3.4 Đáp ứng ngõ ra θ_1 với tín hiệu đặt M1, M2 là hàm bước.....	23
+ Hình 3.5 Đáp ứng ngõ ra θ_2 với tín hiệu đặt M1, M2 là hàm bước.....	24
+ Hình 3.6 Tọa độ đầu công tác với tín hiệu đặt M1, M2 là hàm bước	24
+ Hình 4.1 Hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD	25
+ Hình 4.2 Cấu trúc mạng điều khiển RBF	25
+ Hình 4.3 Sơ đồ simulink bộ điều khiển Neuron PD.....	30
+ Hình 4.4 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là hình sin.....	31

+ Hình 4.5 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là hình sin.....	31
+ Hình 4.6 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông.....	31
+ Hình 4.7 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông.....	32
+ Hình 4.8 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa.....	32
+ Hình 4.9 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa.....	32
+ Hình 4.10 Cấu trúc mạng nhận dạng RBF	33
+ Hình 4.11 Sơ đồ thu thập data	37
+ Hình 4.12 Sơ đồ simulink mạng nhận dạng	37
+ Hình 4.13 Kết quả nhận dạng	38
+ Hình 4.14 Hệ thống điều khiển mô hình nội dùng mạng RBF.....	39
+ Hình 4.15 Sơ đồ simulink bộ điều khiển mô hình nội.....	40
+ Hình 4.16 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là hàm bước.....	41
+ Hình 4.17 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là hàm bước.....	41
+ Hình 4.18 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình sin.....	41
+ Hình 4.19 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình sin.....	42
+ Hình 4.20 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình sin khi tăng tần số	42
+ Hình 4.21 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình sin khi tăng tần số	42
+ Hình 4.22 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông.....	43
+ Hình 4.23 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông.....	43
+ Hình 4.24 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông khi tăng tần số	43
+ Hình 4.25 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông khi tăng tần số	44
+ Hình 4.26 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa.....	44
+ Hình 4.27 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa.....	44
+ Hình 4.28 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa khi tăng tần số	45
+ Hình 4.29 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa khi tăng tần số	45
+ Hình 5.1 Giao diện vẽ tín hiệu đặt.....	46
+ Hình 5.2 Sơ đồ simulink hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neuron PD	47

+ Hình 5.3 Tín hiệu đặt là mẫu cắt hình tròn.....	47
+ Hình 5.4 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là mẫu cắt hình tròn.....	48
+ Hình 5.5 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là mẫu cắt hình tròn.....	48
+ Hình 5.6 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt là mẫu cắt hình tròn..	48
+ Hình 5.7 Tín hiệu đặt là mẫu cắt hình vuông	49
+ Hình 5.8 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là mẫu cắt hình vuông	49
+ Hình 5.9 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là mẫu cắt hình vuông	49
+ Hình 5.10 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt là mẫu cắt hình vuông	50
+ Hình 5.11 Tín hiệu đặt là mẫu cắt hình tam giác	50
+ Hình 5.12 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là mẫu cắt hình tam giác	50
+ Hình 5.13 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là mẫu cắt hình tam giác	51
+ Hình 5.14 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt là mẫu cắt hình tam giác.....	51
+ Hình 5.15 Sơ đồ simulink hệ thống điều khiển mô hình nội cánh tay robot 2 khớp nối	52
+ Hình 5.16 Tín hiệu đặt là mẫu cắt hình tròn.....	53
+ Hình 5.17 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn	53
+ Hình 5.18 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn	53
+ Hình 5.19 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn	54
+ Hình 5.20 Tín hiệu đặt là mẫu cắt hình tam giác	54
+ Hình 5.21 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tam giác.....	54
+ Hình 5.22 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tam giác.....	55
+ Hình 5.23 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tam giác.....	55
+ Hình 5.24 Tín hiệu đặt là mẫu cắt hình chữ nhật	55
+ Hình 5.25 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình chữ nhật.....	56
+ Hình 5.26 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình chữ nhật.....	56
+ Hình 5.27 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình chữ nhật.....	56

+ Hình 5.28 Tín hiệu đặt là mẫu cắt 3 hình tam giác	57
+ Hình 5.29 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tam giác	57
+ Hình 5.30 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tam giác	57
+ Hình 5.31 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tam giác	58
+ Hình 5.32 Tín hiệu đặt là mẫu cắt hình tròn	58
+ Hình 5.33 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn	59
+ Hình 5.34 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn	59
+ Hình 5.35 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn	59
+ Hình 5.36 Tín hiệu đặt là mẫu cắt hình tam giác	60
+ Hình 5.37 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tam giác	60
+ Hình 5.38 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tam giác	60
+ Hình 5.39 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tam giác	61
+ Hình 5.40 Tín hiệu đặt là mẫu cắt hình vuông	61
+ Hình 5.41 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình vuông	61
+ Hình 5.42 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình vuông	62
+ Hình 5.43 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình vuông	62
+ Hình 5.44 Tín hiệu đặt là mẫu cắt hình tròn	63
+ Hình 5.45 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn	63
+ Hình 5.46 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn	63
+ Hình 5.47 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn	64
+ Hình 5.48 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn	64
+ Hình 5.49 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn	64
+ Hình 5.50 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn	65

LỜI MỞ ĐẦU

Ngày nay trong lĩnh vực công nghiệp, robot Scara được sử dụng rất rộng rãi như robot cắt, robot hàn, robot gấp vật.... Tuy nhiên, mô hình toán của robot khá phức tạp. Để điều khiển đối tượng này thì các bộ điều khiển thông thường không đáp ứng được, vì thế các bộ điều khiển thông minh xuất hiện như điều khiển mờ, mạng neural, giải thuật di truyền. . . .

Ưu điểm cơ bản của điều khiển dùng mạng neuron so với điều khiển kinh điển là chỉ cần dữ liệu vào ra của đối tượng là có thể điều khiển được đối tượng đó mà không cần biết mô hình toán của đối tượng. Chính vì lý do đó mà các bộ điều khiển dùng mạng neuron có thể áp dụng để giải nhiều bài toán điều khiển có độ phức tạp cao, có nhiều yếu tố bất định.

Thấy được ưu điểm đó, vận dụng kiến thức đã học, tôi áp dụng để thiết kế hệ thống điều khiển cánh tay robot cắt mẫu dùng mạng RBF. Đề tài chia làm 4 phần chính sau:

- Xây dựng mô hình toán cánh tay robot 2 khớp nối
- Thiết kế bộ điều khiển trực tiếp dùng mạng RBF điều khiển cánh tay robot cắt mẫu
- Xây dựng giải thuật huấn luyện mạng RBF nhận dạng mô hình toán của cánh tay robot
- Thiết kế bộ điều khiển mô hình nội dùng mạng RBF điều khiển cánh tay robot cắt mẫu

CHƯƠNG I:

GIỚI THIỆU TỔNG QUAN VỀ ĐỀ TÀI

1.1 Đặt vấn đề

Sự phát triển không ngừng của khoa học công nghệ làm xuất hiện các đối tượng điều khiển có độ phức tạp ngày càng tăng. Yêu cầu thực tiễn đặt ra là phải điều khiển các hệ thống động ngày càng phức tạp, trong điều kiện các yếu tố bất định ngày càng gia tăng với yêu cầu chất lượng điều khiển ngày càng cao. Các yêu cầu trên không thể được đáp ứng một cách trọn vẹn, đồng thời nếu chỉ dùng các lý thuyết điều khiển thông thường sẵn có. Đây chính là động lực cho sự ra đời của hàng loạt các lý thuyết điều khiển hiện đại hứa hẹn một hướng giải quyết triệt để các bài toán điều khiển phi tuyến phức tạp.

Trong những năm gần đây, mạng thần kinh nhân tạo đã được áp dụng thành công vào những lĩnh vực kỹ thuật như giao thông, robot, thị giác máy tính, tay máy... Nhiều sơ đồ điều khiển dùng mạng thần kinh với thuật toán lan truyền ngược được ứng dụng để giải các bài toán điều khiển các hệ phi tuyến phức tạp và bất ổn định. Tính thích nghi cho phép mạng thần kinh vẫn thực hiện tốt chức năng của nó khi môi trường và đối tượng điều khiển thay đổi theo thời gian bằng cách cập nhật cấu trúc cũng như các trọng số của mạng. Có rất nhiều thuật toán đã được phát triển để huấn luyện mạng thần kinh với những ưu và khuyết điểm riêng. Thuật toán suy giảm độ dốc là một trong những thuật toán đơn giản và thường dùng nhất để cập nhật các trọng số của mạng thần kinh.

Điều khiển tự động hóa là lĩnh vực rất quan trọng trong các ngành công nghiệp, đặc biệt trong ngành chế biến thực phẩm, hóa học, dầu khí,... Việc điều khiển và giám sát chính xác là rất cần thiết và quan trọng, và nó có ảnh hưởng đến chất lượng của sản phẩm. Hiện nay đa số các máy cắt gạch, cắt sắt đều do con người trực tiếp sử dụng máy cắt nên dễ dàng xảy ra tai nạn lao động nếu bất cẩn.

Vấn đề đặt ra là thiết kế bộ điều khiển để điều khiển cánh tay robot cắt theo tín hiệu đặt (đường thẳng, tam giác, hình vuông, gấp khúc, đường tròn). Hệ cánh tay robot là hệ phi tuyến do đó bộ điều khiển phải có khả năng tự chỉnh định thông số

để thích nghi với sự thay đổi của môi trường, thích nghi với nhiễu tác động bên ngoài.

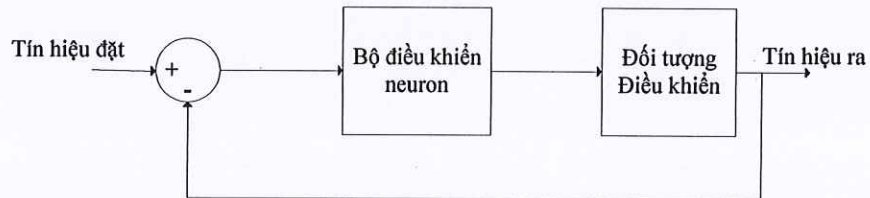
Với những lý do trên, tôi chọn đề tài “Dùng mạng RBF điều khiển cánh tay robot cắt mẫu”.

1.2 Giải quyết vấn đề

Có nhiều cách để điều khiển cánh tay robot như dùng bộ điều khiển mờ, điều khiển thích nghi, điều khiển dùng mạng neuron.....

Hai bộ điều khiển dùng mạng RBF khá phổ biến là điều khiển trực tiếp dùng mạng RBF và điều khiển mô hình nội dùng mạng RBF.

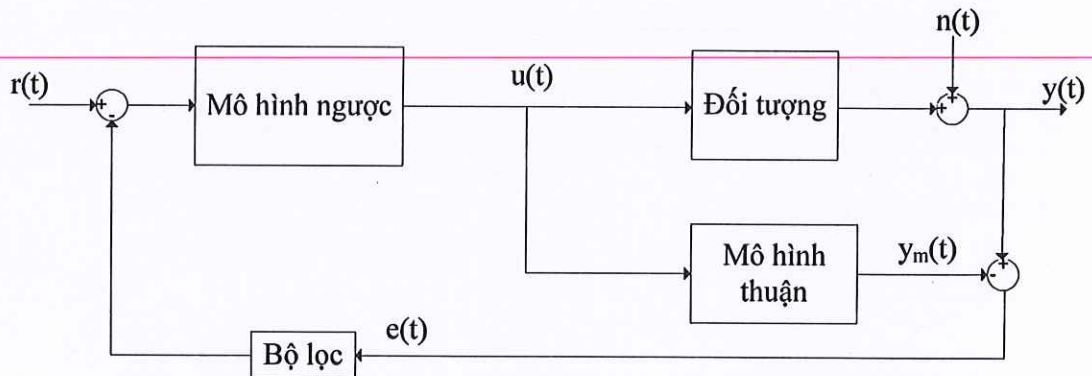
1.2.1 Điều khiển trực tiếp dùng mạng RBF



Hình 1.1 Sơ đồ khối hệ thống điều khiển trực tiếp dùng mạng RBF

Mạng thần kinh được sử dụng để học kinh nghiệm của chuyên gia. Sau khi huấn luyện, mạng thần kinh có thể điều khiển các đối tượng thay thế con người.

1.2.2 Điều khiển mô hình nội dùng mạng RBF



Hình 1.2 Sơ đồ khối hệ thống điều khiển mô hình nội hệ phi tuyến dùng mạng RBF

Hình 1.2 được đưa ra dựa trên sơ đồ điều khiển mô hình nội hệ tuyến tính. Khi mô hình mô tả chính xác đặc tính động học của hệ thống và không có nhiễu

thì tín hiệu hồi tiếp bằng 0, khi đó hệ thống trên trở thành hệ thống điều khiển vòng hở. Nếu có nhiễu $n(t)$ ảnh hưởng đến quá trình điều khiển thì tín hiệu phản hồi sẽ bằng $n(t)$ và không gây ra ảnh hưởng đến tác động điều khiển, nhiễu $n(t)$ bị trừ khỏi tín hiệu chuẩn. Do đó ảnh hưởng của nhiễu đo lường ở ngõ ra của quá trình điều khiển hoàn toàn bị triệt tiêu. Bộ lọc được sử dụng để tăng độ bền vững của hệ thống đối với sai số mô hình ở miền tần số cao và nhiễu đo lường.

1.2.3 Các vấn đề cần giải quyết

- Xây dựng mô hình toán học của cánh tay robot hai khớp nối
- Thiết kế bộ điều khiển trực tiếp dùng mạng RBF điều khiển cánh tay robot cắt mẫu
- Xây dựng giải thuật huấn luyện mạng RBF nhận dạng mô hình toán của cánh tay robot
- Thiết kế bộ điều khiển mô hình nội dùng mạng RBF điều khiển cánh tay robot cắt mẫu
- Mô phỏng hệ thống đã thiết kế.
- Cho kết quả mô phỏng

CHƯƠNG II:

CƠ SỞ LÝ THUYẾT

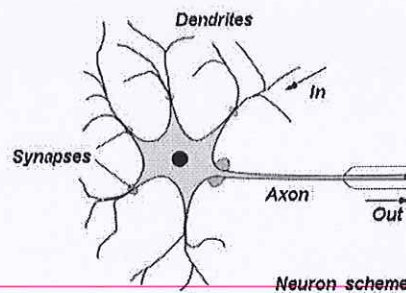
2.1 Lý thuyết mạng nơron

2.1.1. Mạng neuron nhân tạo

Mạng neuron nhân tạo mô phỏng hoạt động của não người để giải quyết các bài toán kỹ thuật. Bộ não người có khoảng 10^{10} neuron. Các neuron này được kết hợp với nhau thành mạng. Việc xử lý thông tin được thực hiện nhờ vào sự lan truyền của tín hiệu từ neuron này sang neuron khác thông qua các sợi trục thần kinh (axon). Mạng neuron nhân tạo được đặc trưng bởi 3 yếu tố:

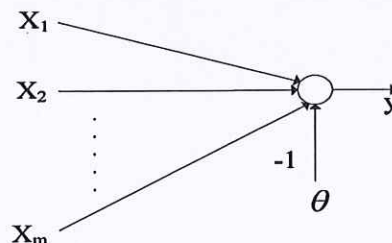
- + Phần tử xử lý (neuron)
- + Cấu trúc và ghép nối của các phần tử xử lý
- + Phương pháp huấn luyện mạng

Hình 2.1 trình bày cấu trúc của 1 neuron



Hình 2.1 Cấu trúc của 1 neuron

Hình 2.2 trình bày cấu trúc của 1 neuron nhân tạo



Hình 2.2 Cấu trúc của 1 neuron nhân tạo

Mỗi neuron nhân tạo được đặc trưng bởi quan hệ giữa các tín hiệu vào X_1, X_2, \dots, X_m và tín hiệu ra y

$$\text{net} = f(X_1, X_2, \dots, X_m) - \theta$$

$$y = a(\text{net})$$

Trong đó:

θ : Ngưỡng tác động của neuron

$a(\cdot)$: Hàm tác động

$f(\cdot)$: Hàm tích hợp

Hàm tích hợp: Kết hợp các thông tin nhận được ở tín hiệu vào. Ta có 1 số hàm tích hợp thông dụng như sau:

+ Hàm tích hợp tuyến tính:

$$\text{net} = f(x) - \theta = \sum_{j=1}^m w_j x_j - \theta = w^T x$$

Với $x = [x_1, x_2, \dots, x_m, -1]^T$ và $w = [w_1, w_2, \dots, w_m, \theta]^T$

+ Hàm tích hợp dạng toàn phương

$$\text{net} = f(x) - \theta = \sum_{j=1}^m w_j x_j^2 - \theta$$

Hàm tác động: Là hàm kích hoạt, hàm truyền tạo tín hiệu ra $y = a(\text{net})$.

Ta có 1 số hàm tác động thông dụng như sau:

+ Hàm tác động tuyến tính:

$$a(\text{net}) = \text{net}$$

+ Bước nhảy đơn vị:

$$a(\text{net}) = \begin{cases} 1 & \text{if } \text{net} > 0 \\ 0 & \text{if } \text{net} < 0 \end{cases}$$

+ Hàm dấu:

$$a(\text{net}) = \begin{cases} 1 & \text{if } \text{net} > 0 \\ -1 & \text{if } \text{net} < 0 \end{cases}$$

+ Hàm dốc:

$$a(net) = \begin{cases} 0 & \text{if } net < 0 \\ net & \text{if } 0 \leq net \leq 1 \\ 1 & \text{if } net > 1 \end{cases}$$

+ Hàm logsig:

$$a(net) = \frac{1}{1 + e^{-\lambda net}}$$

+ Hàm tansig:

$$a(net) = \frac{2}{1 + e^{-\lambda net}} - 1$$

Huấn luyện mạng:

Mục đích của huấn luyện mạng là xác định các bộ trọng số của các neuron trong mạng từ các dữ liệu. Nếu quá trình huấn luyện có sử dụng tín hiệu ra mong muốn d , ta nói các neuron học có giám sát. Nếu quá trình huấn luyện không sử dụng tín hiệu ra mong muốn, ta nói các neuron học không giám sát. Nếu quá trình huấn luyện không sử dụng tín hiệu ra mong muốn nhưng có sử dụng 1 tín hiệu đánh giá chất lượng của mạng, ta nói quá trình học của mạng là quá trình học tăng cường.

Luật học các trọng số thường có dạng sau:

$$w(k+1) = w(k) + \eta r(k)x(k)$$

Trong đó $\eta > 0$: hằng số học

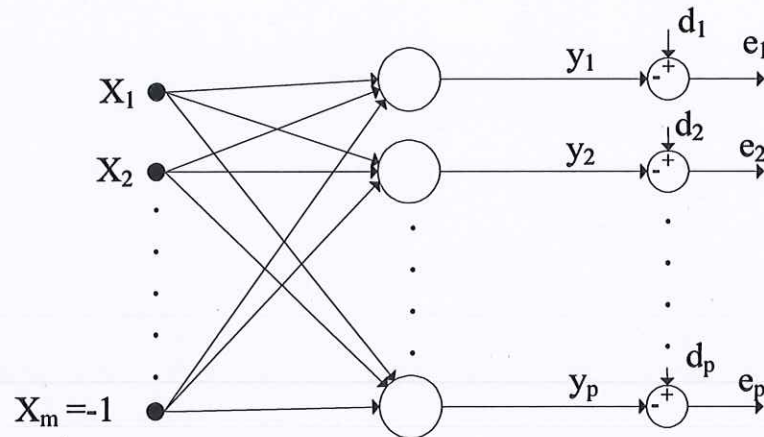
$r(k)$: tín hiệu học

2.1.2 Một số loại mạng neuron thông dụng

2.1.2.1 Mạng Perceptron 1 lớp:

+ Perceptron là neuron với hàm tích hợp tuyến tính

+ Hình 2.3 trình bày mạng perceptron 1 lớp với $m-1$ tín hiệu vào x_j và p tín hiệu ra y_i ($x_m = -1$ tương ứng với ngưỡng tác động w_{im} của neuron)



Hình 2.3 Cấu trúc mạng Perceptron 1 lớp

$$+ y_i = a \left(\sum_{j=1}^m w_{ij} x_j \right) = a(w_i^T x)$$

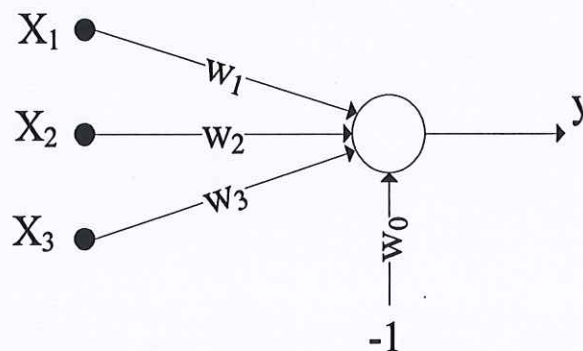
+ Mục tiêu: Huấn luyện mạng dùng N mẫu dữ liệu $\{x(k), d(k), k = 1, 2, \dots, N\}$ sao cho $y(k) = d(k), k = 1, 2, \dots, N$.

+ Một số loại Perceptron:

- **Adaline (Adaptive linear element):**

Là 1 neuron với hàm tác động tuyến tính và hàm tích hợp tuyến tính

Ví dụ: Cho 1 Adaline với 3 tín hiệu vào, 1 tín hiệu ra như hình 2.4



Hình 2.4 Cấu trúc mạng Adaline

$$\text{net} = w_1 X_1 + w_2 X_2 + w_3 X_3 - w_0 = X^T w \tag{2.1}$$

Với:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_0 \end{bmatrix}; X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ -1 \end{bmatrix}$$

$$y = \text{net} = X^T w$$

Huấn luyện mạng: Học có giám sát

○ Hàm mục tiêu: $J = \frac{1}{2}(d - y)^2$ (2.2)

Với:

d: Tín hiệu ra mong muốn

y: Tín hiệu ra của mạng

- Xác định w sao cho J cực tiểu: Dùng phương pháp Steepert descent

$$w(k+1) = w(k) - \eta \nabla J$$

Với:

$$\nabla J = \left(\frac{\partial J}{\partial w} \right)^T = \left(\frac{\partial J}{\partial y} \frac{\partial y}{\partial w} \right)^T = -(d - y)x$$

$$\Rightarrow w(k+1) = w(k) + \eta(d - y)x$$
 (2.3)

- **LGU (Linear graded unit):** là 1 neuron với hàm tác động là hàm sigmoid và hàm tích hợp tuyến tính

Ví dụ: Cho 1 LGU với 3 tín hiệu vào, 1 tín hiệu ra, hàm tác động tansig như hình 2.4.

$$\text{net} = w_1 X_1 + w_2 X_2 + w_3 X_3 - w_0 = X^T w$$
 (2.4)

Với:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_0 \end{bmatrix}; X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ -1 \end{bmatrix}$$

$$y = \frac{2}{1 + e^{-\lambda net}} - 1, \lambda > 0 \quad (2.5)$$

Huấn luyện mạng: Học có giám sát

○ Hàm mục tiêu: $J = \frac{1}{2}(d - y)^2$ (2.6)

○ Luật cập nhật trọng số w:

$$w(k+1) = w(k) - \eta \nabla J$$

Với:

$$\begin{aligned} \nabla J &= \left(\frac{\partial J}{\partial w} \right)^T = \left(\frac{\partial J}{\partial y} \frac{\partial y}{\partial net} \frac{\partial net}{\partial w} \right)^T \\ &= \left(-(d - y) \frac{\lambda}{2} (1 - y^2) x^T \right)^T \\ \Rightarrow w(k+1) &= w(k) + \eta (d - y) \frac{\lambda}{2} (1 - y^2) x \end{aligned} \quad (2.7)$$

- **LTU (Linear threshold unit):** là 1 neuron với hàm tác động là hàm dấu và hàm tích hợp tuyến tính

Ví dụ: Cho 1 LTU với 3 tín hiệu vào, 1 tín hiệu ra như hình 2.4.

$$net = w_1 X_1 + w_2 X_2 + w_3 X_3 - w_0 = X^T w \quad (2.8)$$

Với:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_0 \end{bmatrix}; X = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ -1 \end{bmatrix}$$

$$y = \text{sign}(net) = \begin{cases} 1 & \text{if } net \geq 0 \\ -1 & \text{if } net < 0 \end{cases}$$

Huấn luyện mạng: Học có giám sát

○ Hàm mục tiêu: $J = \frac{1}{2}(d - y)^2$ (2.9)

○ Luật cập nhật trọng số w:

$$w(k+1) = w(k) + \eta (d - y) x \quad (2.10)$$

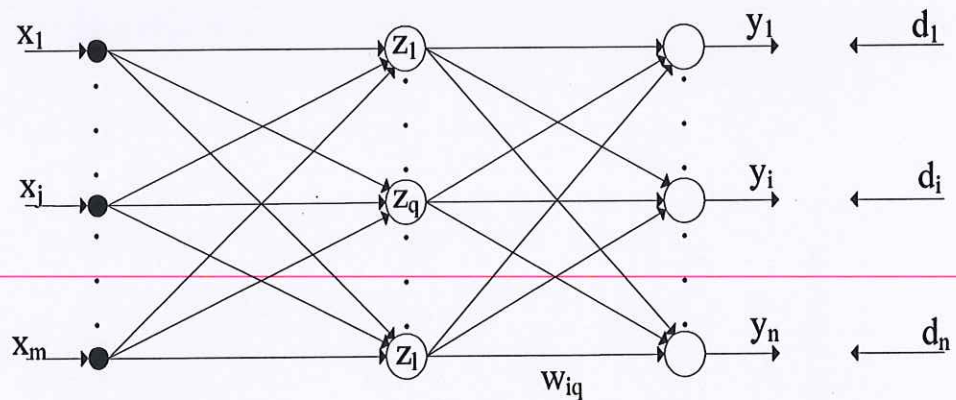
2.1.2.2 Mạng RBF: (Radial Basis Function Networks)

Mạng RBF(Mạng xuyên tâm) là 1 trường hợp đặc biệt của mạng truyền thẳng 2 lớp, trong đó:

- + Hàm tổng ngõ vào của tế bào thần kinh ở lớp ẩn là hàm cầu
- + Hàm kích hoạt của tế bào thần kinh ở lớp ẩn là hàm mũ: $a_0(f) = e^{-f}$
- + Hàm tổng ngõ vào của tế bào thần kinh ở lớp ra là hàm tuyến tính
- + Hàm kích hoạt của tế bào thần kinh ở lớp ra là tuyến tính

Mạng RBF là loại mạng thường dùng trong nhận dạng và điều khiển. Có khả năng xấp xỉ hàm rất tốt và có thể huấn luyện dễ dàng, nhanh chóng. Khả năng xấp xỉ chính xác và tốc độ của mạng RBF còn có thể cải thiện hơn nữa bằng việc chọn tâm, hệ số trải phù hợp cho các hàm cơ sở ở lớp ẩn.

Mạng có cấu trúc như hình 2.5:



Hình 2.5 Cấu trúc mạng RBF

Trong đó: $x = [x_1 \dots x_m]^T$ là vector ngõ vào

z_q là hàm xuyên tâm tạo tín hiệu ra của đơn vị q ở lớp ẩn

$w = [w_{iq}](i = 1 \div n, q = 1 \div l)$ là ma trận trọng số kết nối giữa lớp ẩn và lớp ra

$y = [y_1 \dots y_n]$ là vector ngõ ra thực sự của mạng

$d = [d_1 \dots d_n]$ là ngõ ra mong muốn của mạng

Ngõ ra của tế bào thần kinh ở lớp ẩn là hàm mũ. Do đó vector tín hiệu vào càng gần tâm của hàm cơ sở thì ngõ ra của tế bào thần kinh ở lớp ẩn càng lớn và ngược lại. Ngõ ra của mạng $y_i = f(z_q)$ nên chỉ những vector tín hiệu vào nằm trong lân cận quanh tâm của hàm cơ sở mới ảnh hưởng đến ngõ ra của mạng. Vì vậy hàm cơ sở xuyên tâm có tính cục bộ.

Huấn luyện mạng RBF:

Cho mạng RBF có cấu trúc như hình 2.5

Trường hợp 1:

Ngõ ra của các tế bào thần kinh ở lớp ẩn được cho như sau:

$$z_q(k) = e^{-\frac{\|x(k)-m_q\|^2}{\sigma_q^2}}; \quad q = 1 \div l \tag{2.11}$$

$$+ \|x - m_q\| = \sqrt{(x - m_q)^T (x - m_q)}$$

+ $m_q = [m_{q1}, \dots, m_{qn}]^T$ là vector điểm xuyên tâm của đơn vị q ở lớp ẩn

+ σ_q là khổ rộng của hàm Gauss (hệ số trái)

Ngõ ra của mạng được tính như sau:

$$y_i(k) = \sum_{q=1}^l w_{iq}(k)z_q(k) + \theta_i; \quad i = 1 \div n \tag{2.12}$$

Với θ_i là đơn vị ngưỡng

Hàm mục tiêu:

$$J = \frac{1}{2} \sum_{i=1}^n (d_i(k) - y_i(k))^2 \tag{2.13}$$

Luật học cập nhật trọng số, tâm, hệ số trái:

- o Cho hàm chi phí đo sai số giữa tín hiệu ra mong muốn và tín hiệu ra thực sự của mạng là:

$$J = \frac{1}{2} \sum_{i=1}^n (d_i(k) - y_i(k))^2 \tag{2.14}$$

- Luật học cập nhật trọng số w_{iq} :

$$\Delta w_{iq} = -\eta_1 \frac{\partial J}{\partial w_{iq}} = -\eta_1 \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial w_{iq}} = \eta_1 (d_i - y_i) z_q \quad (2.15)$$

Phương trình cập nhật trọng số được thiết lập:

$$w_{iq}(k+1) = w_{iq}(k) + \Delta w_{iq} \quad (2.16)$$

- Luật học cập nhật tâm m_q :

$$\begin{aligned} \Delta m_q &= -\eta_2 \frac{\partial J}{\partial m_q} = -\eta_2 \left(\sum_{i=1}^n \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial z_q} \right) \frac{\partial z_q}{\partial m_q} \\ &= \eta_2 \left(\sum_{i=1}^n (d_i - y_i) w_{iq} \right) 2 \frac{\|x - m_q\|}{\sigma_q^2} z_q \end{aligned} \quad (2.17)$$

Phương trình cập nhật tâm:

$$m_q(k+1) = m_q(k) + \Delta m_q \quad (2.18)$$

- Luật cập nhật hệ số trái:

$$\begin{aligned} \Delta \sigma_q &= -\eta_3 \frac{\partial J}{\partial \sigma_q} = -\eta_3 \left(\sum_{i=1}^n \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial z_q} \right) \frac{\partial z_q}{\partial \sigma_q} \\ &= \eta_3 \left(\sum_{i=1}^n (d_i - y_i) w_{iq} \right) 2 \frac{\|x - m_q\|^2}{\sigma_q^3} z_q \end{aligned} \quad (2.19)$$

Phương trình cập nhật hệ số trái:

$$\sigma_q(k+1) = \sigma_q(k) + \Delta \sigma_q \quad (2.20)$$

Trường hợp 2:

Ngõ ra của các tế bào thần kinh ở lớp ẩn được cho như sau:

$$z_q(k) = e^{-\frac{\|x(k) - m_q\|^2}{2\sigma_q}}; \quad q = 1 \div l \quad (2.21)$$

Ngõ ra của mạng được tính như sau:

$$y_i(k) = \sum_{q=1}^l w_{iq}(k) z_q(k) + \theta_i; \quad i = 1 \div n \quad (2.22)$$

Với θ_i là đơn vị ngưỡng

Hàm mục tiêu:

$$J = \frac{1}{2} \sum_{i=1}^n (d_i(k) - y_i(k))^2 \quad (2.23)$$

Luật học cập trọng số, tâm, hệ số trái:

- Cho hàm chi phí đo sai số giữa tín hiệu ra mong muốn và tín hiệu ra thực sự của mạng là:

$$J = \frac{1}{2} \sum_{i=1}^n (d_i(k) - y_i(k))^2 \quad (2.24)$$

- Luật học cập nhật trọng số w_{iq} :

$$\Delta w_{iq} = -\eta_1 \frac{\partial J}{\partial w_{iq}} = -\eta_1 \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial w_{iq}} = \eta_1 (d_i - y_i) z_q \quad (2.25)$$

Phương trình cập nhật trọng số được thiết lập:

$$w_{iq}(k+1) = w_{iq}(k) + \Delta w_{iq} \quad (2.26)$$

- Luật học cập nhật tâm m_q :

$$\begin{aligned} \Delta m_q &= -\eta_2 \frac{\partial J}{\partial m_q} = -\eta_2 \left(\sum_{i=1}^n \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial z_q} \right) \frac{\partial z_q}{\partial m_q} \\ &= \eta_2 \left(\sum_{i=1}^n (d_i - y_i) w_{iq} \right) \frac{\|x - m_q\|}{\sigma_q} z_q \end{aligned} \quad (2.27)$$

Phương trình cập nhật tâm:

$$m_q(k+1) = m_q(k) + \Delta m_q \quad (2.28)$$

- Luật cập nhật hệ số trái:

$$\begin{aligned} \Delta \sigma_q &= -\eta_3 \frac{\partial J}{\partial \sigma_q} = -\eta_3 \left(\sum_{i=1}^n \frac{\partial J}{\partial y_i} \frac{\partial y_i}{\partial z_q} \right) \frac{\partial z_q}{\partial \sigma_q} \\ &= \eta_3 \left(\sum_{i=1}^n (d_i - y_i) w_{iq} \right) \frac{\|x - m_q\|^2}{2\sigma_q^2} z_q \end{aligned} \quad (2.29)$$

Phương trình cập nhật hệ số trái:

$$\sigma_q(k+1) = \sigma_q(k) + \Delta \sigma_q \quad (2.30)$$

2.2 Ứng dụng mạng RBF trong nhận dạng và điều khiển

2.2.1 Nhận dạng dùng mạng RBF

Nếu biết trước rất ít (thậm chí không biết) thông tin về đối tượng ta phải giải bài toán nhận dạng hệ thống chủ yếu dựa vào dữ liệu vào – ra thực nghiệm quan sát được. Trong trường hợp này, mạng thần kinh là cấu trúc mô hình phi tuyến được chọn lựa đầu tiên để mô hình hóa hệ thống vì mạng thần kinh có tính linh hoạt rất cao, có thể xấp xỉ hàm phi tuyến trơn với sai số nhỏ tùy ý nếu số tế bào thần kinh ở lớp ẩn đủ lớn, đồng thời trọng số của mạng có thể huấn luyện hoàn toàn dựa vào dữ liệu mà không cần thêm bất cứ thông tin gì khác về đối tượng cần nhận dạng.

Khi sử dụng mạng thần kinh để nhận dạng hệ thống cần phải chọn cấu hình mạng, cụ thể:

- + Số tế bào thần kinh ở lớp vào: phụ thuộc vào số phần tử hồi quy
- + Số tế bào thần kinh ở lớp ẩn: Phụ thuộc vào tính phi tuyến của đối tượng cần nhận dạng, đối tượng có tính phi tuyến càng cao thì cần nhiều tế bào thần kinh ở lớp ẩn. Tuy nhiên, không phải chọn càng nhiều tế bào thần kinh ở lớp ẩn càng tốt vì có thể dẫn đến hiện tượng quá khớp. Nên chọn số tế bào thần kinh ở lớp ẩn càng ít càng tốt sao cho sai lệch giữa ngõ ra của mạng và ngõ ra của hệ thống cần nhận dạng đủ nhỏ chấp nhận được

Mạng RBF khi sử dụng trong nhận dạng mô hình cần lưu ý một số ưu điểm và nhược điểm sau:

- + Ưu: Quá trình học tương đối đơn giản
- + Nhược: Khi vector đầu vào có số chiều lớn sẽ kéo theo số tâm lớn, làm khối lượng tính toán nhiều và có thể làm sai lệch quá trình nhận dạng mô hình. Ngoài ra, nếu tính ngẫu nhiên của dữ liệu tăng sẽ dẫn đến việc làm tăng số lượng tâm và như vậy phải tăng thêm chiều cho vector dữ liệu đầu vào để

đảm bảo độ chính xác nhận dạng mô hình. Khối lượng tính toán tăng lên làm cho thời gian nhận dạng bị kéo dài

Ví dụ: Cho cổng Not gồm 1 ngõ vào A và 1 ngõ ra F như sau:

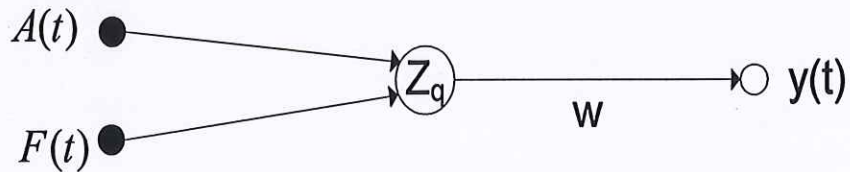
$$A \in \{0,1\}; F = \text{not}(A) = \{1,-1\}.$$

+ Phương trình toán của cổng Not như sau:

$$F = \text{sign}(\text{net}) = \begin{cases} 1, & \text{if } \text{net} > 0 \\ -1, & \text{if } \text{net} < 0 \end{cases}$$

$$\text{Chọn } \text{net} = -A + 0.5 \tag{2.31}$$

+ Cấu trúc mạng:



$$\text{Với } q = 1 \div 2$$

Hình 2.6 Cấu trúc mạng nhận dạng RBF

+ Huấn luyện mạng:

- Lớp vào: $X(t) = \{A(t), F(t)\}$
- Lớp ẩn: Đầu ra của các neuron lớp ẩn

$$Z = g(\|X - C\|) = e^{-\frac{(\|X - C\|)^2}{\sigma^2}}$$

- Lớp ra: $y(t)$

$$y = W.Z$$

- Hàm mục tiêu: $J = \frac{1}{2}(F - y)^2$

- Luật cập nhật tâm:

$$C(k+1) = C(k) + \eta_c (F - y) \cdot 2 \cdot \frac{\|X - C\|}{\sigma^2} e^{-\frac{\|X - C\|^2}{\sigma^2}} \tag{2.32}$$

- Luật cập nhật hệ số trái:

$$\sigma(k+1) = \sigma(k) + \eta_{\sigma} (F - y) \cdot 2 \cdot \frac{\|X - C\|^2}{\sigma^3} e^{-\frac{\|X - C\|^2}{\sigma^2}} \quad (2.33)$$

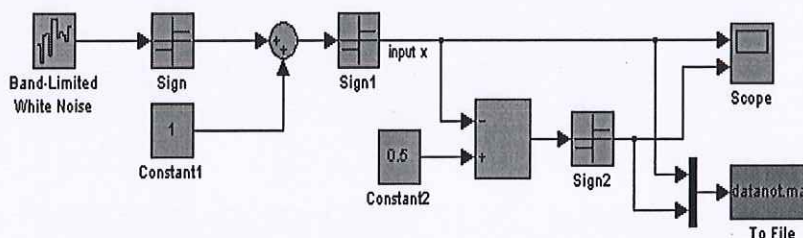
- Luật cập nhật trọng số:

$$W(k+1) = W(k) + \eta_w (F - y) Z \quad (2.34)$$

+ Mô phỏng:

- Bước 1: Thu thập dữ liệu:

o Sơ đồ simulink thu thập dữ liệu:



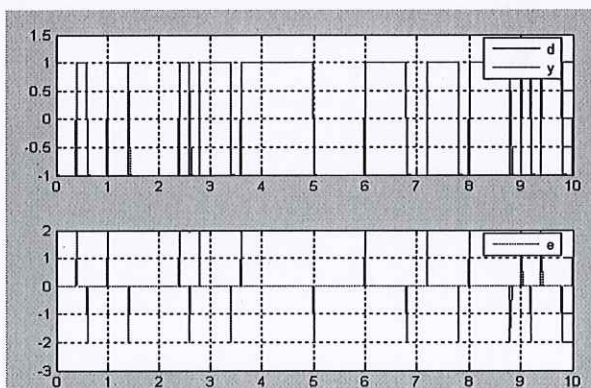
Hình 2.7 Sơ đồ simulink thu thập dữ liệu

o Sau khi thu thập dữ liệu xong ta được dữ liệu ngõ vào và ngõ ra mong muốn của mạng nhận dạng được lưu trong file datanot.mat

- Bước 2: Nhận dạng

o Huấn luyện tâm, hệ số trái, trọng số lớp ra của mạng

o Kết quả nhận dạng:



Hình 2.8 Kết quả nhận dạng cổng Not

Với: $d = F$: Ngõ ra mong muốn của mạng

y : Ngõ ra của mạng

$e = d - y$: Sai số giữa ngõ ra mong muốn và ngõ ra của mạng

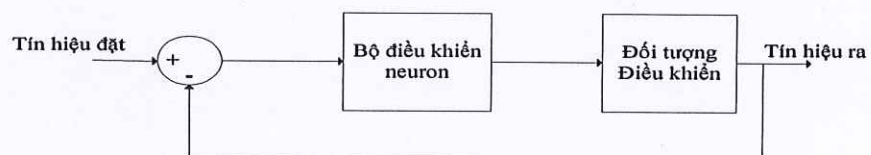
2.2.2 Điều khiển dùng mạng RBF

2.2.2.1 Điều khiển trực tiếp dùng mạng RBF

Trong công nghiệp có những đối tượng điều khiển rất phức tạp, việc thiết kế các bộ điều khiển thông thường hầu như không thể thực hiện được do không thể xác định được mô hình toán của đối tượng, hoặc mô hình toán của đối tượng quá phức tạp, có tính phi tuyến cao. Trong khi đó con người với khả năng xử lý tinh xảo của bộ não sau khi trải qua quá trình đào tạo và tích lũy kinh nghiệm lại có thể điều khiển được các đối tượng phức tạp mà không cần biết gì về mô tả toán học của đối tượng. Từ đó các nhà nghiên cứu đã đưa ra các sơ đồ điều khiển có khả năng phỏng theo khả năng xử lý dựa vào kinh nghiệm, khả năng học của con người, đó là các sơ đồ điều khiển dùng logic mờ, mạng thần kinh.

Điều khiển trực tiếp dùng mạng neuron:

Bộ điều khiển thần kinh được dùng trong đường thuận của hệ thống điều khiển hồi tiếp. Tín hiệu ra của đối tượng điều khiển được so sánh với tín hiệu đặt, nếu có sai lệch thì bộ điều khiển thần kinh sẽ xuất tín hiệu tác động vào đối tượng nhằm mục đích làm sai lệch giảm về 0. Đây là sơ đồ điều khiển rất quen thuộc, trong sơ đồ này bộ điều khiển thần kinh được dùng để thay thế bộ điều khiển kinh điển.



Hình 2.9 Điều khiển trực tiếp dùng mạng neuron

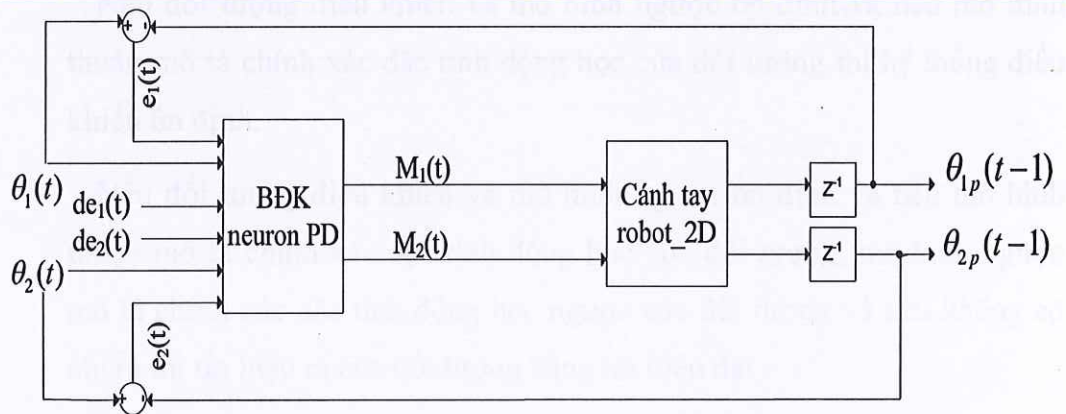
Mạng thần kinh có thể được sử dụng để học kinh nghiệm chuyên gia trong việc điều khiển các đối tượng phức tạp, khó xác định mô hình toán hoặc sao chép 1 bộ điều khiển vật lý có sẵn. Ví dụ mạng thần kinh có thể học kinh nghiệm của người lái tàu, lái xe hoặc học kinh nghiệm điều khiển máy móc thiết bị của các chuyên gia lành nghề. Sau khi huấn luyện, mạng thần kinh có thể điều khiển các đối tượng thay thế con người.

Bộ điều khiển neuron PD dùng mạng RBF:

Bộ điều khiển kinh điển PI, PD, PID chỉ không chế tín hiệu ra của đối tượng với chất lượng tốt nếu hệ thống làm việc trong miền tín hiệu nhỏ (hệ tuyến tính). Nếu đối tượng là phi tuyến thì các bộ điều khiển này không thể đảm bảo chất lượng điều khiển tại mọi điểm làm việc. Do vậy, để điều khiển các đối tượng phi tuyến ta sử dụng bộ điều khiển neuron PD, PD mờ, PI mờ, PID mờ . . .

Đối với bộ điều khiển neuron PD thì tín hiệu ra của bộ điều khiển tỉ lệ phi tuyến với tín hiệu vào và vi phân tín hiệu vào theo luật cập nhật trọng số, tâm, hệ số trải của mạng RBF.

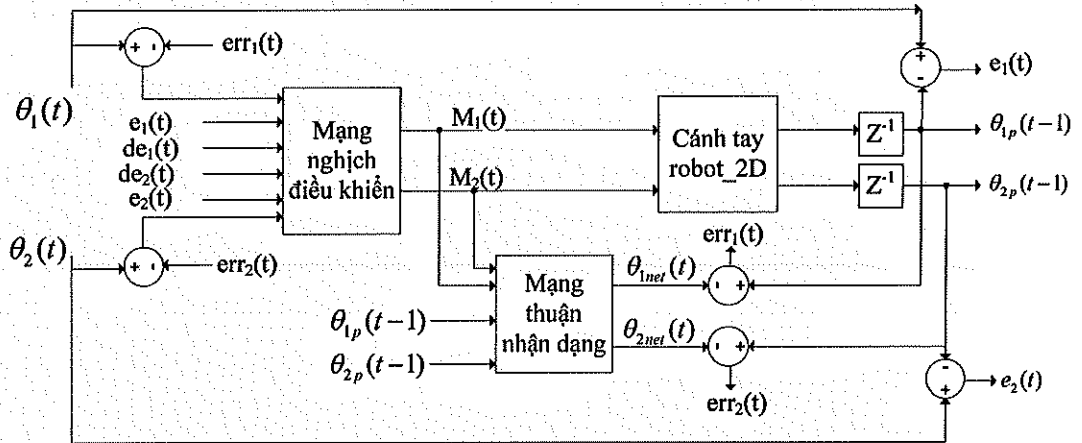
Sơ đồ khối hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD:



Hình 2.10 Hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD

2.2.2.2 Điều khiển mô hình nội dùng mạng RBF

Sơ đồ khối hệ thống điều khiển mô hình nội dùng mạng RBF như hình 2.30



Hình 2.11 Sơ đồ khối hệ thống điều khiển mô hình nội dùng mạng RBF

Trường hợp này bộ lọc bằng 1. Như vậy mạng neural điều khiển chính là mạng nhận dạng mô hình ngược của đối tượng (sử dụng bộ điều khiển neuron PD), mạng thuận chính là mạng neural nhận dạng mô hình toán của đối tượng. Khi có nhiễu tác động thì sẽ sinh ra 1 sai số và sai số này bị trừ bởi tín hiệu đặt nên không ảnh hưởng đến tác động điều khiển.

Hệ thống điều khiển mô hình nội phi tuyến có các tính chất sau:

Nếu đối tượng điều khiển và mô hình ngược ổn định và nếu mô hình thuận mô tả chính xác đặc tính động học của đối tượng thì hệ thống điều khiển ổn định.

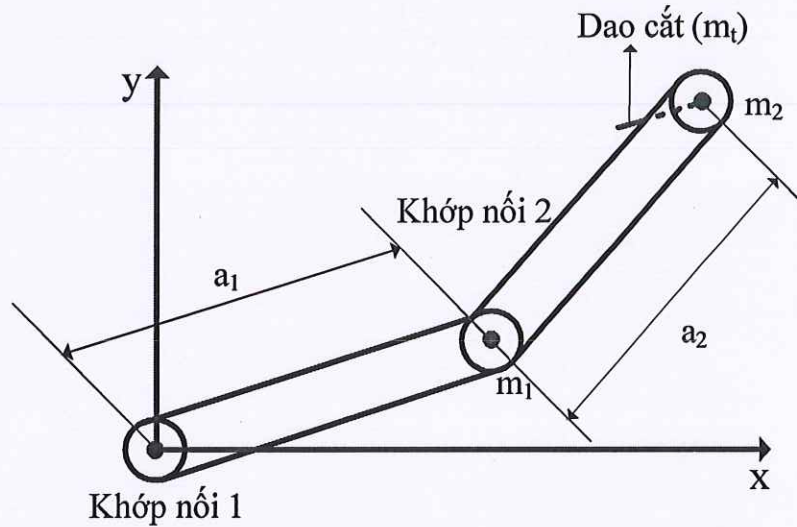
Nếu đối tượng điều khiển và mô hình ngược ổn định và nếu mô hình thuận mô tả chính xác đặc tính động học của đối tượng, mô hình ngược mô tả chính xác đặc tính động học ngược của đối tượng và nếu không có nhiễu thì tín hiệu ra của đối tượng bằng tín hiệu đặt

CHƯƠNG III:

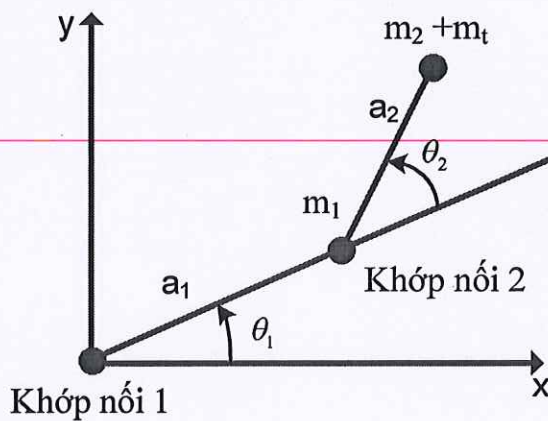
KHẢO SÁT ĐỐI TƯỢNG CÁNH TAY ROBOT 2 KHỚP NỐI

3.1 Khảo sát mô hình toán động lực học của cánh tay robot 2 khớp nối:

Cho hệ thống vật lý của cánh tay robot hai khớp nối như hình 3.1



Hình 3.1 Mô tả cánh tay robot Scara



Hình 3.2 Cánh tay robot 2 khớp nối

Phần này sẽ trình bày mô hình toán động lực học của cánh tay robot hai bậc tự do (two-degree-of-freedom). Hình 3.2 mô tả cánh tay robot Scara gồm hai trục liên kết (2 links), hai khớp nối (2 joints), hoạt động cụ thể như sau:

- + Bộ chấp hành (actuator) 1 cung cấp moment quay M_1 để điều khiển khớp nối 1 được kết nối trực tiếp với trục 1 tạo lực chuyển động chính cho toàn cánh tay.
- + Bộ chấp hành 2 cung cấp moment quay M_2 để điều khiển khớp nối 2, khớp nối này kết nối trục 2 với trục 1.
- + a_1 là khoảng cách giữa tâm của khớp nối 1 và tâm của khớp nối 2.
- + a_2 là khoảng cách giữa tâm của khớp nối 2 với tải (m_t).
- + θ_1 và θ_2 tương ứng là góc quay của bộ chấp hành 1 và 2.

Với khâu 1, ta có biểu thức động năng và thế năng tương ứng là:

$$K_1 = \frac{1}{2} m_1 a_1^2 \dot{\theta}_1^2 \quad (3.1)$$

$$P_1 = m_1 g a_1 \sin \theta_1 \quad (3.2)$$

Với khâu 2 ta có:

$$x_2 = a_1 \cos \theta_1 + a_2 \cos(\theta_1 + \theta_2) \quad (3.3)$$

$$y_2 = a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2) \quad (3.4)$$

$$\dot{x}_2 = -a_1 \dot{\theta}_1 \sin \theta_1 - a_2 (\dot{\theta}_1 + \dot{\theta}_2) \sin(\theta_1 + \theta_2) \quad (3.5)$$

$$\dot{y}_2 = a_1 \dot{\theta}_1 \cos \theta_1 + a_2 (\dot{\theta}_1 + \dot{\theta}_2) \cos(\theta_1 + \theta_2) \quad (3.6)$$

Bình phương vận tốc:

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = a_1^2 \dot{\theta}_1^2 + a_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + 2a_1 a_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 \quad (3.7)$$

Động năng của khâu 2 là:

$$K_2 = \frac{1}{2} (m_2 + m_t) v_2^2 = \frac{1}{2} (m_2 + m_t) a_1^2 \dot{\theta}_1^2 + \frac{1}{2} (m_2 + m_t) a_2^2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + (m_2 + m_t) a_1 a_2 (\dot{\theta}_1^2 + \dot{\theta}_1 \dot{\theta}_2) \cos \theta_2 \quad (3.8)$$

Thế năng của khâu 2:

$$P_2 = (m_2 + m_t) g y_2 = (m_2 + m_t) g [a_1 \sin \theta_1 + a_2 \sin(\theta_1 + \theta_2)] \quad (3.9)$$

Hàm Lagrange cho tay máy này là:

$$L = K - P = K_1 + K_2 - P_1 - P_2 = \frac{1}{2}(m_1 + m_2 + m_t)a_1^2\dot{\theta}_1^2 + \frac{1}{2}(m_2 + m_t)a_2^2(\dot{\theta}_1 + \dot{\theta}_2)^2 + (m_2 + m_t)a_1a_2(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2)\cos\theta_2 - (m_1 + m_2 + m_t)ga_1\sin\theta_1 - (m_2 + m_t)ga_2\sin(\theta_1 + \theta_2) \quad (3.10)$$

Phương trình chuyển động Lagrange thiết lập cho 1 cơ hệ:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}} - \frac{\partial L}{\partial \theta} = M \quad (3.11)$$

Ta có:

$$\frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2 + m_t)a_1^2\dot{\theta}_1 + (m_2 + m_t)a_2^2(\dot{\theta}_1 + \dot{\theta}_2) + (m_2 + m_t)a_1a_2(2\dot{\theta}_1 + \dot{\theta}_2)\cos\theta_2 \quad (3.12)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2 + m_t)a_1^2\ddot{\theta}_1 + (m_2 + m_t)a_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + (m_2 + m_t)a_1a_2(2\ddot{\theta}_1 + \ddot{\theta}_2)\cos\theta_2 - (m_2 + m_t)a_1a_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2)\sin\theta_2 \quad (3.13)$$

$$\frac{\partial L}{\partial \theta_1} = -(m_1 + m_2 + m_t)ga_1\cos\theta_1 - (m_2 + m_t)ga_2\cos(\theta_1 + \theta_2) \quad (3.14)$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = (m_2 + m_t)a_2^2(\dot{\theta}_1 + \dot{\theta}_2) + (m_2 + m_t)a_1a_2\dot{\theta}_1\cos\theta_2 \quad (3.15)$$

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_2} = (m_2 + m_t)a_2^2(\ddot{\theta}_1 + \ddot{\theta}_2) + (m_2 + m_t)a_1a_2\ddot{\theta}_1\cos\theta_2 - (m_2 + m_t)a_1a_2\dot{\theta}_1\dot{\theta}_2\sin\theta_2 \quad (3.16)$$

$$\frac{\partial L}{\partial \theta_2} = -(m_2 + m_t)a_1a_2(\dot{\theta}_1^2 + \dot{\theta}_1\dot{\theta}_2)\sin\theta_2 - (m_2 + m_t)ga_2\cos(\theta_1 + \theta_2) \quad (3.17)$$

Như vậy, phương trình chuyển động của cơ hệ tay máy được cho bởi hệ 2 phương trình vi phân:

$$M_1 = [(m_1 + m_2 + m_t)a_1^2 + (m_2 + m_t)a_2^2 + 2(m_2 + m_t)a_1a_2\cos\theta_2]\ddot{\theta}_1 + [(m_2 + m_t)a_2^2 + (m_2 + m_t)a_1a_2\cos\theta_2]\ddot{\theta}_2 - (m_2 + m_t)a_1a_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2)\sin\theta_2 + (m_1 + m_2 + m_t)ga_1\cos\theta_1 + (m_2 + m_t)ga_2\cos(\theta_1 + \theta_2) \quad (3.18)$$

$$M_2 = [(m_2 + m_t)a_2^2 + (m_2 + m_t)a_1a_2\cos\theta_2]\ddot{\theta}_1 + (m_2 + m_t)a_2^2\ddot{\theta}_2 + (m_2 + m_t)a_1a_2\dot{\theta}_1^2\sin\theta_2 + (m_2 + m_t)ga_2\cos(\theta_1 + \theta_2) \quad (3.19)$$

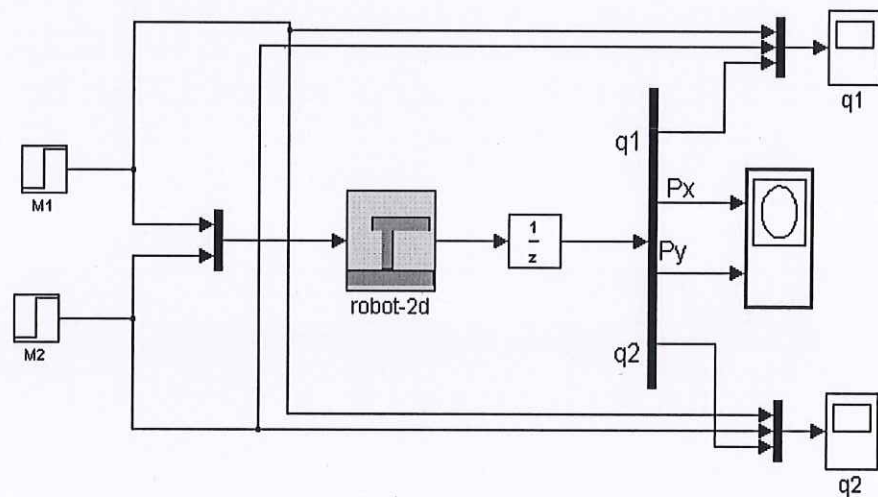
Với: M_1, M_2 là các moment phát động của các động cơ điện

Phương trình chuyển động của cơ hệ tay máy được viết dưới dạng ma trận:

$$\begin{bmatrix} (m_1 + m_2 + m_l)a_1^2 + (m_2 + m_l)a_2^2 + 2(m_2 + m_l)a_1a_2 \cos \theta_2 & (m_2 + m_l)a_2^2 + (m_2 + m_l)a_1a_2 \cos \theta_2 \\ (m_2 + m_l)a_2^2 + (m_2 + m_l)a_1a_2 \cos \theta_2 & (m_2 + m_l)a_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} -(m_2 + m_l)a_1a_2(2\dot{\theta}_1\dot{\theta}_2 + \dot{\theta}_2^2) \sin \theta_2 \\ (m_2 + m_l)a_1a_2\dot{\theta}_1^2 \sin \theta_2 \end{bmatrix} + \begin{bmatrix} (m_1 + m_2 + m_l)ga_1 \cos \theta_1 + (m_2 + m_l)ga_2 \cos(\theta_1 + \theta_2) \\ (m_2 + m_l)ga_2 \cos(\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} \quad (3.20)$$

3.2 khảo sát đáp ứng bước của đối tượng:

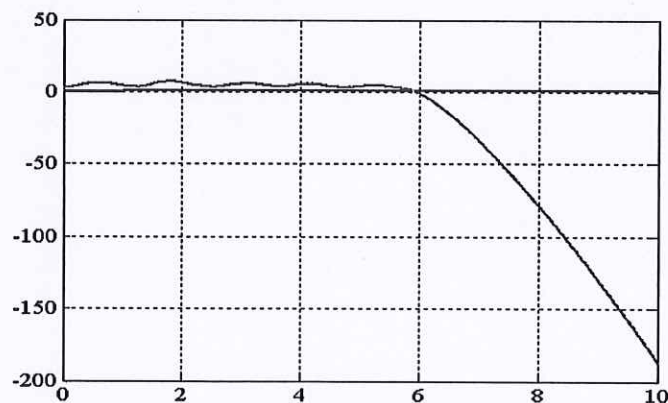
3.2.1 Sơ đồ simulink:



Hình 3.3 Sơ đồ simulink mô phỏng cánh tay robot 2 khớp nối

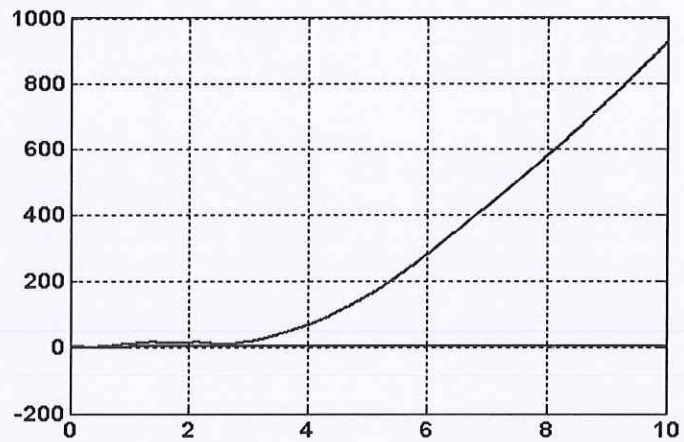
3.2.2 Đáp ứng bước của đối tượng:

+ Đáp ứng ngõ ra θ_1 với tín hiệu đặt M_1, M_2 là hàm bước:



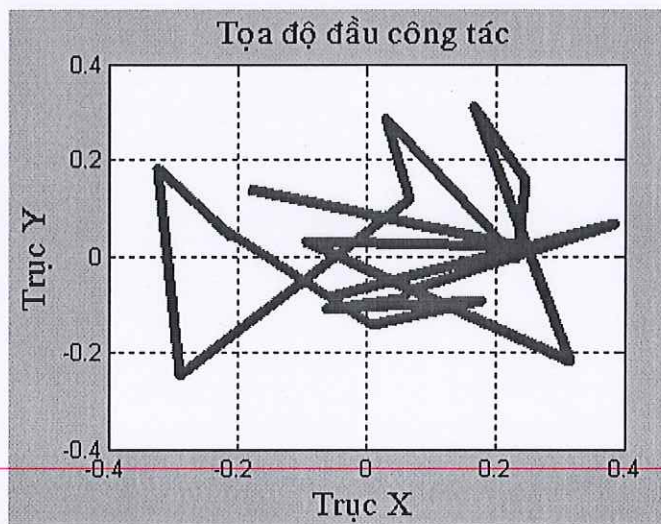
Hình 3.4 Đáp ứng ngõ ra θ_1 với tín hiệu đặt M_1, M_2 là hàm bước

+ Đáp ứng ngõ ra θ_2 với tín hiệu đặt M1, M2 là hàm bước:



Hình 3.5 Đáp ứng ngõ ra θ_2 với tín hiệu đặt M1, M2 là hàm bước

+ Tọa độ đầu công tác với tín hiệu đặt M1, M2 là hàm bước:



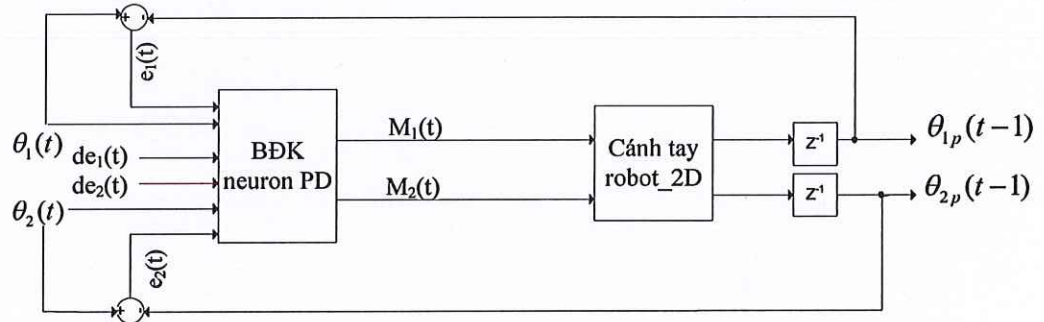
Hình 3.6 Tọa độ đầu công tác với tín hiệu đặt M1, M2 là hàm bước

CHƯƠNG IV:

THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN CÁNH TAY ROBOT 2 KHỚP NỐI DÙNG MẠNG RBF

4.1 Thiết kế hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD

4.1.1 Sơ đồ khối:

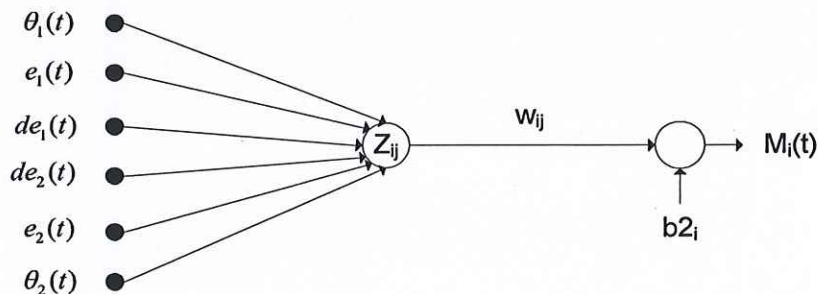


Hình 4.1 Hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neural PD

Để huấn luyện bộ điều khiển neuron PD học tập động học nghịch của cánh tay robot, trước tiên ta thu thập dữ liệu vào ra của đối tượng. Sau đó sử dụng dữ liệu đã thu thập để huấn luyện mạng thần kinh sao cho

$$J = \frac{1}{2}(e_1 + ade_1)^2 + \frac{1}{2}(e_2 + bde_2)^2 \text{ cực tiểu}$$

4.1.2 Cấu trúc bộ điều khiển neuron PD dùng mạng RBF:



Với $i = 1 \div 2, j = 1 \div 6$

Hình 4.2 Cấu trúc mạng điều khiển RBF

Mạng RBF điều khiển đối tượng gồm có:

+ 6 ngõ vào: $u(t) = \{\theta_1(t), e_1(t), de_1(t), \theta_2(t), e_2(t), de_2(t)\}$

Với:

$$e_1(t) = \theta_1(t) - \theta_{1p}(t-1)$$

$$e_2(t) = \theta_2(t) - \theta_{2p}(t-1)$$

+ 2 ngõ ra: $u_{net}(t) = \{M_1(t), M_2(t)\}$

+ 6 neuron lớp ẩn

4.1.3 Huấn luyện mạng RBF:

- Lớp vào: Có 6 ngõ vào

$$u = \{u_1, u_2, u_3, u_4, u_5, u_6\} = \{\theta_1(t), e_1(t), de_1(t), de_2(t), e_2(t), \theta_2(t)\}$$

- Lớp ẩn:

Đầu ra của các neuron lớp ẩn:

$$z_1 = z_{1j} = g(\|u - c_1\|) = e^{-\frac{\|u - c_1\|^2}{\sigma_1^2}} \quad (4.1)$$

$$z_2 = z_{2j} = g(\|u - c_2\|) = e^{-\frac{\|u - c_2\|^2}{\sigma_2^2}} \quad (4.2)$$

$$z = [z_1 \quad z_2]$$

Với:

$u = [u_1 \quad u_2 \quad u_3 \quad u_4 \quad u_5 \quad u_6]$ là 6 ngõ vào

$$c = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\ c_{21} & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \text{ là 12 tâm}$$

$$\sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} & \sigma_{15} & \sigma_{16} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} & \sigma_{24} & \sigma_{25} & \sigma_{26} \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ \sigma_2 \end{bmatrix} \text{ là 12 hệ số trải}$$

$$z = [z_1 \quad z_2] = \begin{bmatrix} z_{11} & z_{12} & z_{13} & z_{14} & z_{15} & z_{16} \\ z_{21} & z_{22} & z_{23} & z_{24} & z_{25} & z_{26} \end{bmatrix}^T \text{ là 12 ngõ ra của neuron lớp ẩn}$$

- **Lớp ra:** Gồm 2 ngõ ra

$$M_1 = w_{11}z_{11} + w_{12}z_{12} + w_{13}z_{13} + w_{14}z_{14} + w_{15}z_{15} + w_{16}z_{16} + b2_1 = w_1z_1 + b2_1 \quad (4.3)$$

$$M_2 = w_{21}z_{21} + w_{22}z_{22} + w_{23}z_{23} + w_{24}z_{24} + w_{25}z_{25} + w_{26}z_{26} + b2_2 = w_2z_2 + b2_2 \quad (4.4)$$

Vậy:

$$u_{net} = \begin{bmatrix} M_1 \\ M_2 \end{bmatrix} = w.z + b_2 \quad (4.5)$$

Với:

$$w_1 = [w_{11} \ w_{12} \ w_{13} \ w_{14} \ w_{15} \ w_{16}]$$

$$w_2 = [w_{21} \ w_{22} \ w_{23} \ w_{24} \ w_{25} \ w_{26}]$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} & w_{26} \end{bmatrix}$$

$$z = \begin{bmatrix} z_{11} & z_{21} \\ z_{12} & z_{22} \\ z_{13} & z_{23} \\ z_{14} & z_{24} \\ z_{15} & z_{25} \\ z_{16} & z_{26} \end{bmatrix}$$

$$b_2 = \begin{bmatrix} b2_1 \\ b2_2 \end{bmatrix}$$

- **Giải thuật huấn luyện:**

Áp dụng phương pháp Steepert descent để tìm tâm, hệ số trái và các trọng số bằng quá trình tối thiểu hóa sai số bình phương.

+ **Hàm mục tiêu:**

$$J = \frac{1}{2}(e_1 + 0.3de_1)^2 + \frac{1}{2}(e_2 + 0.25de_2)^2 \quad (4.6)$$

+ **Cập nhật tâm:**

Luật cập nhật tâm c_1 :

$$c_1(k+1) = c_1(k) - \eta_c \nabla_{c_1} J$$

Với:

$$\nabla_{c_1} J = \left(\frac{\partial J}{\partial c_1} \right) = \left(\frac{\partial J}{\partial \theta_{1p}} \cdot \frac{\partial \theta_{1p}}{\partial M_1} \cdot \frac{\partial M_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial c_1} \right)$$

Do bộ điều khiển neuron PD chính là mạng neuron nhận dạng mô hình nghịch của đối tượng nên sai số được lan truyền trực tiếp sang mô hình mạng neuron của bộ điều khiển. Vì vậy ta không xét sai số của đối tượng $\frac{\partial \theta_{1p}}{\partial M_1}$.

$$\text{Vậy } \nabla_{c_1} J = \left(\frac{\partial J}{\partial c_1} \right) = - \left((e_1 + 0.3de_1) w_1 \cdot 2 \cdot \frac{\|u - c_1\|}{\sigma_1^2} e^{-\frac{\|u - c_1\|^2}{\sigma_1^2}} \right)$$

Vậy luật cập nhật tâm c_1 được viết lại như sau:

$$c_1(k+1) = c_1(k) + \eta_c \left((e_1 + 0.3de_1) w_1 \cdot 2 \cdot \frac{\|u - c_1\|}{\sigma_1^2} e^{-\frac{\|u - c_1\|^2}{\sigma_1^2}} \right) \quad (4.7)$$

Luật cập nhật tâm c_2 như sau:

$$c_2(k+1) = c_2(k) + \eta_c \left((e_2 + 0.25de_2) w_2 \cdot 2 \cdot \frac{\|u - c_2\|}{\sigma_2^2} e^{-\frac{\|u - c_2\|^2}{\sigma_2^2}} \right) \quad (4.8)$$

Vậy luật cập nhật tâm:

$$c(k+1) = \begin{bmatrix} c_1(k+1) \\ c_2(k+1) \end{bmatrix} \quad (4.9)$$

+ Cập nhật hệ số trái:

Luật cập nhật hệ số trái σ_1 :

$$\sigma_1(k+1) = \sigma_1(k) - \eta_\sigma \nabla_{\sigma_1} J$$

Với:

$$\begin{aligned} \nabla_{\sigma_1} J &= \left(\frac{\partial J}{\partial \sigma_1} \right) = \left(\frac{\partial J}{\partial \theta_{1p}} \cdot \frac{\partial \theta_{1p}}{\partial M_1} \cdot \frac{\partial M_1}{\partial z_1} \cdot \frac{\partial z_1}{\partial \sigma_1} \right) \\ &= - \left((e_1 + 0.3de_1) w_1 \cdot 2 \cdot \frac{\|u - c_1\|^2}{\sigma_1^3} e^{-\frac{\|u - c_1\|^2}{\sigma_1^2}} \right) \end{aligned}$$

Vậy luật cập nhật hệ số trái σ_1 được viết lại như sau:

$$\sigma_1(k+1) = \sigma_1(k) + \eta_\sigma \left((e_1 + 0.3de_1) w_1 \cdot 2 \cdot \frac{\|u - c_1\|^2}{\sigma_1^3} e^{-\frac{\|u - c_1\|^2}{\sigma_1^2}} \right) \quad (4.10)$$

Luật cập nhật hệ số trái σ_2 :

$$\sigma_2(k+1) = \sigma_2(k) + \eta_\sigma \left((e_2 + 0.25de_2) w_2 \cdot 2 \cdot \frac{\|u - c_2\|^2}{\sigma_2^3} e^{-\frac{\|u - c_2\|^2}{\sigma_2^2}} \right) \quad (4.11)$$

Vậy luật cập nhật hệ số trái:

$$\sigma(k+1) = \begin{bmatrix} \sigma_1(k+1) \\ \sigma_2(k+1) \end{bmatrix} \quad (4.12)$$

+ Cập nhật trọng số:

Luật cập nhật trọng số

$$w_1(k+1) = w_1(k) - \eta_w \nabla_{w_1} J$$

$$w_2(k+1) = w_2(k) - \eta_w \nabla_{w_2} J$$

Với:

$$\nabla_{w_1} J = \left(\frac{\partial J}{\partial w_1} \right) = \left(\frac{\partial J}{\partial \theta_{1p}} \cdot \frac{\partial \theta_{1p}}{\partial M_1} \cdot \frac{\partial M_1}{\partial w_1} \right) = -((e_1 + 0.3de_1) z_1)$$

$$\nabla_{w_2} J = \left(\frac{\partial J}{\partial w_2} \right) = \left(\frac{\partial J}{\partial \theta_{2p}} \cdot \frac{\partial \theta_{2p}}{\partial M_2} \cdot \frac{\partial M_2}{\partial w_2} \right) = -((e_2 + 0.25de_2) z_2)$$

Vậy luật cập nhật trọng số w_1, w_2 được viết lại như sau:

$$w_1(k+1) = w_1(k) + \eta_w ((e_1 + 0.3de_1) z_1) \quad (4.13)$$

$$w_2(k+1) = w_2(k) + \eta_w ((e_2 + 0.25de_2) z_2) \quad (4.14)$$

Luật cập nhật trọng số :

$$w(k+1) = \begin{bmatrix} w_1(k+1) \\ w_2(k+1) \end{bmatrix} \quad (4.15)$$

+ Cập nhật ngưỡng(b2):

Luật cập nhật ngưỡng b_{2_1} :

$$b_{2_1}(k+1) = b_{2_1}(k) - \eta_{b_2} \nabla_{b_{2_1}} J$$

$$\nabla_{b_{2_1}} J = \left(\frac{\partial J}{\partial b_{2_1}} \right) = \left(\frac{\partial J}{\partial \theta_{1_p}} \cdot \frac{\partial \theta_{1_p}}{\partial M_1} \cdot \frac{\partial M_1}{\partial b_{2_1}} \right) = -(e_1 + 0.3de_1)$$

Vậy luật cập nhật ngưỡng b_{2_1} :

$$b_{2_1}(k+1) = b_{2_1}(k) + \eta_{b_2}(e_1 + 0.3de_1) \tag{4.16}$$

Luật cập nhật ngưỡng b_{2_2} :

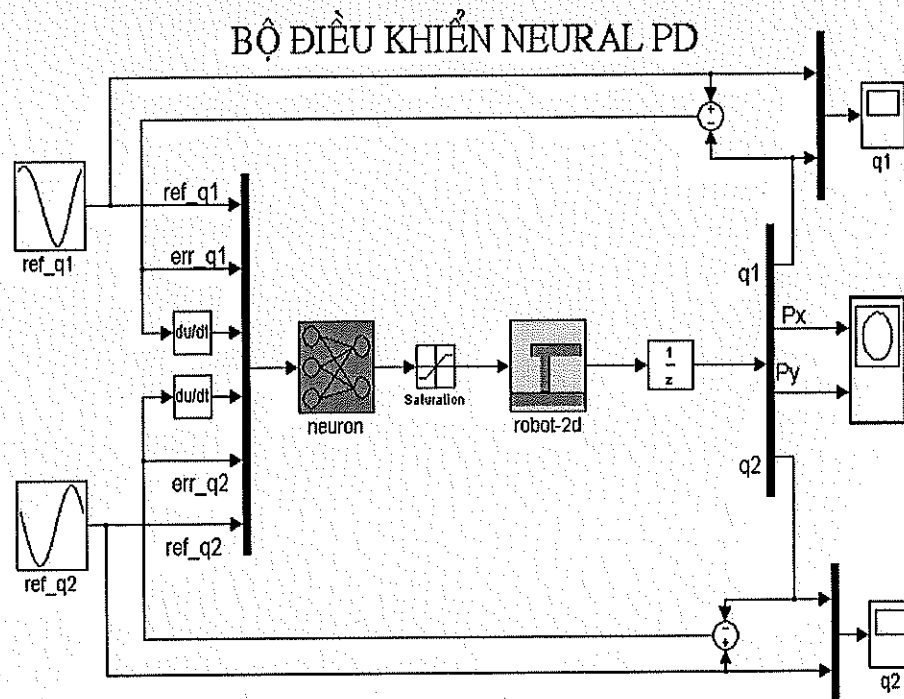
$$b_{2_2}(k+1) = b_{2_2}(k) + \eta_{b_2}(e_2 + 0.25de_2) \tag{4.17}$$

Luật cập nhật ngưỡng b_2 :

$$b_2(k+1) = \begin{bmatrix} b_{2_1}(k+1) \\ b_{2_2}(k+1) \end{bmatrix} \tag{4.18}$$

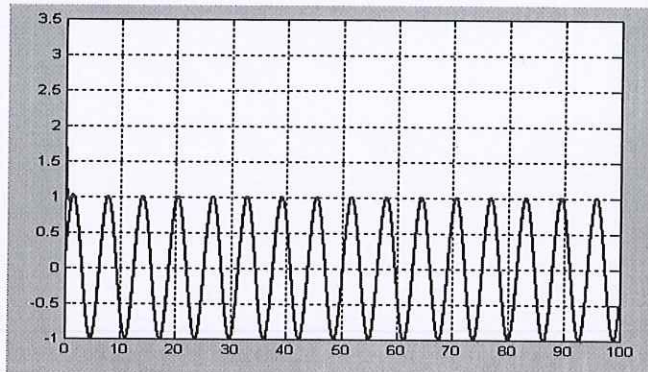
4.1.4 Mô phỏng bộ điều khiển neural PD theo mô hình toán:

- Sơ đồ simulink:



Hình 4.3 Sơ đồ simulink bộ điều khiển Neuron PD

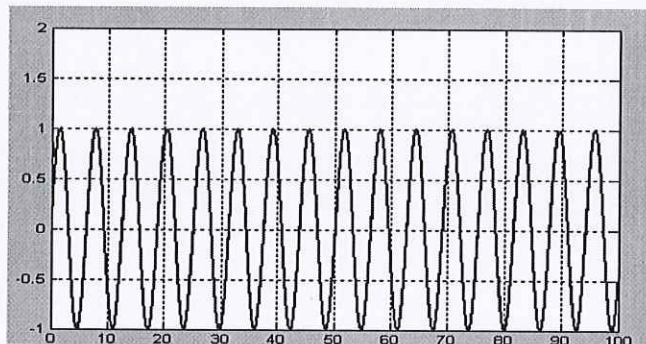
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là hình sin:



Hình 4.4 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là hình sin

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ.

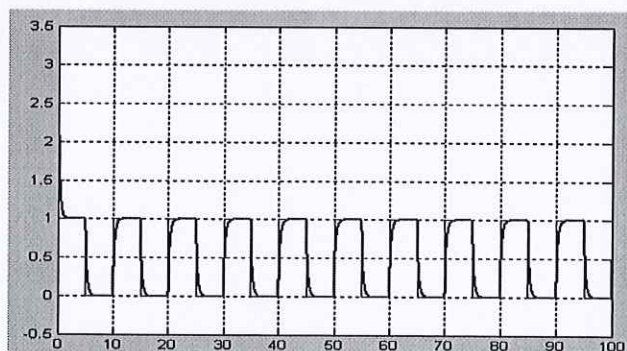
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là hình sin:



Hình 4.5 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là hình sin

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số nhỏ.

- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông:



Hình 4.6 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ

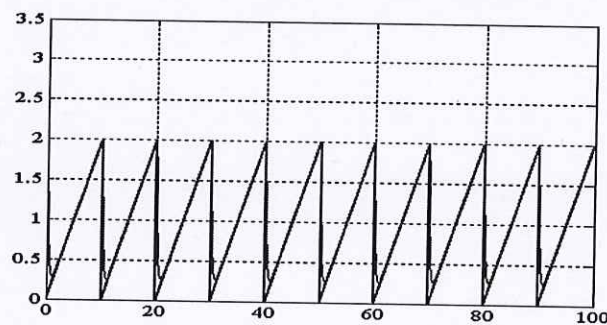
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông:



Hình 4.7 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số nhỏ

- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa:



Hình 4.8 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa

Nhận xét: Đáp ứng ngõ ra θ_1 bị sai số ở các đỉnh của tín hiệu đặt

- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa:

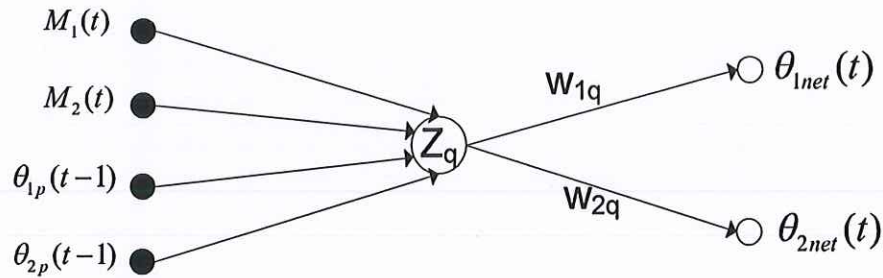


Hình 4.9 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa

Nhận xét: Đáp ứng ngõ ra θ_2 bị sai số ở các đỉnh của tín hiệu đặt

4.2 Thiết kế mạng RBF nhận dạng mô hình cánh tay robot 2 khớp nối

4.2.1 Cấu trúc mạng:



Với $q = 1 \div 4$

Hình 4.10 Cấu trúc mạng nhận dạng RBF

Mạng RBF nhận dạng đối tượng gồm có:

- + 4 ngõ vào: $x(t) = \{M_1(t), M_2(t), \theta_{1p}(t-1), \theta_{2p}(t-1)\}$
- + 2 ngõ ra: $y_{net}(t) = \{\theta_{1net}(t), \theta_{2net}(t)\}$
- + 4 neuron lớp ẩn, 4 tâm

4.2.2 Huấn luyện mạng RBF:

- Lớp vào: Có 4 ngõ vào

$$X = \{X_1, X_2, X_3, X_4\} = \{M_1, M_2, \theta_{1p}, \theta_{2p}\}$$

- Lớp ẩn:

Đầu ra của các neuron lớp ẩn:

$$Z = g(\|X - C\|) = e^{-\frac{\|X - C\|^2}{\sigma^2}}$$

Với:

$X = [X_1 \ X_2 \ X_3 \ X_4]$ là 4 ngõ vào

$C = [C_1 \ C_2 \ C_3 \ C_4]$ là 4 tâm

$\sigma = [\sigma_1 \ \sigma_2 \ \sigma_3 \ \sigma_4]$ là 4 hệ số trải

$Z = [Z_1; Z_2; Z_3; Z_4]$ là 4 ngõ ra của neuron lớp ẩn

$$\|X - C\| = \sqrt{(X - C)^T (X - C)}$$

- **Lớp ra:** Gồm 2 ngõ ra

$$\theta_{1net} = W_{11}Z_1 + W_{12}Z_2 + W_{13}Z_3 + W_{14}Z_4 = W_1Z$$

$$\theta_{2net} = W_{21}Z_1 + W_{22}Z_2 + W_{23}Z_3 + W_{24}Z_4 = W_2Z$$

Với:

$$W_1 = W_{1q} = [W_{11} \ W_{12} \ W_{13} \ W_{14}]$$

$$W_2 = W_{2q} = [W_{21} \ W_{22} \ W_{23} \ W_{24}]$$

$$W = \begin{bmatrix} W_1 \\ W_2 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \end{bmatrix}$$

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix}$$

- **Giải thuật huấn luyện:** Áp dụng phương pháp Steepert descent để tìm tâm, hệ số trải và các trọng số bằng quá trình tối thiểu hóa sai số bình phương.

+ **Hàm mục tiêu:** $J = \frac{1}{2} \sum_{j=1}^2 (\theta_{jp} - \theta_{jnet})^2$ (4.19)

+ **Cập nhật tâm:**

Luật cập nhật tâm:

$$C(k+1) = C(k) - \eta_c \nabla_c J$$

Với:

$$\begin{aligned} \nabla_c J &= \left(\frac{\partial J}{\partial C} \right) = \left(\frac{\partial J}{\partial \theta_{1net}} \cdot \frac{\partial \theta_{1net}}{\partial Z} \cdot \frac{\partial Z}{\partial C} \right) + \left(\frac{\partial J}{\partial \theta_{2net}} \cdot \frac{\partial \theta_{2net}}{\partial Z} \cdot \frac{\partial Z}{\partial C} \right) \\ &= \left(\left(\frac{\partial J}{\partial \theta_{1net}} \cdot \frac{\partial \theta_{1net}}{\partial Z} \right) + \left(\frac{\partial J}{\partial \theta_{2net}} \cdot \frac{\partial \theta_{2net}}{\partial Z} \right) \right) \frac{\partial Z}{\partial C} \\ &= -((\theta_{1p} - \theta_{1net})W_1 + (\theta_{2p} - \theta_{2net})W_2) \cdot 2 \cdot \frac{\|X - C\|}{\sigma^2} e^{-\frac{\|X - C\|^2}{\sigma^2}} \end{aligned}$$

Vậy luật cập nhật tâm được viết lại như sau:

$$C(k+1) = C(k) + \eta_c \left((\theta_{1p} - \theta_{1net})W_1 + (\theta_{2p} - \theta_{2net})W_2 \right) \cdot 2 \cdot \frac{\|X - C\|}{\sigma^2} e^{-\frac{\|X - C\|^2}{\sigma^2}} \quad (4.20)$$

+ Cập nhật hệ số trái:

$$\sigma(k+1) = \sigma(k) - \eta_{\sigma} \nabla_{\sigma} J$$

Với:

$$\begin{aligned} \nabla_{\sigma} J &= \left(\frac{\partial J}{\partial \sigma} \right) = \left(\frac{\partial J}{\partial \theta_{1net}} \cdot \frac{\partial \theta_{1net}}{\partial Z} \cdot \frac{\partial Z}{\partial \sigma} \right) + \left(\frac{\partial J}{\partial \theta_{2net}} \cdot \frac{\partial \theta_{2net}}{\partial Z} \cdot \frac{\partial Z}{\partial \sigma} \right) \\ &= \left(\left(\frac{\partial J}{\partial \theta_{1net}} \cdot \frac{\partial \theta_{1net}}{\partial Z} \right) + \left(\frac{\partial J}{\partial \theta_{2net}} \cdot \frac{\partial \theta_{2net}}{\partial Z} \right) \right) \frac{\partial Z}{\partial \sigma} \\ &= -((\theta_{1p} - \theta_{1net})W_1 + (\theta_{2p} - \theta_{2net})W_2) \cdot 2 \cdot \frac{\|X - C\|^2}{\sigma^3} e^{-\frac{\|X - C\|^2}{\sigma^2}} \end{aligned}$$

Vậy luật cập nhật hệ số trái được viết lại như sau:

$$\sigma(k+1) = \sigma(k) + \eta_{\sigma} ((\theta_{1p} - \theta_{1net})W_1 + (\theta_{2p} - \theta_{2net})W_2) \cdot 2 \cdot \frac{\|X - C\|^2}{\sigma^3} e^{-\frac{\|X - C\|^2}{\sigma^2}} \quad (4.21)$$

+ Cập nhật trọng số:

Luật cập nhật trọng số:

$$W_1(k+1) = W_1(k) - \eta_W \nabla_{W_1} J$$

$$W_2(k+1) = W_2(k) - \eta_W \nabla_{W_2} J$$

Với:

$$\nabla_{W_1} J = \left(\frac{\partial J}{\partial W_1} \right) = \left(\frac{\partial J}{\partial \theta_{1net}} \cdot \frac{\partial \theta_{1net}}{\partial W_1} \right) = -((\theta_{1p} - \theta_{1net})Z)$$

$$\nabla_{W_2} J = \left(\frac{\partial J}{\partial W_2} \right) = \left(\frac{\partial J}{\partial \theta_{2net}} \cdot \frac{\partial \theta_{2net}}{\partial W_2} \right) = -((\theta_{2p} - \theta_{2net})Z)$$

Vậy luật cập nhật trọng số được viết lại như sau:

$$W_1(k+1) = W_1(k) + \eta_W (\theta_{1p} - \theta_{1net})Z \quad (4.22)$$

$$W_2(k+1) = W_2(k) + \eta_W (\theta_{2p} - \theta_{2net})Z \quad (4.23)$$

$$W(k+1) = \begin{bmatrix} W_1(k+1) \\ W_2(k+1) \end{bmatrix} = \begin{bmatrix} W_1(k) + \eta_W (\theta_{1p} - \theta_{1net})Z \\ W_2(k) + \eta_W (\theta_{2p} - \theta_{2net})Z \end{bmatrix} \quad (4.24)$$

Thuật toán huấn luyện tâm, hệ số trái, trọng số lớp ra của mạng nhận dạng:

+ Bước 1:

Chọn hằng số học $\eta = 0.3$

Load mẫu dữ liệu vào ra mong muốn

Chọn hàm mục tiêu:

$$J = \frac{1}{2} \sum_{j=1}^2 (\theta_{jp} - \theta_{jnet})^2$$

+ Bước 2 : khởi động:

Gán biến chạy $k = 1$

Gán trọng số, tâm, hệ số trái bằng giá trị ngẫu nhiên bất kỳ

+ Bước 3: Tính ngõ ra của các neural lớp ẩn, lớp ra:

$$Z = g(\|X - C\|) = e^{-\frac{\|X - C\|^2}{\sigma^2}}$$

$$\theta_{1net} = W_{11}Z_1 + W_{12}Z_2 + W_{13}Z_3 + W_{14}Z_4 = W_1Z$$

$$\theta_{2net} = W_{21}Z_1 + W_{22}Z_2 + W_{23}Z_3 + W_{24}Z_4 = W_2Z$$

+ Bước 4: Cập nhật trọng số của lớp ra, tâm và hệ số trái của mạng:

$$W(k+1) = \begin{bmatrix} W_1(k+1) \\ W_2(k+1) \end{bmatrix} = \begin{bmatrix} W_1(k) + \eta_W (\theta_{1p} - \theta_{1net})Z \\ W_2(k) + \eta_W (\theta_{2p} - \theta_{2net})Z \end{bmatrix} \quad (4.25)$$

$$C(k+1) = C(k) + \eta_C ((\theta_{1p} - \theta_{1net})W_1 + (\theta_{2p} - \theta_{2net})W_2) \cdot 2 \cdot \frac{\|X - C\|}{\sigma^2} e^{-\frac{\|X - C\|^2}{\sigma^2}} \quad (4.26)$$

$$\sigma(k+1) = \sigma(k) + \eta_\sigma ((\theta_{1p} - \theta_{1net})W_1 + (\theta_{2p} - \theta_{2net})W_2) \cdot 2 \cdot \frac{\|X - C\|^2}{\sigma^3} e^{-\frac{\|X - C\|^2}{\sigma^2}} \quad (4.27)$$

+ Bước 5 :

Nếu $k < K$ (số mẫu huấn luyện) thì gán $k = k+1$ và trở lại bước 3

Nếu $k = K$ thì tiếp tục bước 6

Bước 6: kết thúc 1 chu kỳ huấn luyện

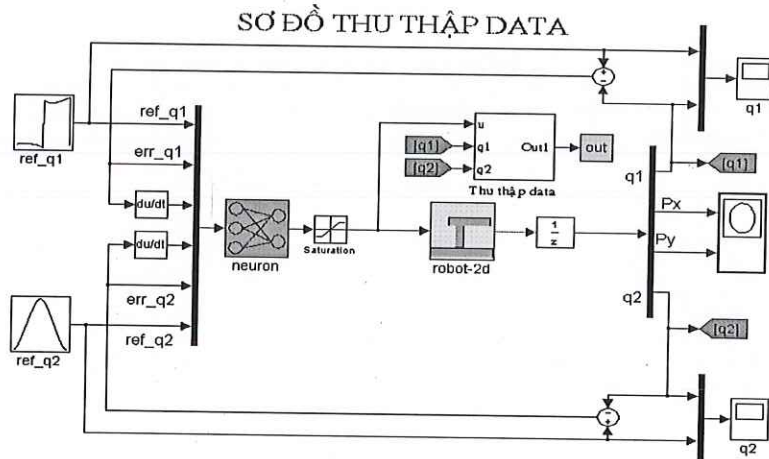
Nếu $J \leq l_r$ thì kết thúc quá trình học

Nếu $J > l_r$ thì gán $k = 1$ và trở lại bước 3

4.2.3 Mô phỏng mạng nhận dạng:

Thu thập dữ liệu:

- + Tạo tín hiệu đặt là 1 trong các mẫu cắt (hình tròn, vuông, tam giác . .)
- + Cho hệ thống điều khiển chạy để thu thập data:



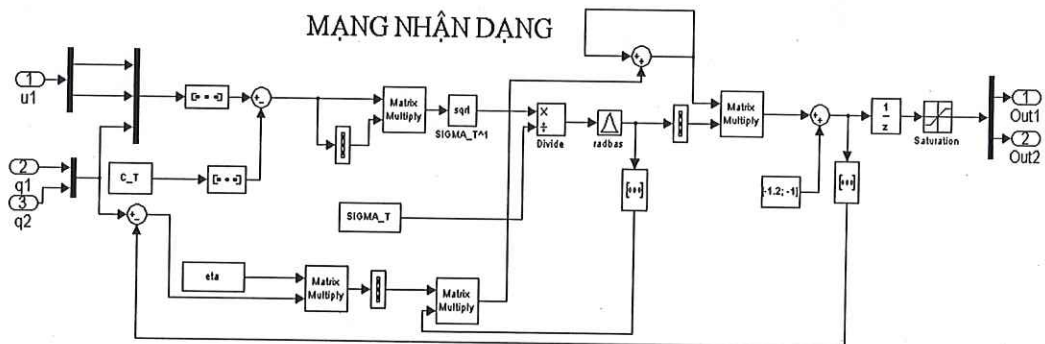
Hình 4.11 Sơ đồ thu thập data

- + Sau khi thu thập dữ liệu xong ta được dữ liệu ngõ vào của mạng nhận dạng được lưu trong file input.mat, dữ liệu ngõ ra mong muốn của mạng nhận dạng được lưu trong file output.mat

Nhận dạng dùng mạng RBF:

+ Hàm mục tiêu $J = \frac{1}{2}(d_1 - y_1)^2 + \frac{1}{2}(d_2 - y_2)^2 = \frac{1}{2}erro_{T_1}^2 + \frac{1}{2}erro_{T_2}^2$

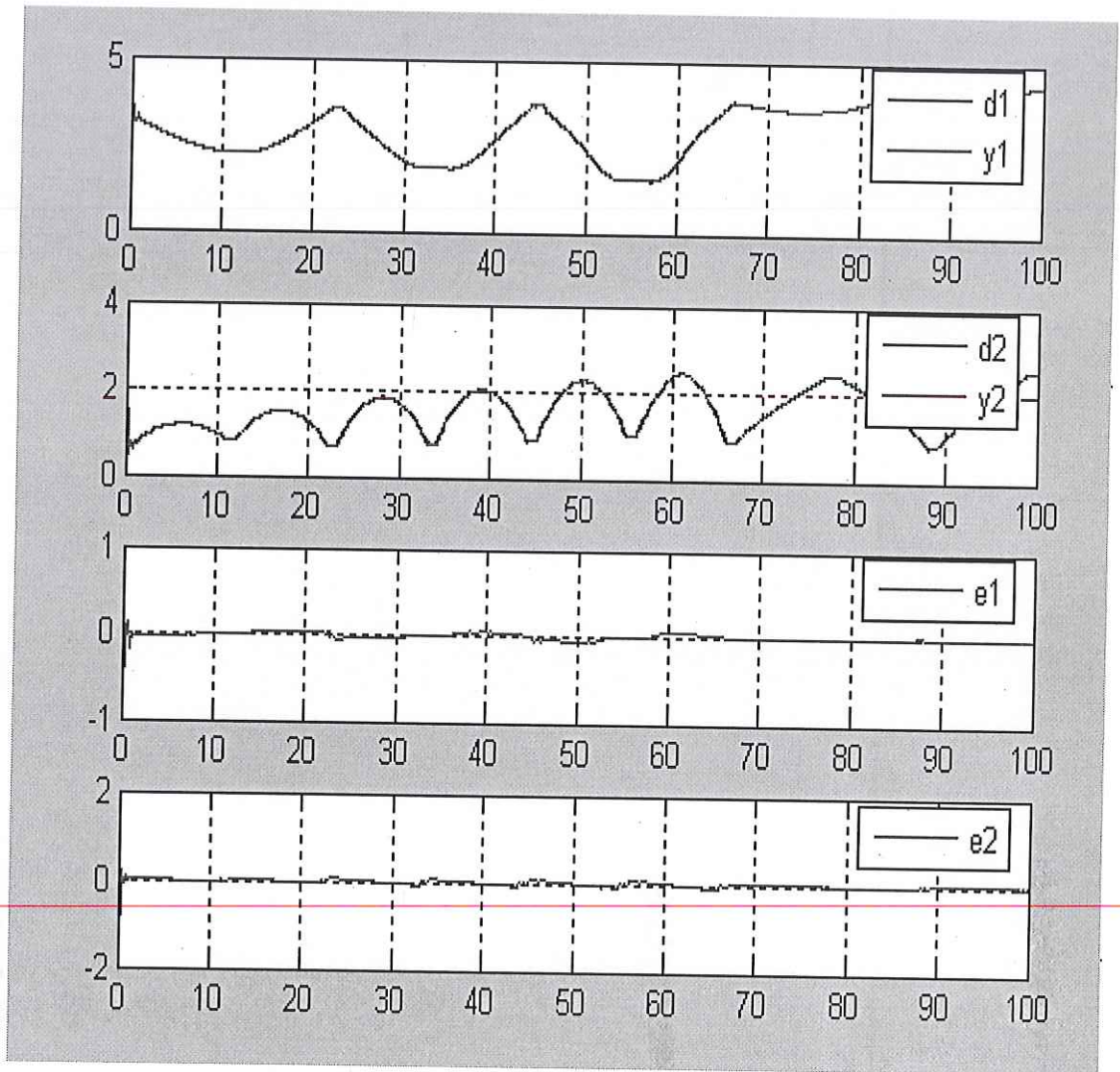
- + Sơ đồ simulink:



Hình 4.12 Sơ đồ simulink mạng nhận dạng

+ Huấn luyện tâm, hệ số trái, trọng số lớp ra của mạng nhận dạng rồi đưa đến sơ đồ simulink mạng nhận dạng

Kết quả nhận dạng:



Hình 4.13 Kết quả nhận dạng

Với:

d_1, d_2 là 2 ngõ ra mong muốn của mạng nhận dạng

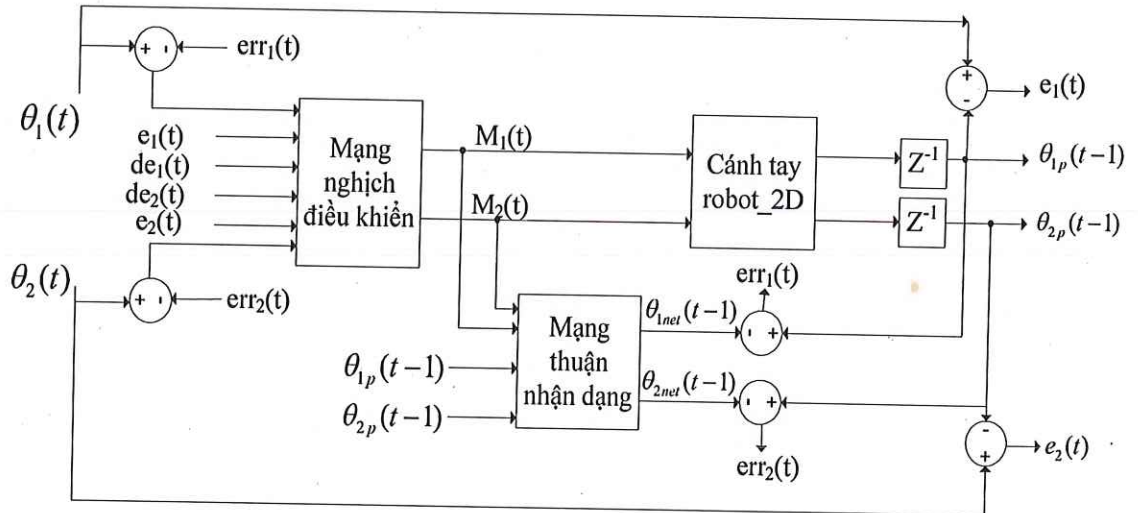
y_1, y_2 là 2 ngõ ra của mạng nhận dạng

$$e_1 = d_1 - y_1$$

$$e_2 = d_2 - y_2$$

4.3 Thiết kế hệ thống điều khiển mô hình nội cánh tay Robot 2 khớp nối dùng mạng RBF:

4.3.1 Sơ đồ khối:



Hình 4.14 Hệ thống điều khiển mô hình nội dùng mạng RBF

Hệ thống gồm các khối:

- Khối điều khiển: Sử dụng bộ điều khiển neural PD

+ Ngõ vào gồm có tín hiệu đặt $\theta_1(t) - err_1(t)$, $\theta_2(t) - err_2(t)$, sai số $e_1(t)$, $e_2(t)$, vi phân sai số $de_1(t)$, $de_2(t)$

Với :

$$err_1(t) = (\theta_{1p}(t-1) - \theta_{1net}(t-1))$$

$$err_2(t) = (\theta_{2p}(t-1) - \theta_{2net}(t-1))$$

$$e_1(t) = \theta_1(t) - \theta_{1p}(t-1)$$

$$e_2(t) = \theta_2(t) - \theta_{2p}(t-1)$$

+ Ngõ ra $u_{net}(t) = \{M_1(t), M_2(t)\}$

- Khối đối tượng:

+ Ngõ vào: $u_{net}(t) = \{M_1(t), M_2(t)\}$

+ Ngõ ra của đối tượng: $y_p(t) = \{ \theta_{1p}(t), \theta_{2p}(t) \}$. Ngõ ra của đối tượng được tìm bằng cách giải phương trình vi phân bậc 2 dùng phương pháp Euler

- Khối nhận dạng: Có chức năng nhận dạng mô hình toán của cánh tay robot
- + Ngõ vào của mạng nhận dạng: $x(t) = \{M_1(t), M_2(t), \theta_{1p}(t-1), \theta_{2p}(t-1)\}$
- + Ngõ ra của mạng nhận dạng: $y_{net}(t) = \{\theta_{1net}(t), \theta_{2net}(t)\}$
- + Các sai số $err_1(t), err_2(t)$ dùng để cập nhật trọng số khi huấn luyện mạng RBF

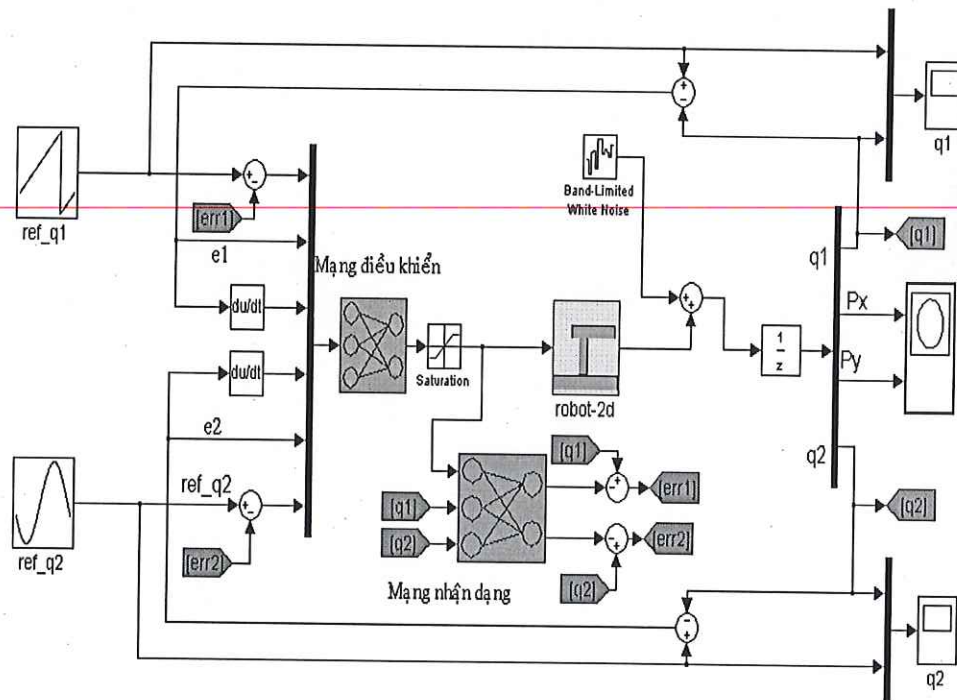
Trong hệ thống điều khiển mô hình nội thì sai số được lan truyền qua đối tượng nhờ mạng thuận nhận dạng mô hình đối tượng. Các sai số này ($err_1(t), err_2(t)$) được đưa đến mô hình mạng neuron của bộ điều khiển.

Việc huấn luyện mạng nghịch tương tự mục 4.1.3. Mạng gồm có 6 ngõ vào $u(t) = \{\theta_1(t) - err_1(t), e_1(t), de_1(t), \theta_2(t) - err_2(t), e_2(t), de_2(t)\}$. Ở ngõ vào thứ 1 và thứ 4 ta thêm các sai số $err_1(t), err_2(t)$ vào, 2 ngõ ra $u_{net}(t) = \{M_1(t), M_2(t)\}$

4.3.2 Mô phỏng bộ điều khiển mô hình nội theo mô hình toán:

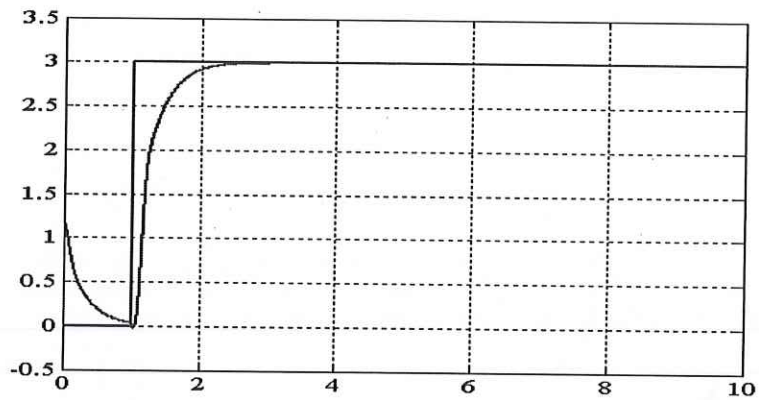
- Sơ đồ simulink:

HỆ THỐNG ĐIỀU KHIỂN MÔ HÌNH NỘI CÁN TAY ROBOT DÙNG MẠNG RBF



Hình 4.15 Sơ đồ simulink bộ điều khiển mô hình nội

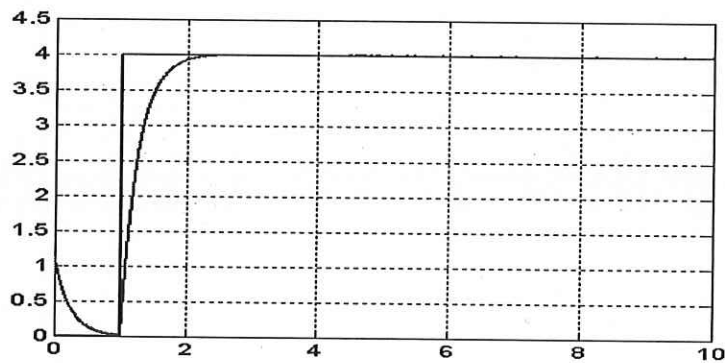
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là hàm bước:



Hình 4.16 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là hàm bước

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ, t_{qd} lớn

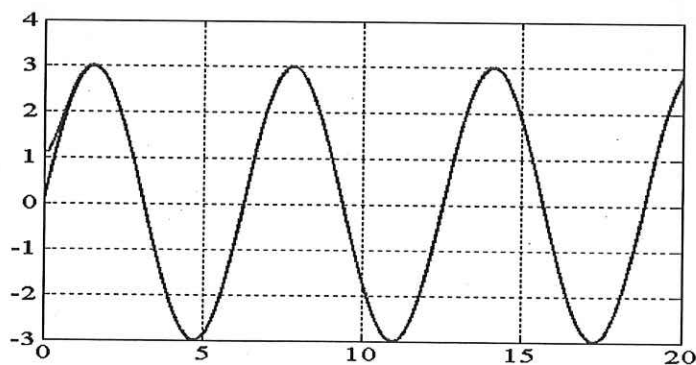
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là hàm bước:



Hình 4.17 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là hàm bước

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số nhỏ, t_{qd} lớn

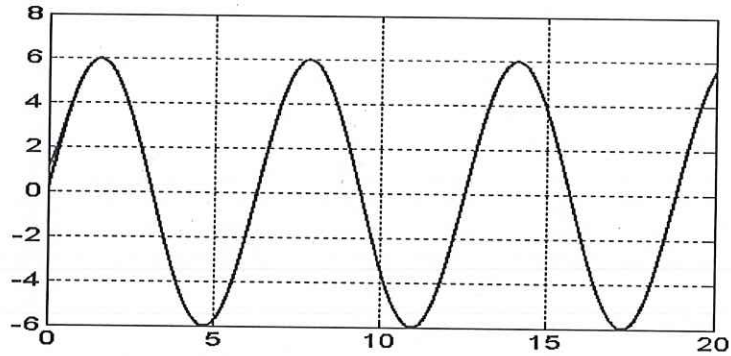
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình sin:



Hình 4.18 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình sin

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ

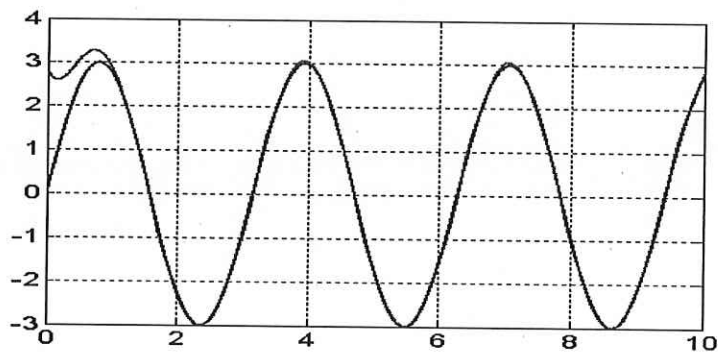
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình sin:



Hình 4.19 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình sin

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số nhỏ

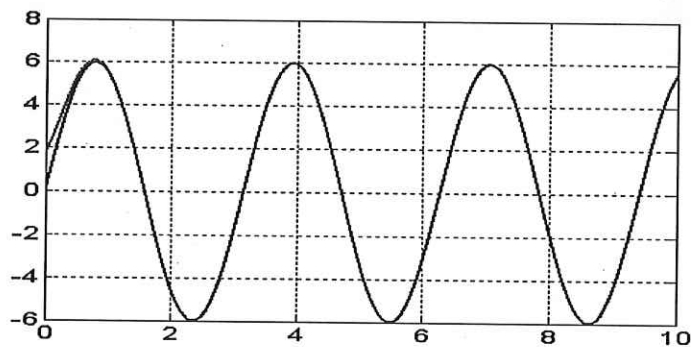
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình sin khi tăng tần số:



Hình 4.20 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình sin khi tăng tần số

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ

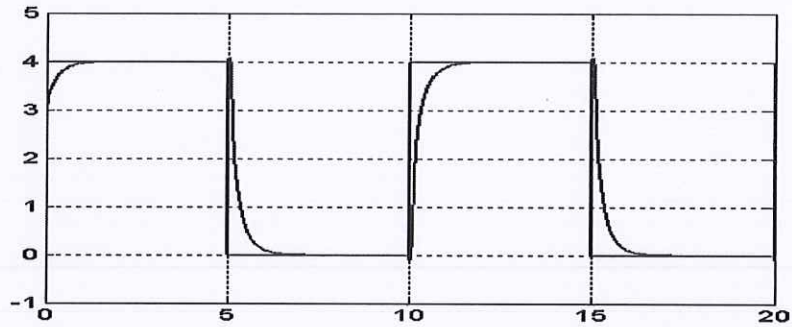
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình sin khi tăng tần số:



Hình 4.21 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình sin khi tăng tần số

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số nhỏ

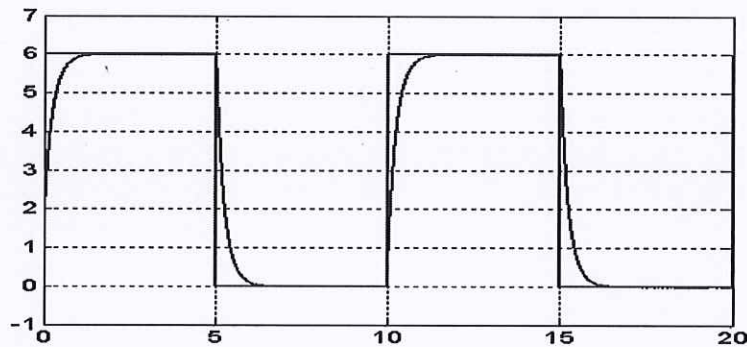
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông:



Hình 4.22 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ, t_{qd} lớn

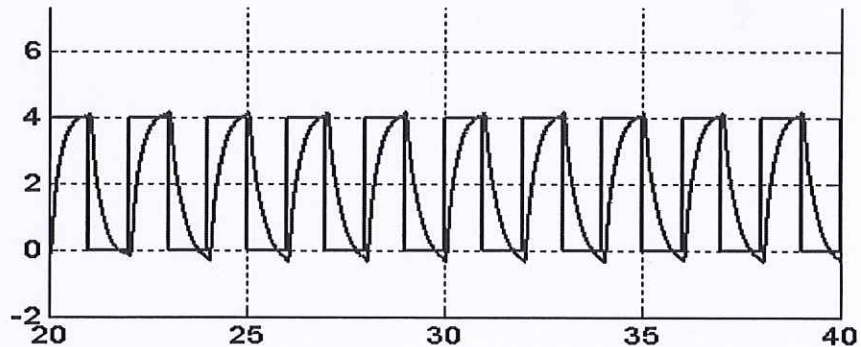
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông:



Hình 4.23 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông

Nhận xét: Đáp ứng ngõ ra θ_2 không vọt lố, sai số nhỏ, t_{qd} lớn

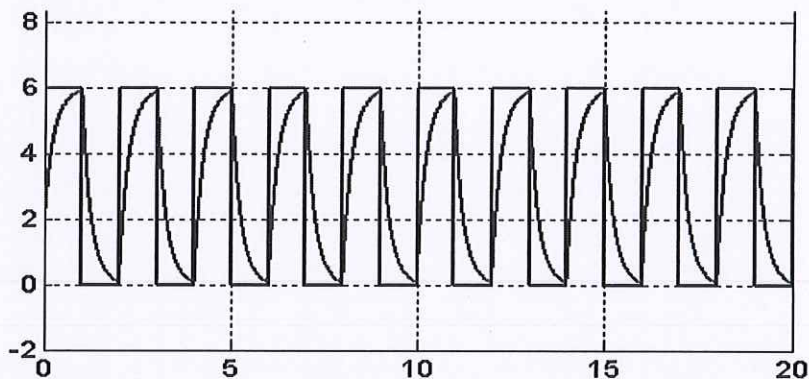
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông khi tăng tần số:



Hình 4.24 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung vuông khi tăng tần số

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số lớn

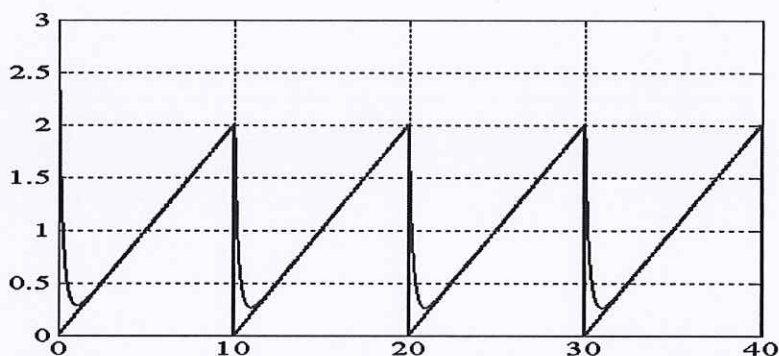
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông khi tăng tần số:



Hình 4.25 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung vuông khi tăng tần số

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số lớn

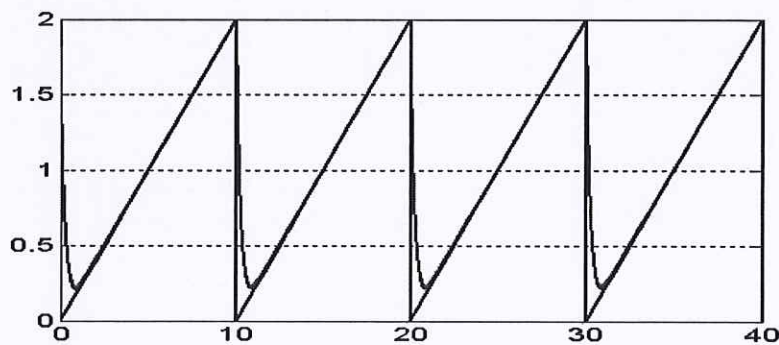
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa:



Hình 4.26 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số nhỏ ở các đỉnh của xung tam giác

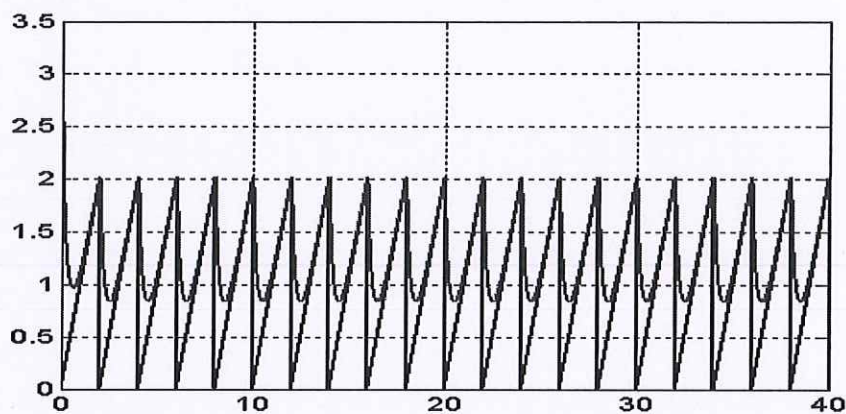
- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa:



Hình 4.27 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa

Nhận xét: Đáp ứng ngõ ra θ_2 có sai số nhỏ ở các đỉnh của xung tam giác

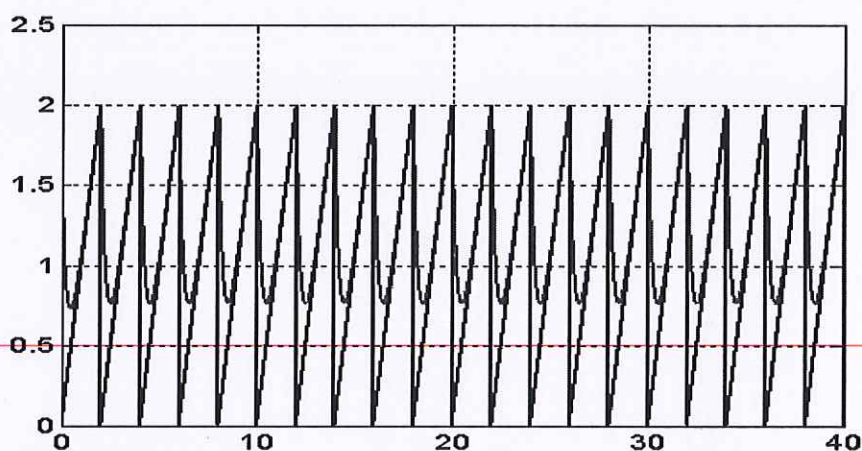
- Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa khi tăng tần số:



Hình 4.28 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là xung răng cưa khi tăng tần số

Nhận xét: Đáp ứng ngõ ra θ_1 có sai số lớn ở các đỉnh của xung tam giác

- Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa khi tăng tần số:



Hình 4.29 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là xung răng cưa khi tăng tần số

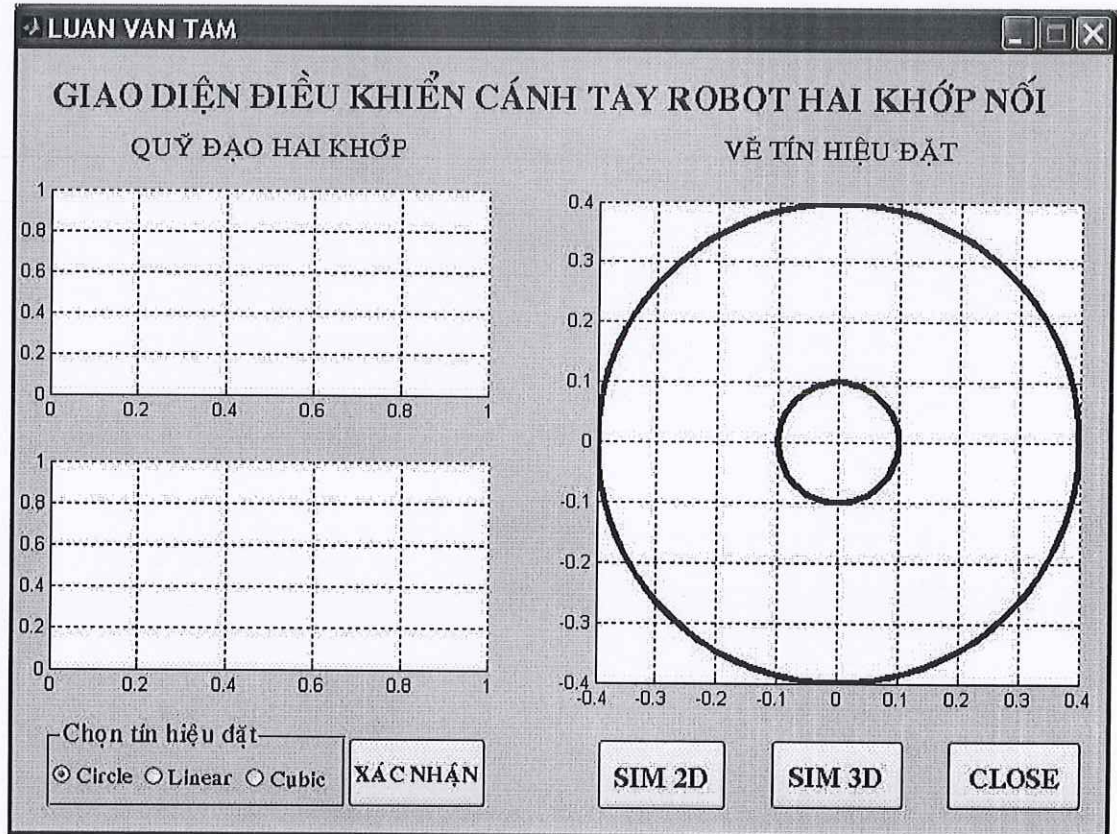
Nhận xét: Đáp ứng ngõ ra θ_2 có sai số lớn ở các đỉnh của xung tam giác

CHƯƠNG V:

KẾT QUẢ MÔ PHỎNG

Chương trình mô phỏng chạy trên matlab 7.0

5.1 Giao diện vẽ tín hiệu đặt:



Hình 5.1 Giao diện vẽ tín hiệu đặt

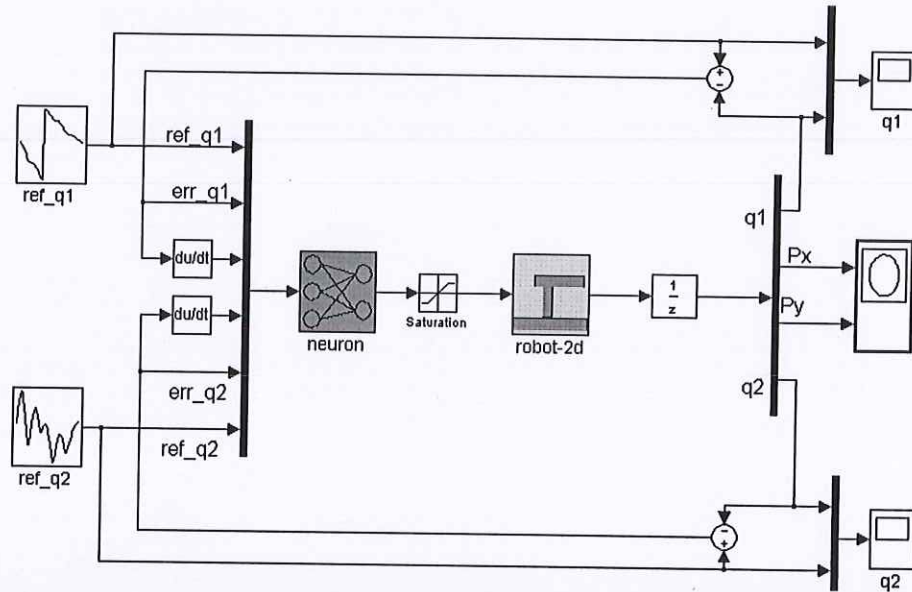
Các bước vẽ tín hiệu đặt:

- + Chọn tín hiệu đặt: Có 3 loại tín hiệu đặt là đường tròn, đường thẳng và đường cong
- + Click nút xác nhận
- + Vẽ tín hiệu đặt ở giữa 2 đường tròn giới hạn

Khi tín hiệu đặt được vẽ thì chương trình matlab sẽ tính ra tọa độ 2 khớp và vẽ lên giao diện, đồng thời lưu thành 2 file toa_do_q1.mat và toa_do_q2.mat để đưa đến sơ đồ simulink.

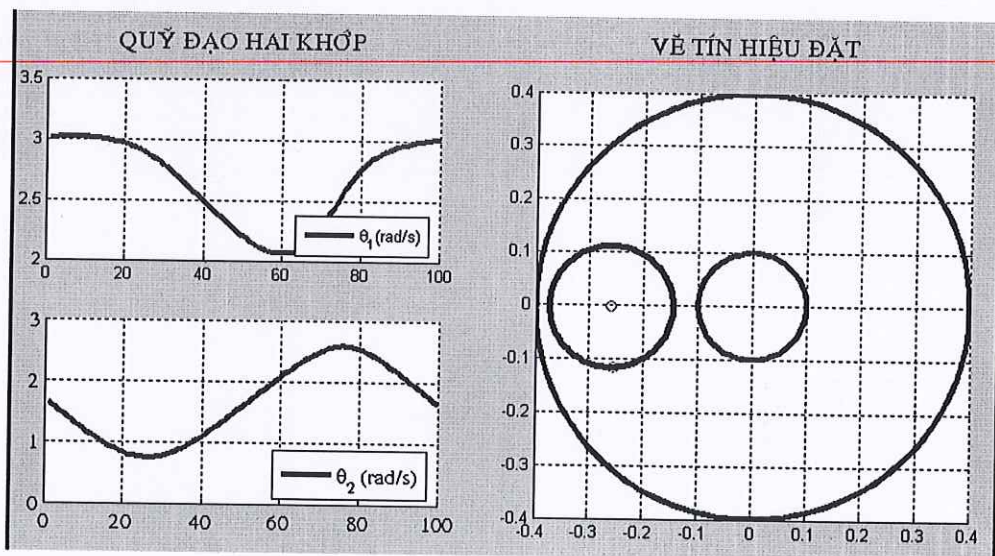
5.2 Kết quả mô phỏng của hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neuron PD:

5.2.1 Sơ đồ simulink hệ thống điều khiển cánh tay Robot 2 khớp nối dùng bộ điều khiển neuron PD:

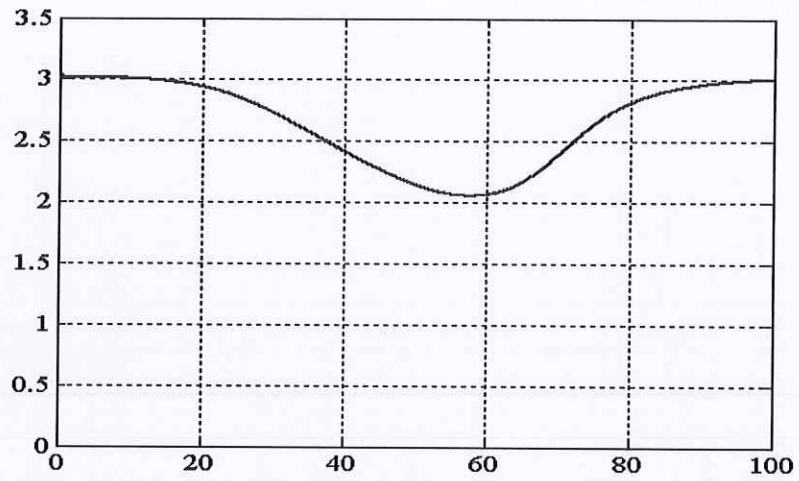


Hình 5.2 Sơ đồ simulink hệ thống điều khiển cánh tay robot 2 khớp nối dùng bộ điều khiển neuron PD

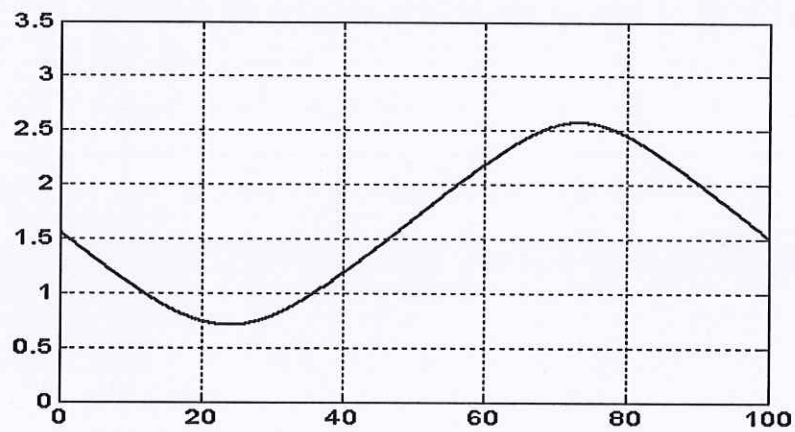
5.2.2 Kết quả mô phỏng:



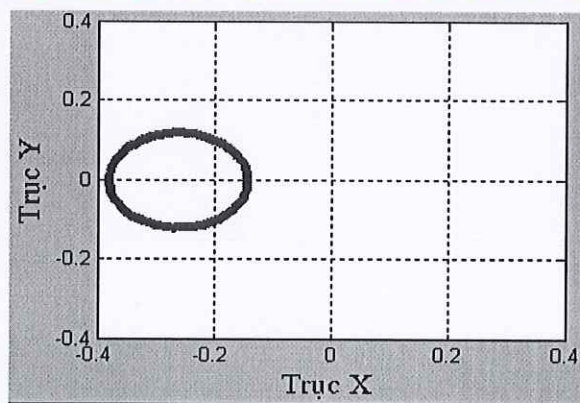
Hình 5.3 Tín hiệu đặt là mẫu cắt hình tròn



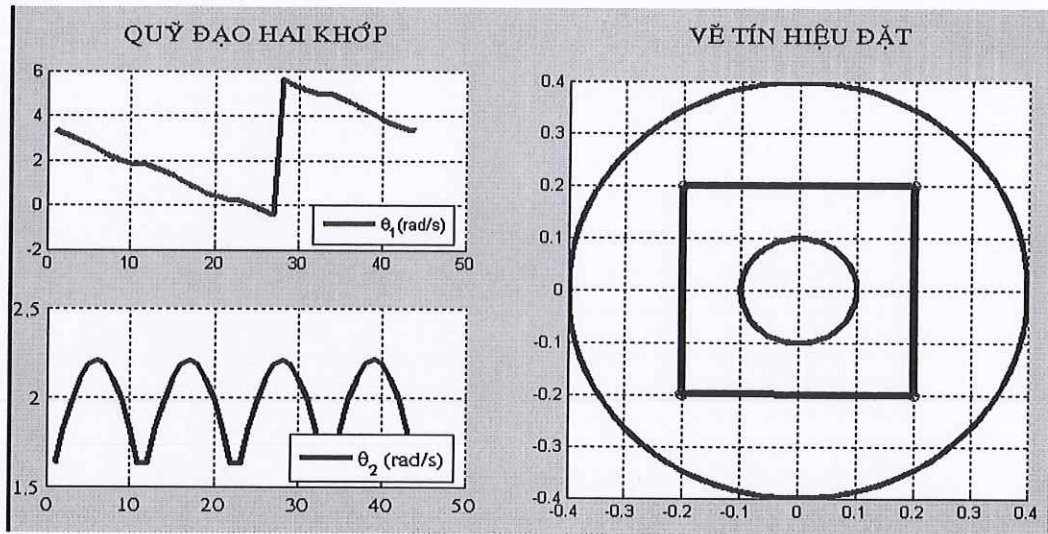
Hình 5.4 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là mẫu cắt hình tròn



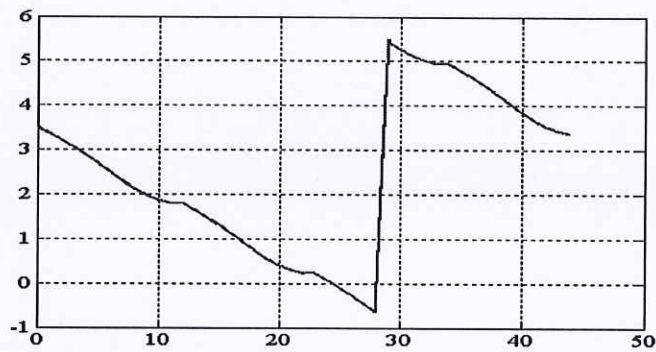
Hình 5.5 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là mẫu cắt hình tròn



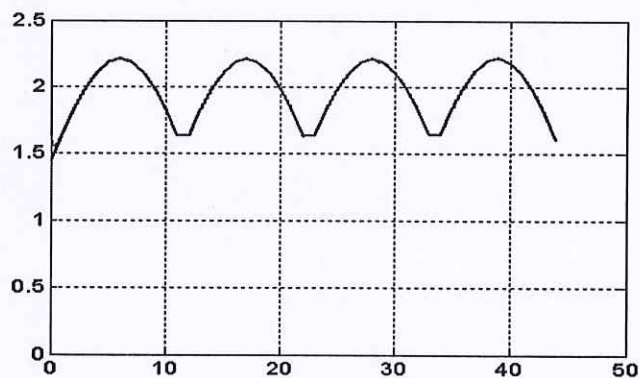
Hình 5.6 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt là mẫu cắt hình tròn



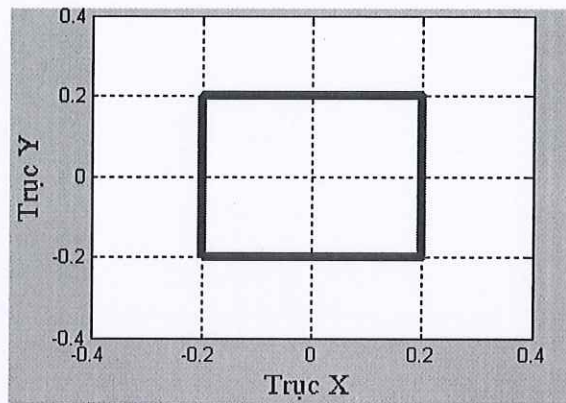
Hình 5.7 Tín hiệu đặt là mẫu cắt hình vuông



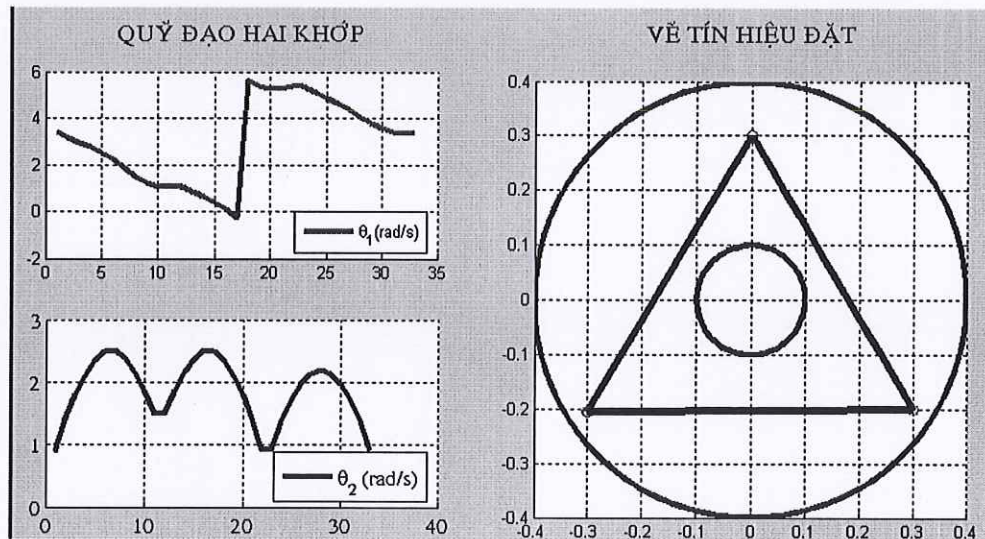
Hình 5.8 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là mẫu cắt hình vuông



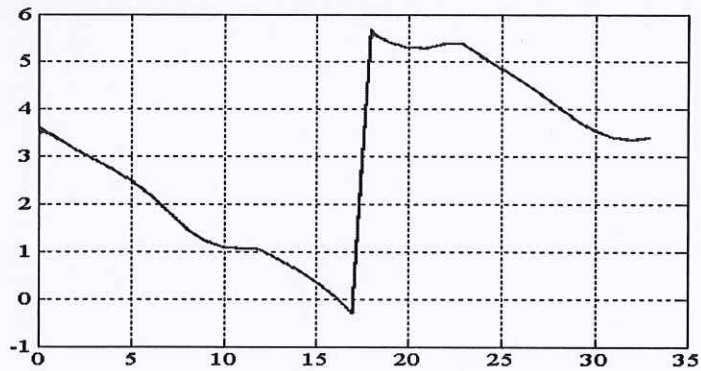
Hình 5.9 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là mẫu cắt hình vuông



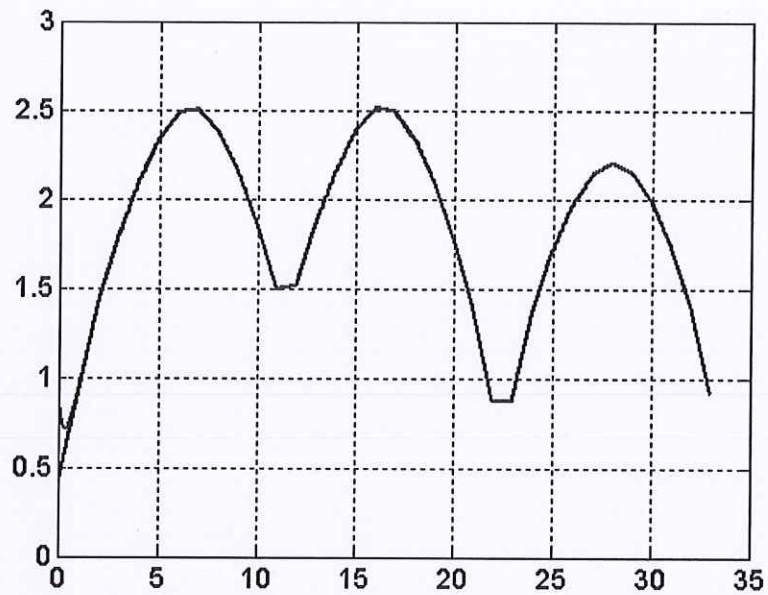
Hình 5.10 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt là mẫu cắt hình vuông



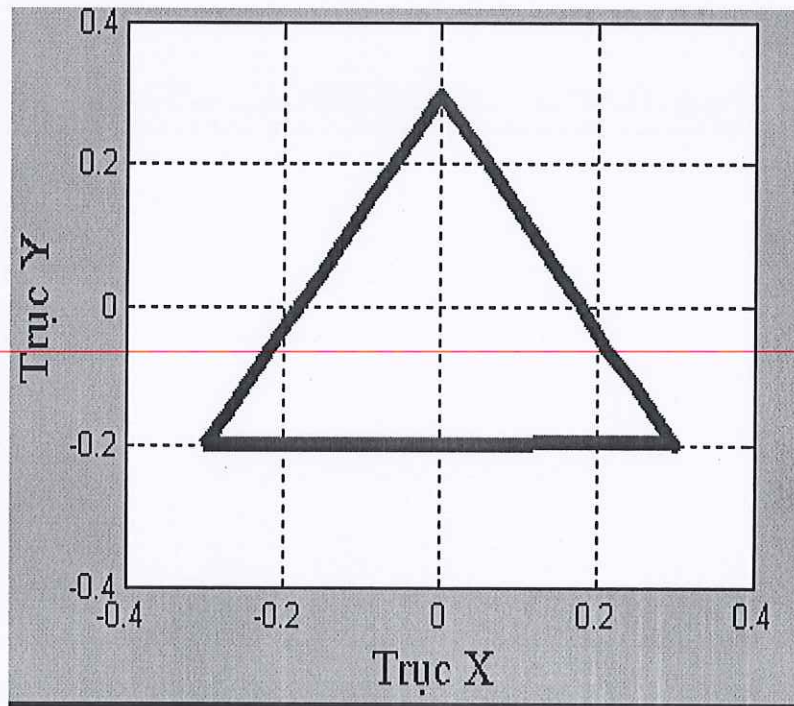
Hình 5.11 Tín hiệu đặt là mẫu cắt hình tam giác



Hình 5.12 Đáp ứng ngõ ra θ_1 với tín hiệu đặt là mẫu cắt hình tam giác



Hình 5.13 Đáp ứng ngõ ra θ_2 với tín hiệu đặt là mẫu cắt hình tam giác

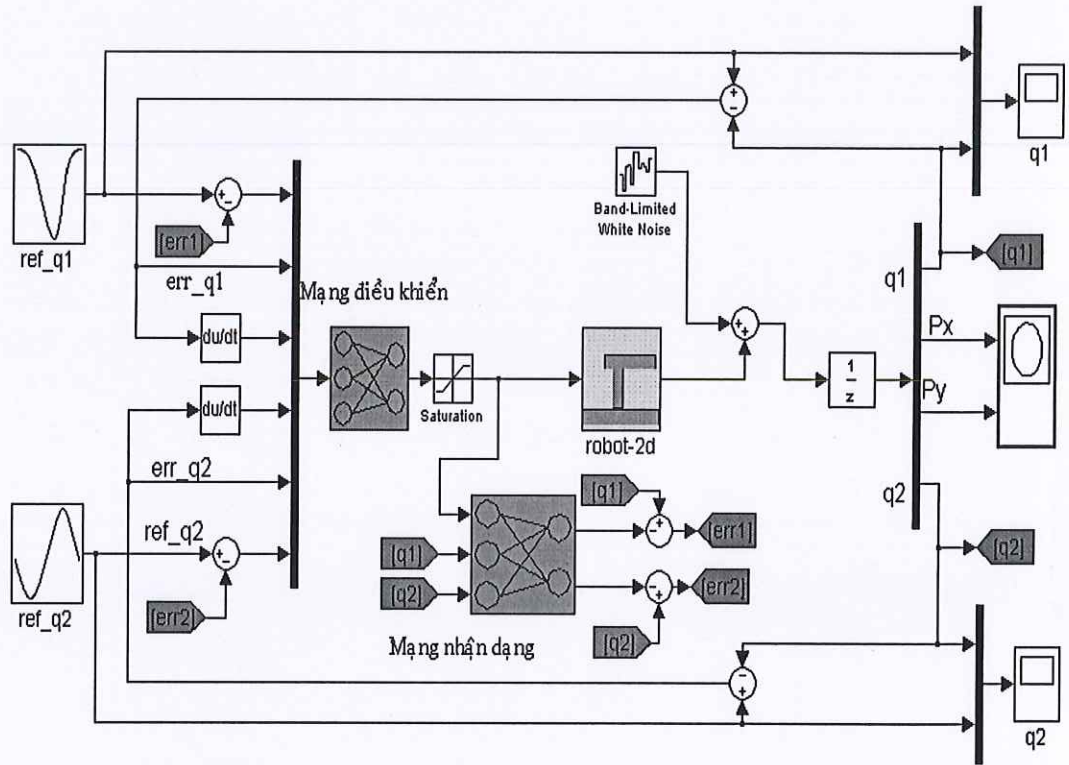


Hình 5.14 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt là mẫu cắt hình tam giác

Nhận xét: Đáp ứng ngõ ra θ_1 , θ_2 ít dao động, ít vọt lố, thời gian đáp ứng nhanh, sai số nhỏ, đáp ứng của cơ cấu chấp hành có sai số nhỏ

5.3 Kết quả mô phỏng của hệ thống điều khiển mô hình nội cánh tay Robot 2 khớp nối dùng mạng RBF:

5.3.1 Sơ đồ simulink hệ thống điều khiển mô hình nội cánh tay Robot 2 khớp nối dùng mạng RBF:



Hình 5.15 Sơ đồ simulink hệ thống điều khiển mô hình nội cánh tay robot 2 khớp nối

Trong đó:

$$q_1 = \theta_{1p}(t-1)$$

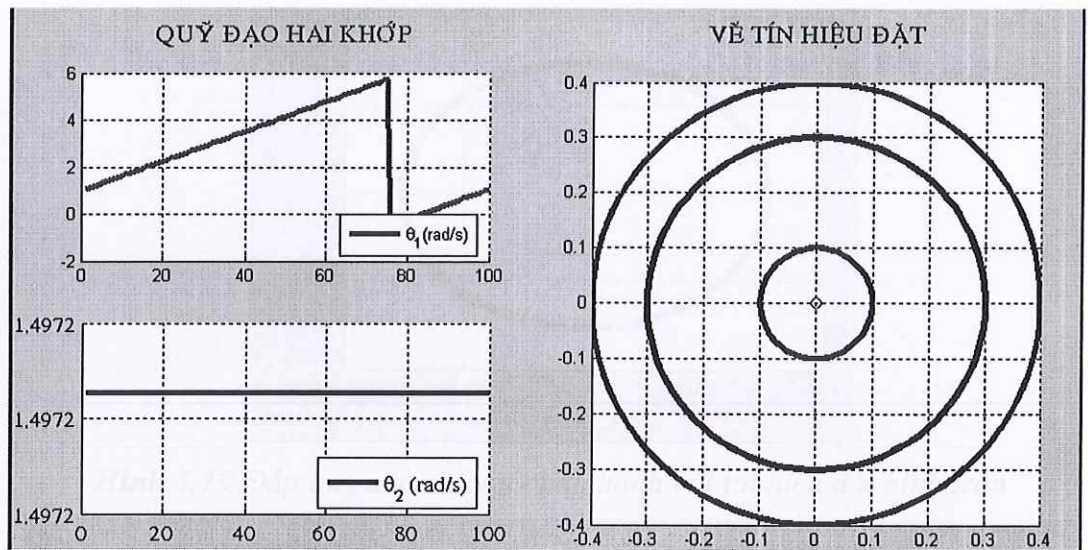
$$q_2 = \theta_{2p}(t-1)$$

$$ref_q_1 = \theta_1$$

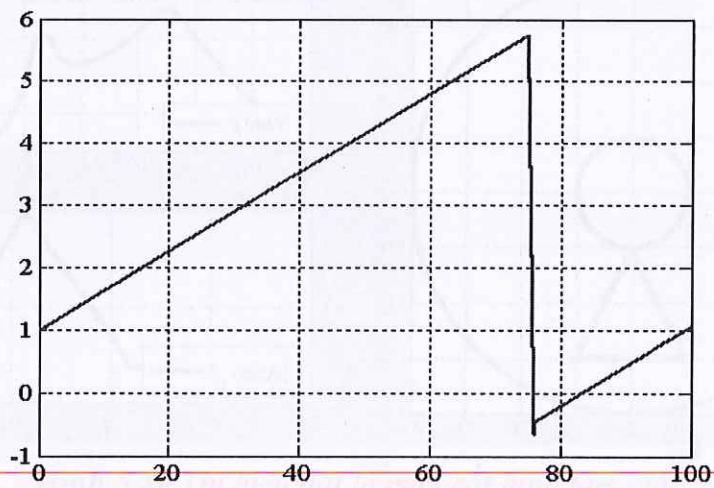
$$ref_q_2 = \theta_2$$

5.3.2 Kết quả mô phỏng khi không có nhiễu tác động ở ngõ ra:

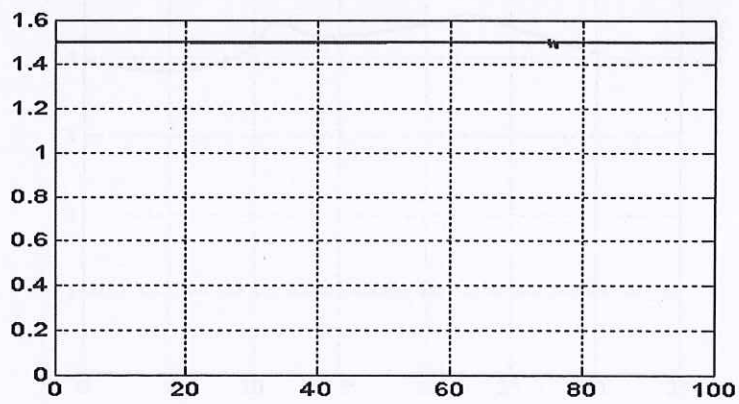
- + Trường hợp 1: Thông số cánh tay robot: $a_1 = 0.25m$, $a_2 = 0.15m$, $m_1 = 2.5kg$, $m_2 = 1kg$, $m_t = 0.5kg$



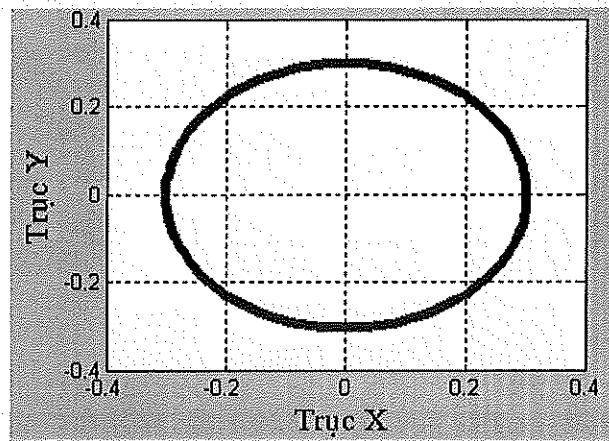
Hình 5.16 Tín hiệu đặt là mẫu cắt hình tròn



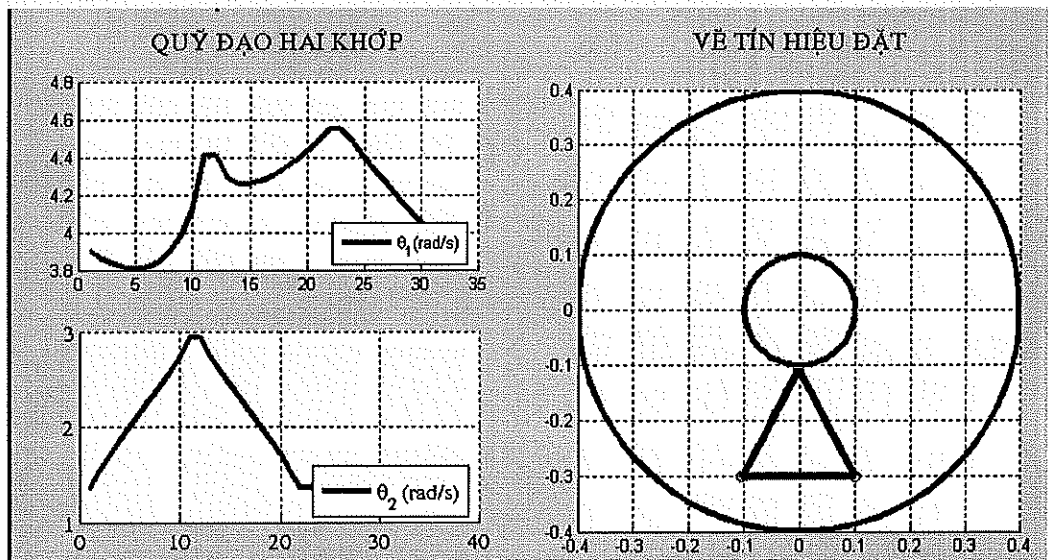
Hình 5.17 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn



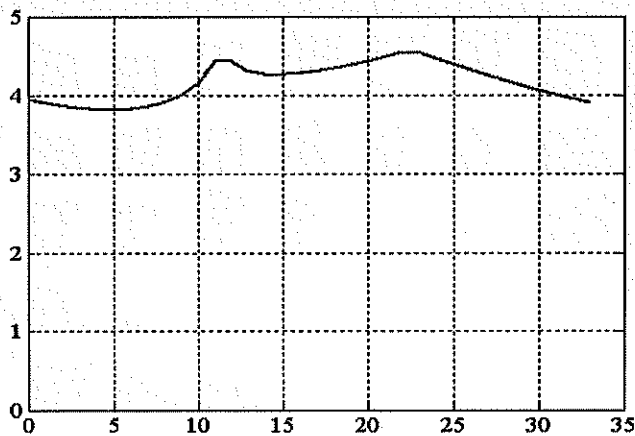
Hình 5.18 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn



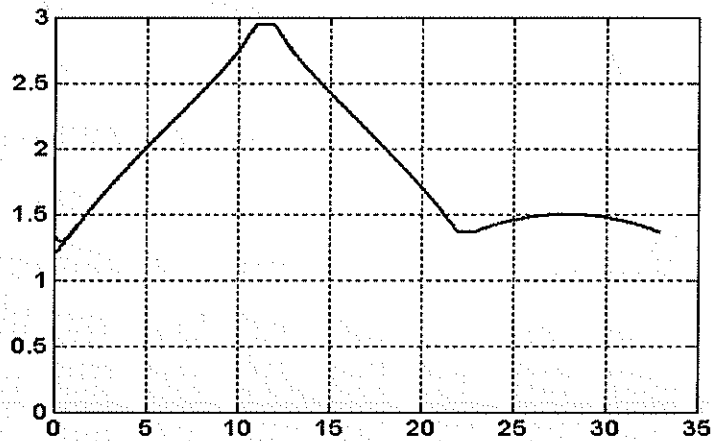
Hình 5.19 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn



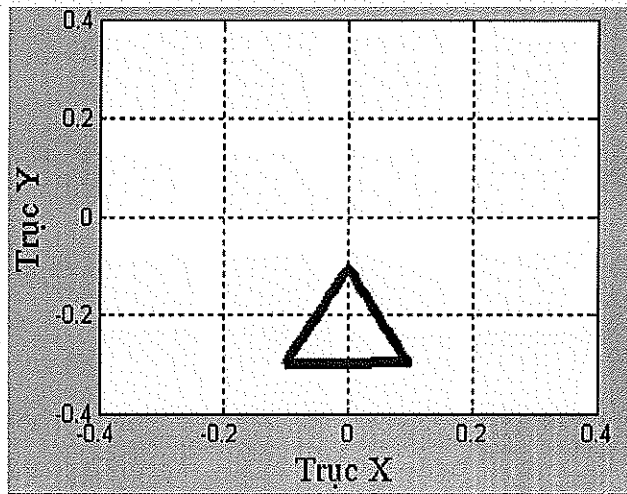
Hình 5.20 Tín hiệu đặt là mẫu cắt hình tam giác



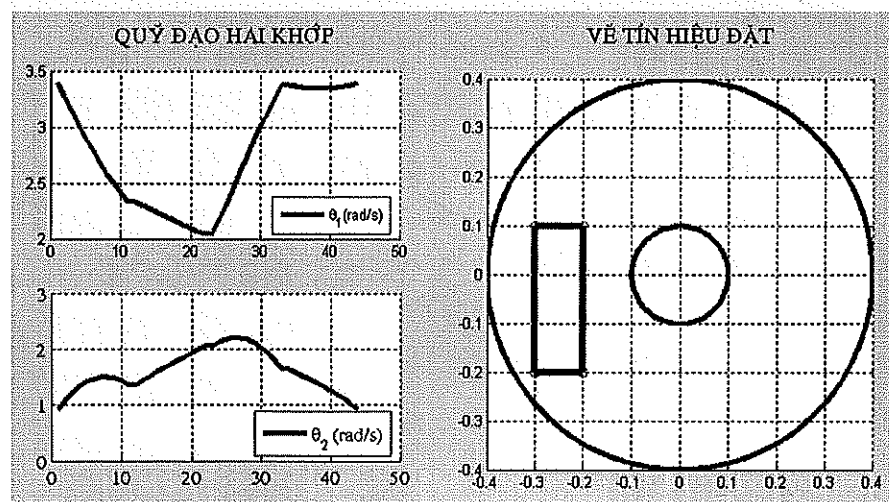
Hình 5.21 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tam giác



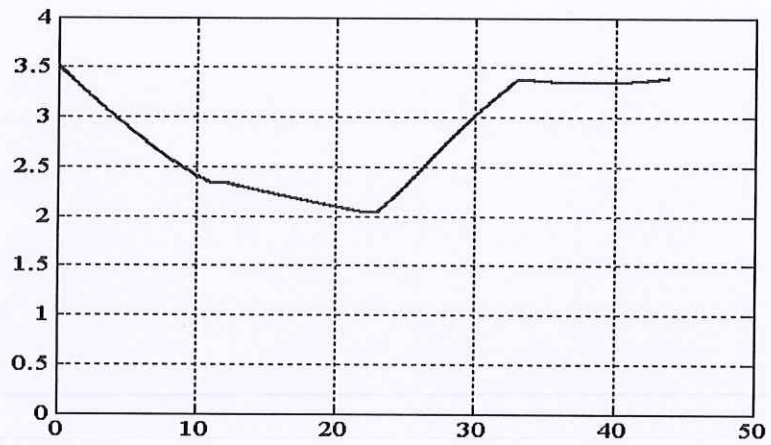
Hình 5.22 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tam giác



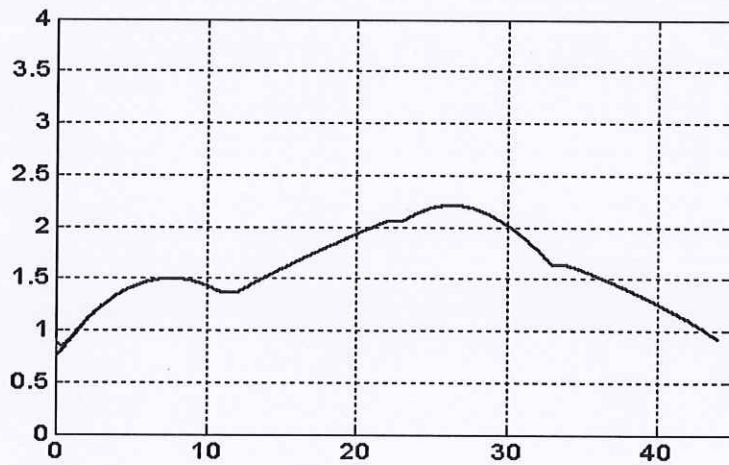
Hình 5.23 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tam giác



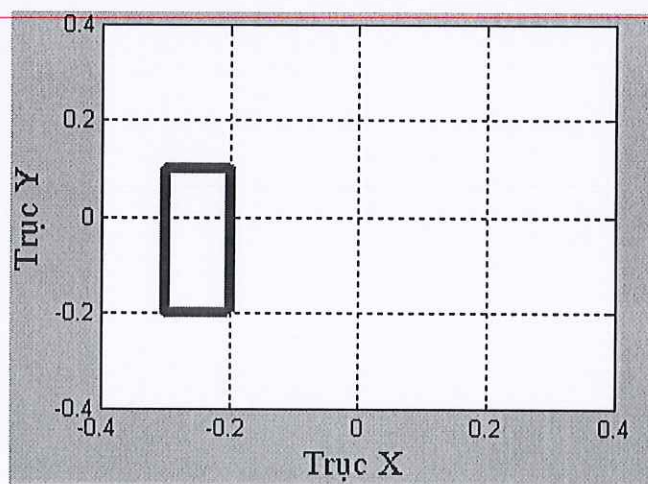
Hình 5.24 Tín hiệu đặt là mẫu cắt hình chữ nhật



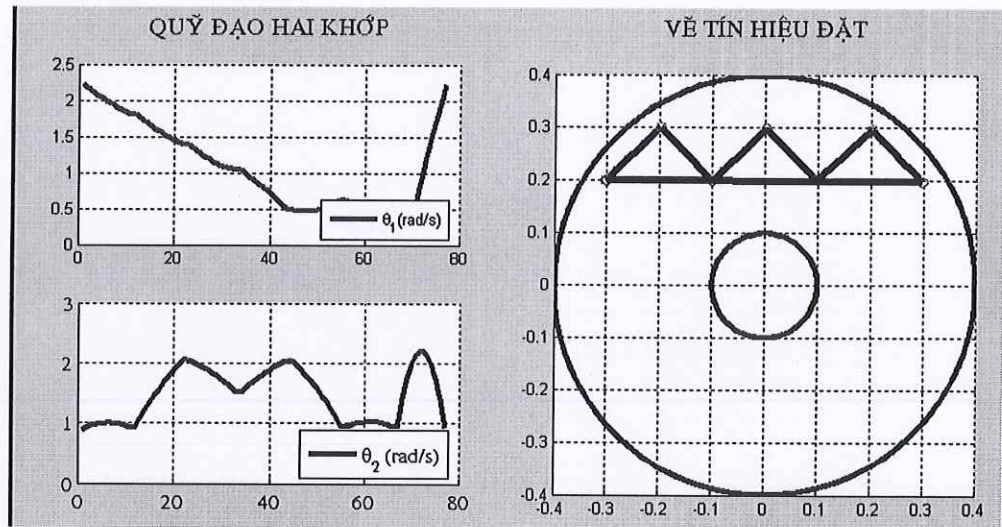
Hình 5.25 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình chữ nhật



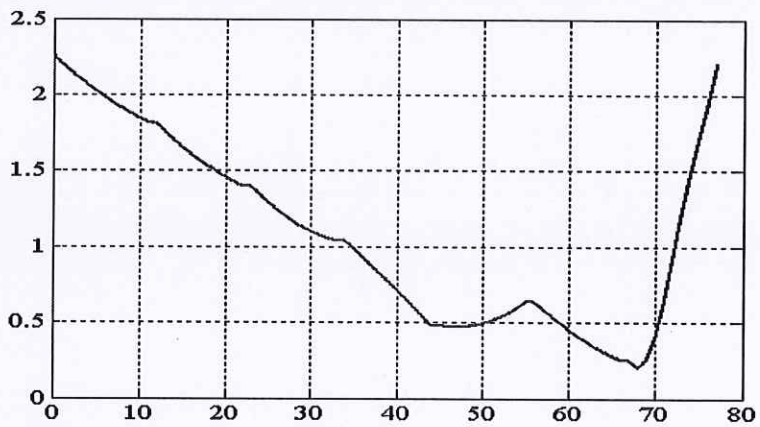
Hình 5.26 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình chữ nhật



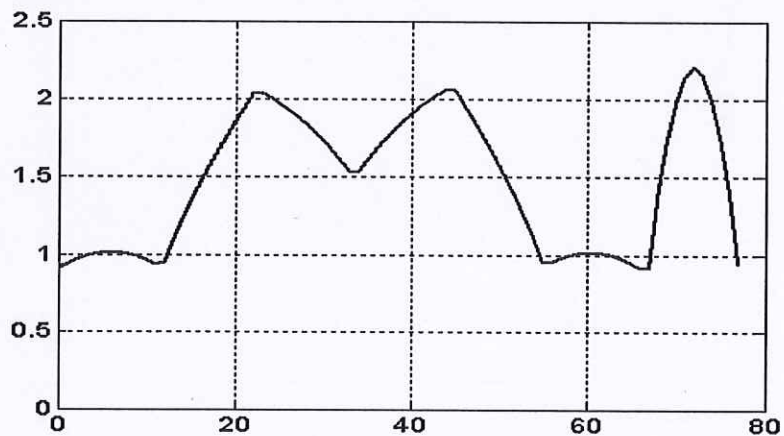
Hình 5.27 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình chữ nhật



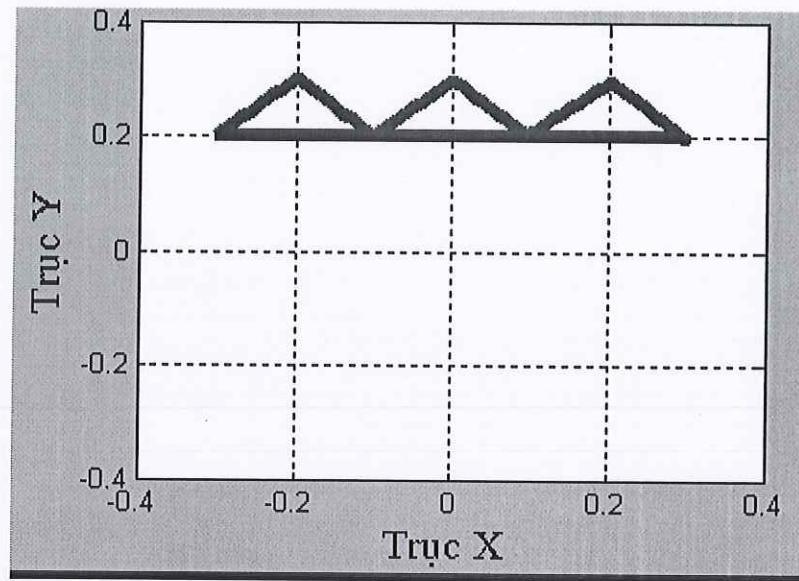
Hình 5.28 Tín hiệu đặt là mẫu cắt 3 hình tam giác



Hình 5.29 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tam giác



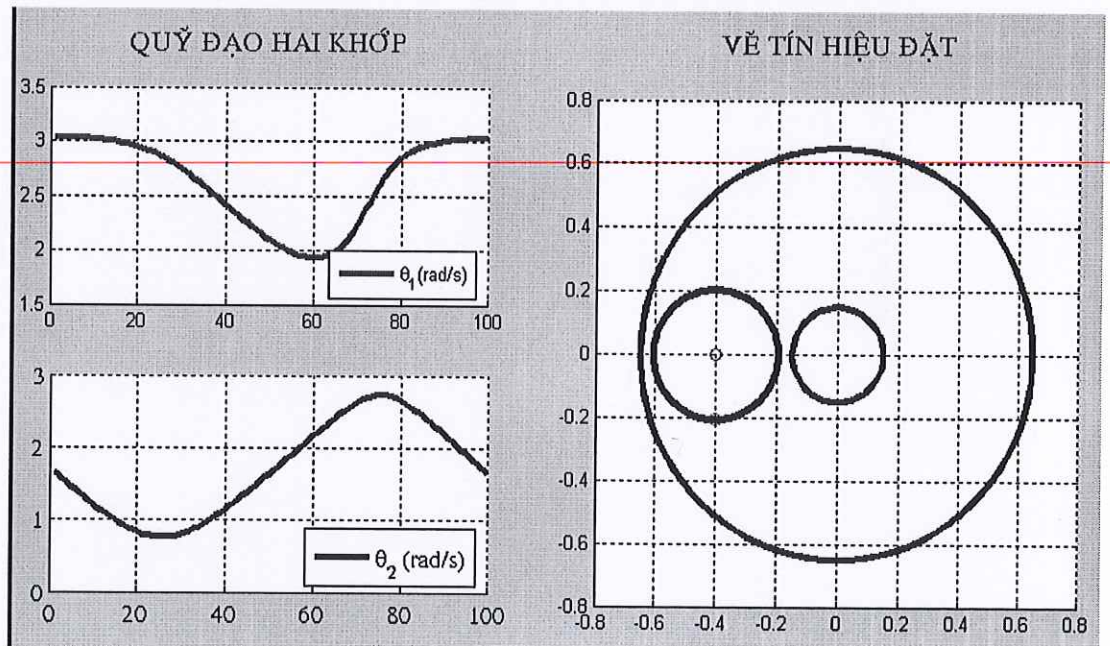
Hình 5.30 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tam giác



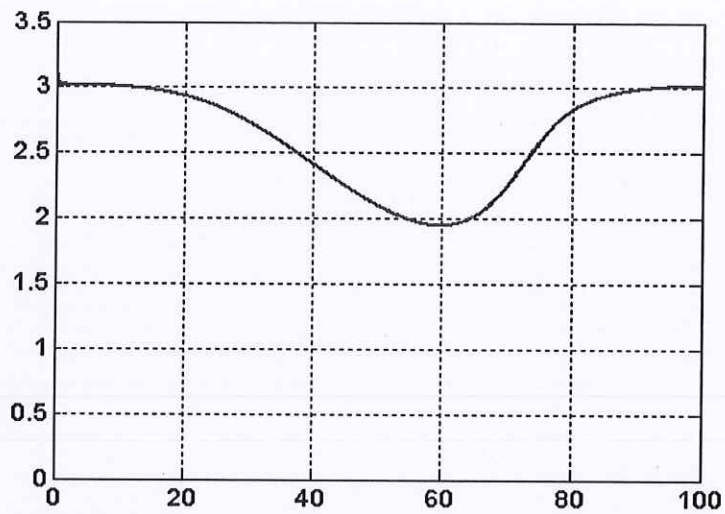
Hình 5.31 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tam giác

Nhận xét: Đáp ứng ngõ ra θ_1, θ_2 ít dao động, ít vọt lố, thời gian đáp ứng nhanh, sai số nhỏ, đáp ứng của cơ cấu chấp hành có sai số nhỏ

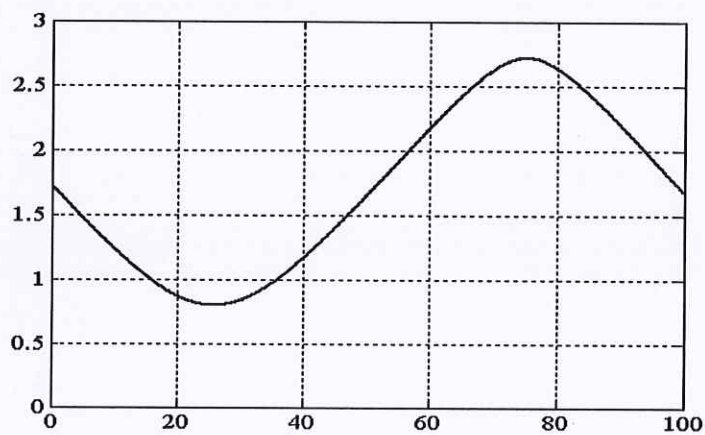
+ Trường hợp 2: Thông số cánh tay robot: $a_1 = 0.4m, a_2 = 0.25m, m_1 = 2.7kg, m_2 = 1.6kg, m_t = 0.5kg$



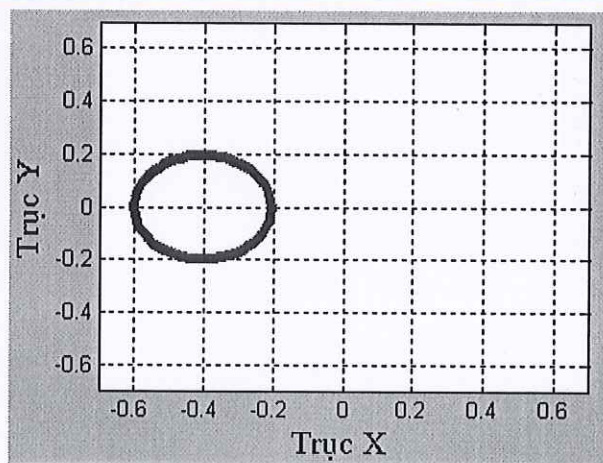
Hình 5.32 Tín hiệu đặt là mẫu cắt hình tròn



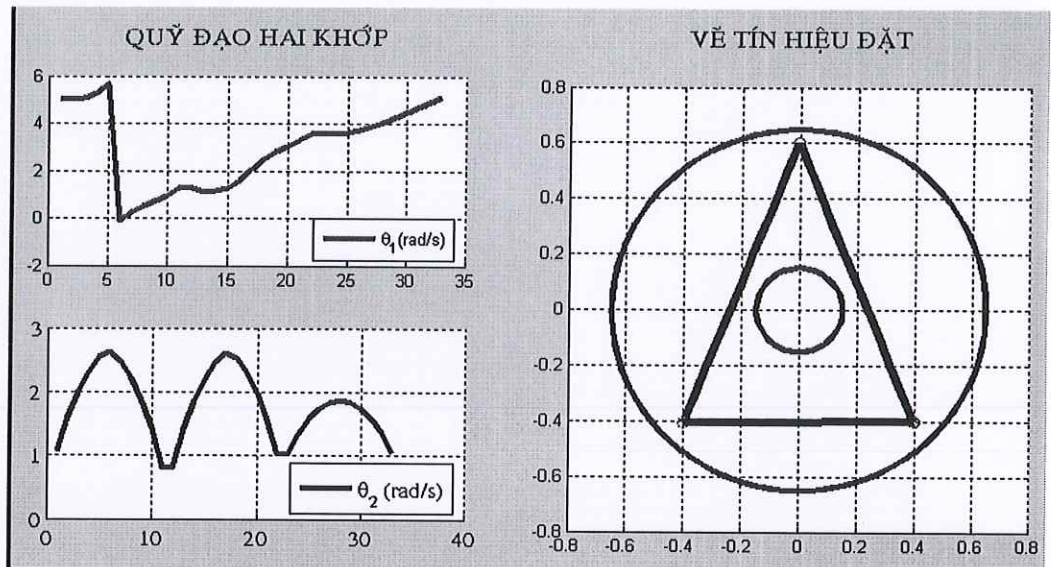
Hình 5.33 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn



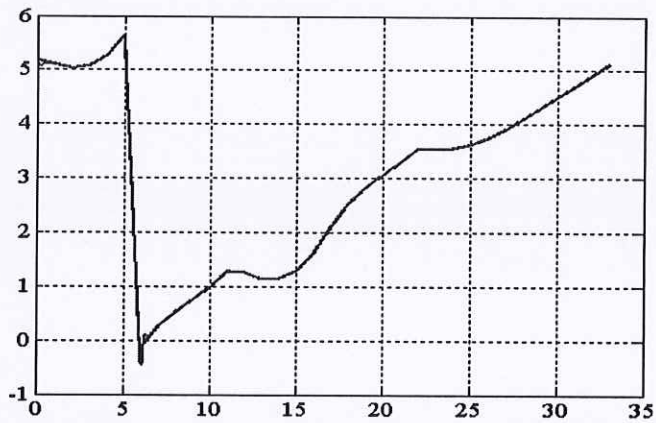
Hình 5.34 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn



Hình 5.35 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn



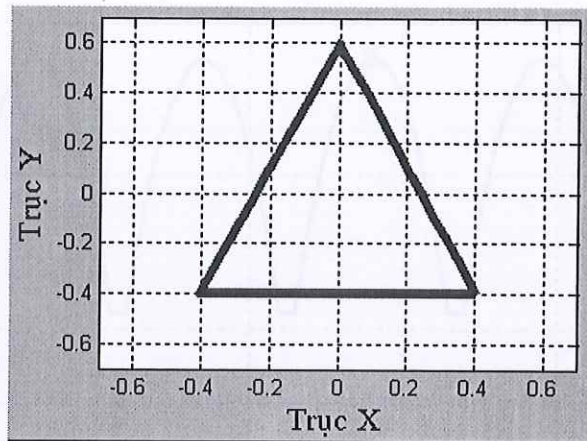
Hình 5.36 Tín hiệu đặt là mẫu cắt hình tam giác



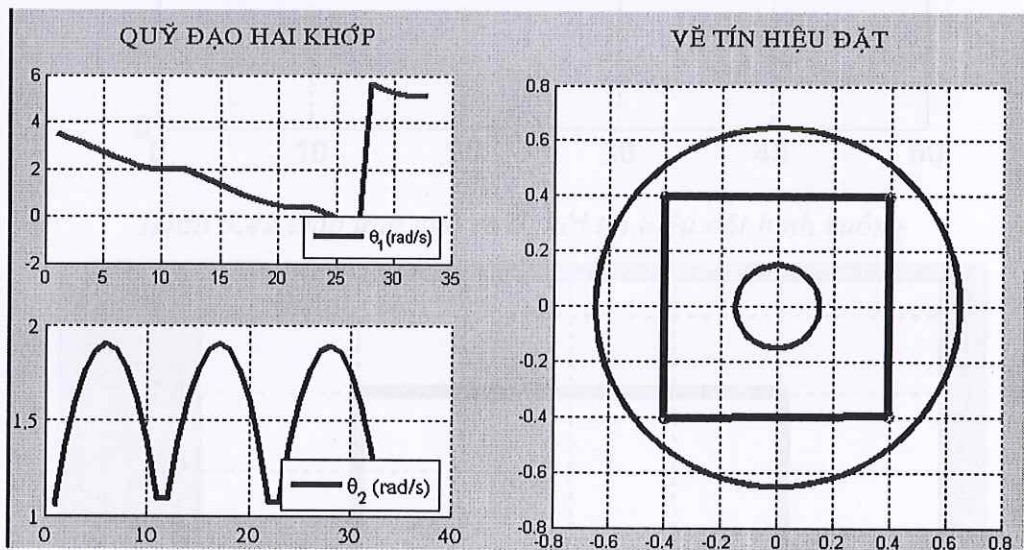
Hình 5.37 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tam giác



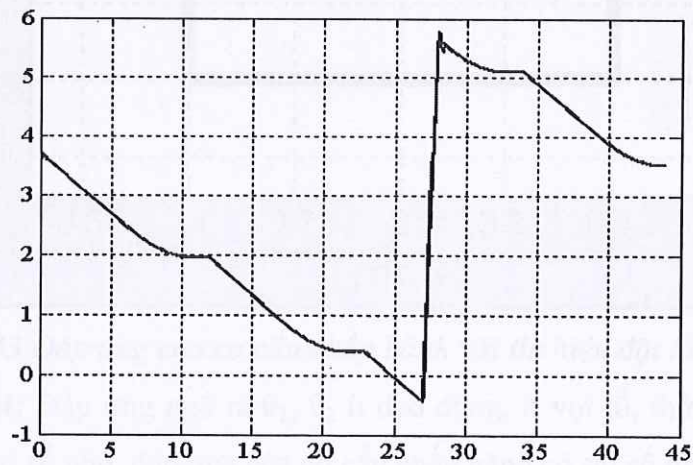
Hình 5.38 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tam giác



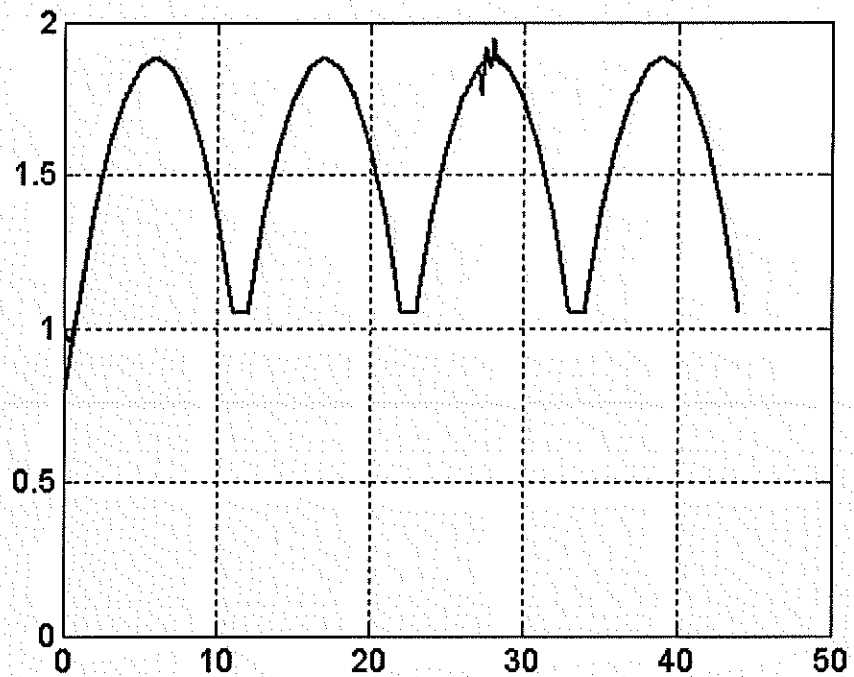
Hình 5.39 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tam giác



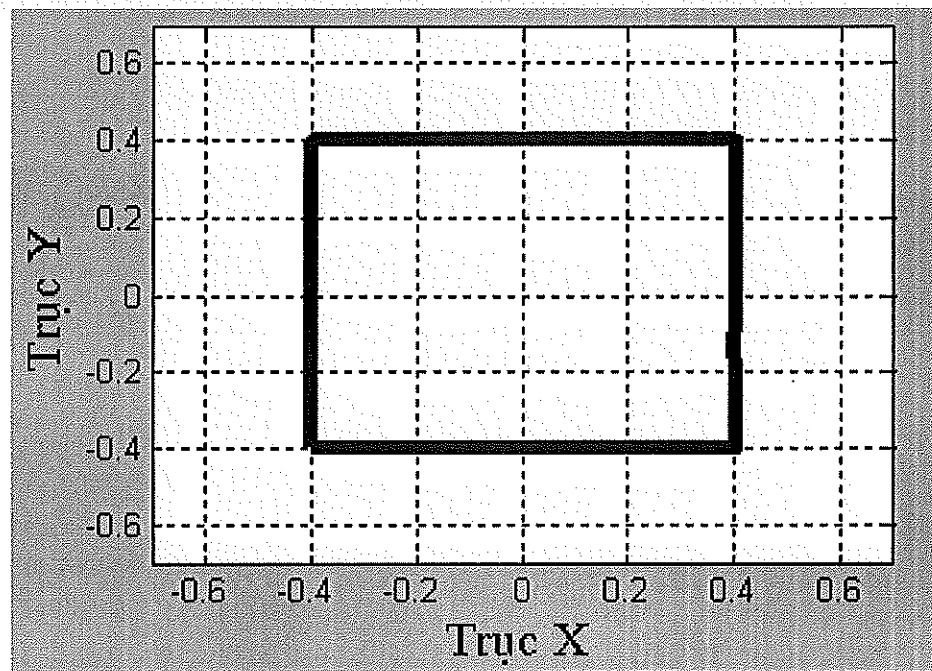
Hình 5.40 Tín hiệu đặt là mẫu cắt hình vuông



Hình 5.41 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình vuông



Hình 5.42 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình vuông

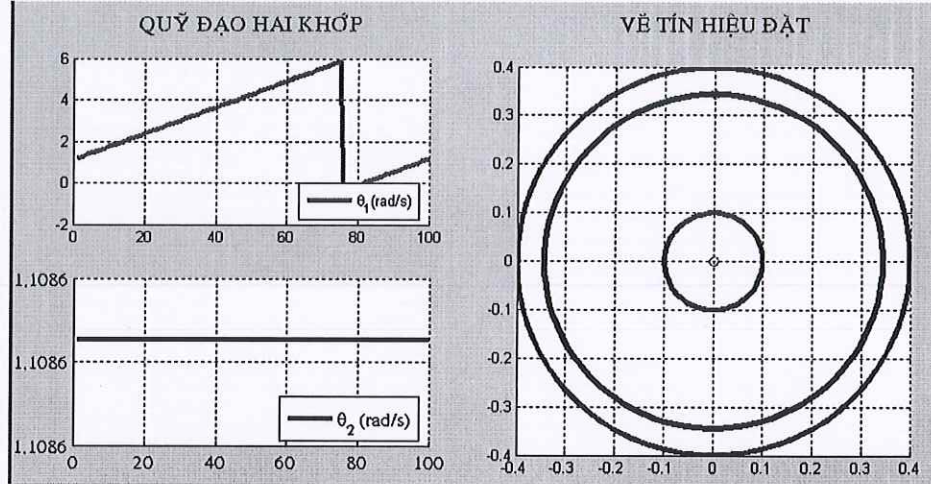


Hình 5.43 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình vuông

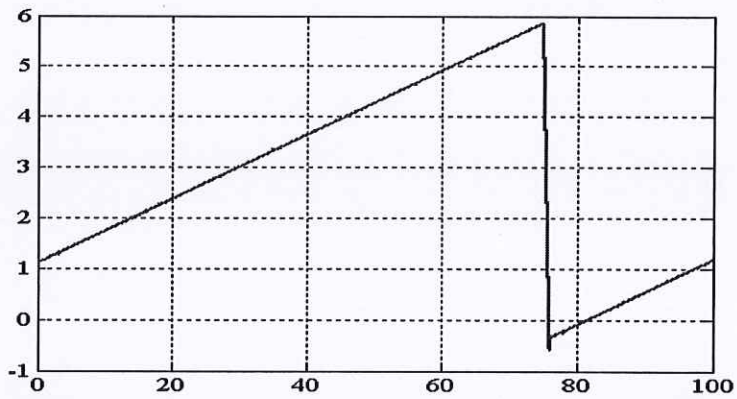
Nhận xét: Đáp ứng ngõ ra θ_1 , θ_2 ít dao động, ít vọt lố, thời gian đáp ứng nhanh, sai số nhỏ, đáp ứng của cơ cấu chấp hành có sai số nhỏ

5.3.3 Kết quả mô phỏng khi có nhiễu tác động ở ngõ ra:

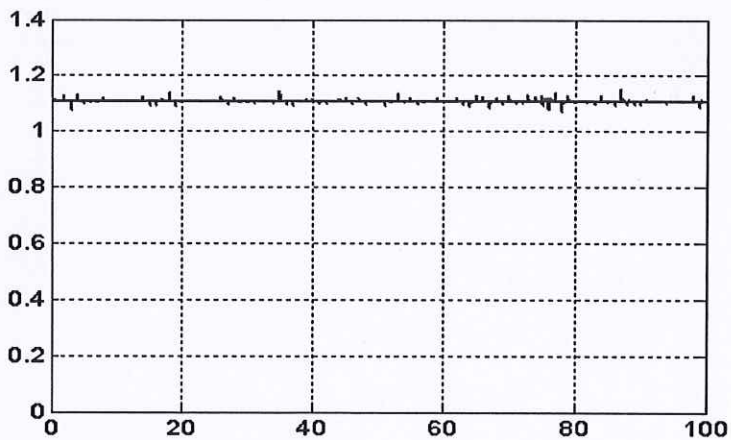
Trường hợp 1: Xét ảnh hưởng của nhiễu với chu kỳ lấy mẫu 1s:



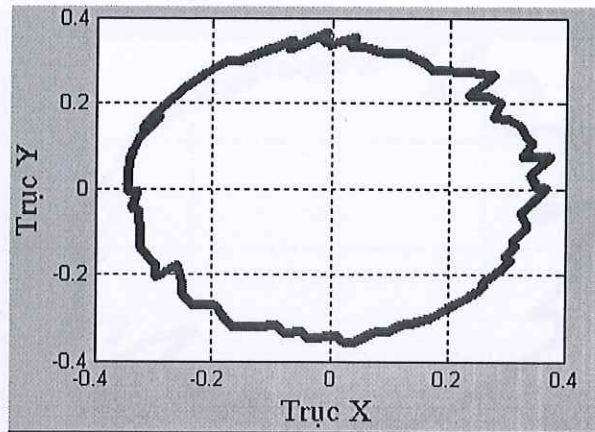
Hình 5.44 Tín hiệu đặt là mẫu cắt hình tròn



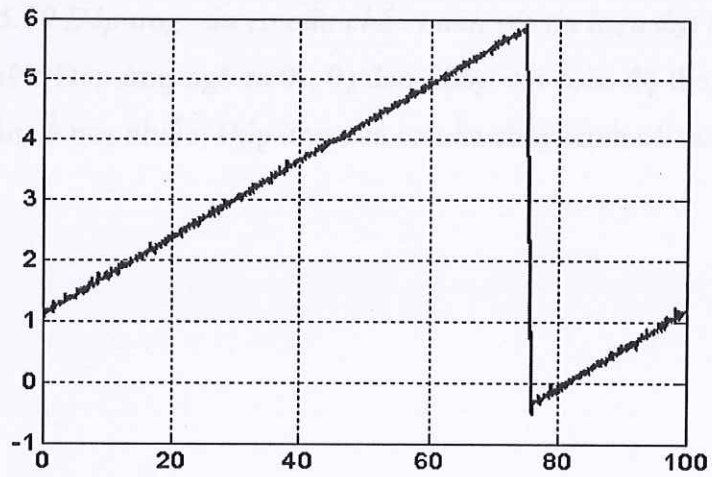
Hình 5.45 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn



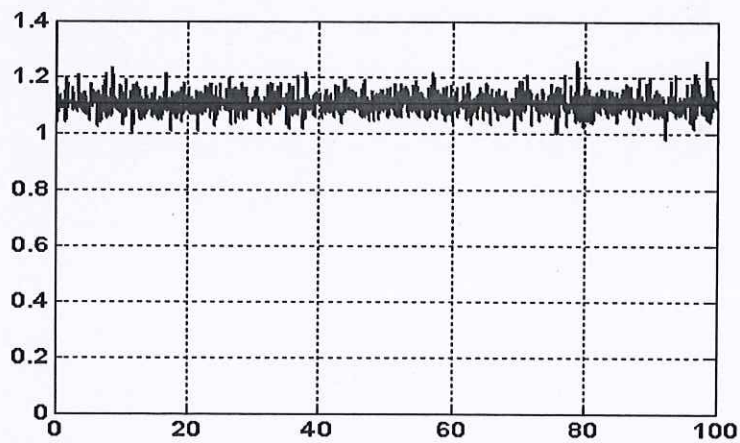
Hình 5.46 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn



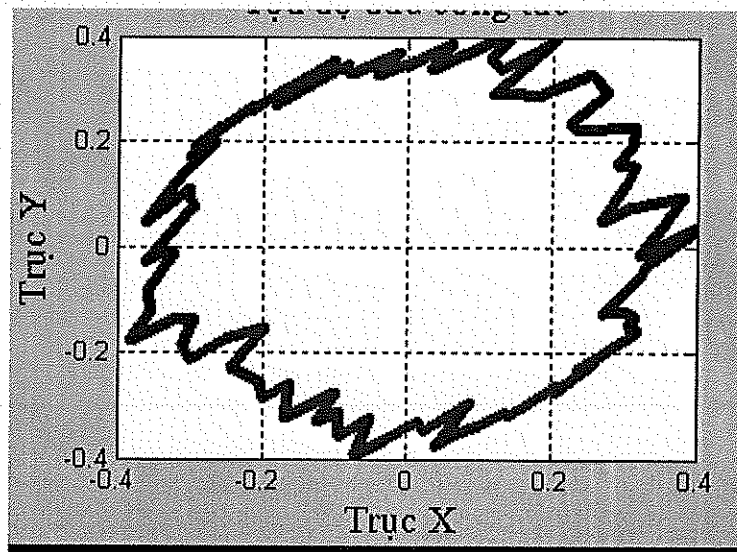
Hình 5.47 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn
 Trường hợp 2: Xét ảnh hưởng của nhiễu với chu kỳ lấy mẫu 0.1s:



Hình 5.48 Đáp ứng ngõ ra θ_1 với tín hiệu đặt hình tròn



Hình 5.49 Đáp ứng ngõ ra θ_2 với tín hiệu đặt hình tròn



Hình 5.50 Đáp ứng của cơ cấu chấp hành với tín hiệu đặt hình tròn
Nhận xét: Đáp ứng ngõ ra θ_1, θ_2 dao động với biên độ thay đổi theo biên độ và tần số của nhiễu, Đáp ứng của cơ cấu chấp hành có sai số lớn

CHƯƠNG VI:

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN ĐỀ TÀI

6.1 Kết luận

Luận văn đã thực hiện được các việc sau:

- + Xây dựng mô hình mô phỏng cánh tay robot 2 khớp nối
- + Thiết kế bộ điều khiển trực tiếp dùng mạng RBF điều khiển cánh tay robot cắt mẫu
- + Xây dựng giải thuật huấn luyện mạng RBF nhận dạng mô hình toán của đối tượng
- + Thiết kế bộ điều khiển mô hình nội dùng mạng RBF điều khiển cánh tay robot cắt mẫu
- + Các mô phỏng cho thấy hệ thống điều khiển đã thiết kế đáp ứng được yêu cầu đặt ra. Bộ điều khiển rất thích hợp cho các bài toán đeo bám mục tiêu

Tuy nhiên, đề tài còn một số hạn chế sau:

- + Do kiến thức matlab còn hạn chế nên trong giao diện vẽ tín hiệu đặt chưa đạt được các mẫu cắt hình elip chính xác.
- + Chưa khắc phục được nhiễu tác động ở ngõ ra

6.2 Hướng phát triển đề tài

- + Khi có nhiễu tác động đến hệ thống làm hệ thống kém ổn định. Việc điều chỉnh các thông số bộ điều khiển đòi hỏi có kinh nghiệm mới khắc phục được nhiễu. Vì vậy thuật toán có thể nghiên cứu thêm về ảnh hưởng của nhiễu đến giải thuật cập nhật trọng số của mạng. Thuật toán có thể tìm ra các giới hạn về nhiễu để trong giới hạn này hệ thống vẫn ổn định.
- + Dùng phần mềm vẽ chính xác hơn (Autocad) để vẽ tín hiệu đặt rồi nội suy tọa độ 2 khớp từ file Autocad này. Khi đó có thể đặt được các mẫu cắt bất kỳ.
- + Một phương án được đề nghị là dùng giải thuật di truyền để tìm ra bộ trọng số tối ưu hơn.
- + Thiết kế và thi công mô hình thực để kiểm chứng lý thuyết.

TÀI LIỆU THAM KHẢO

I. Các tài liệu tham khảo tiếng Việt:

- [1] Bùi Công Cường, “Hệ mờ, mạng neural và ứng dụng”, NXB Khoa học và kỹ thuật
- [2] Nguyễn Thị Phương Hà, Huỳnh Thái Hoàng , “Lý thuyết điều khiển tự động ”, NXB ĐH Quốc Gia Tp.HCM, 2005.
- [3] Huỳnh Thái Hoàng, “Hệ thống điều khiển thông minh”, NXB ĐH Quốc gia TP.HCM, 2006.
- [4] Dương Hoài Nghĩa, “Công nghệ tính toán mềm”
- [5] Nguyễn Phùng Quang, “Matlab & Simulink Dùng cho kỹ sư điều khiển tự động” ,NXB KH&KT, Hà Nội, VN, 2005.
- [6] Lê Hoài Quốc, “Kỹ thuật người máy phần I - Robot công nghiệp”, NXB Đại Học Quốc Gia TPHCM 2005.
- [7] Nguyễn Đức Thành, “Matlab và ứng dụng trong điều khiển”, NXB Đại Học Quốc Gia TPHCM, 2004.
- [8] Nguyễn Thiện Thành, “Mạng Neuron, Nhận Dạng, Dự Báo và Điều Khiển”

II. Các tài liệu tham khảo tiếng Anh:

- [1] Chin-Teng Lin and C.S. George Lee, “Neural Fuzzy Systems”, Prentice Hall, 1996.
- [2] Cornelius T. Leondes, “Fuzzy Logic and Expert Systems Applications”, Academic Press, 1998.
- [3] Duc Truong Pham and Liu Xing, “Neural Networks for Identification, Prediction and Control”, Springer, 1997.
- [4] Frank L.Lewis, Darren M.Dawson, Chaouki T.Abdallah, “Robot Manipulator Control Theory and Practice”, Marcel Dekker, INC, 2004.
- [5] Heikki Koivo, “ANFIS (Adaptive Neuro-Fuzzy Inference System)”, 2000.
- [6] Katalin M.Hangos, Rozalia Lakner, Miklos Gerzson, “Intelligent Control systems”, Kluwer Academic Publisher,2001.

- [7] Katsuhiko Ogata, "Modern Control Engineering", Third Edition, Prentical Hall, 1997.
- [8] K.-L. Du and M.N.S. Swamy, "Neural Networks in a Softcomputing Framework", Springer, 2006.
- [9] L. X. Wang, "Adaptive Fuzzy Systems and Control", Prentice-Hall, Englewood Cliffs, 1994.
- [10] M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, "Neural and Fuzzy Logic Control of Drives and Power Systems", Newnes, 2002.
- [11] M. Vidyasagar, "Nonlinear system Analysis", Prentice Hall, 1993
- [12] Robin M. Murphy, "Introduction to AI Robotics", Massachusetts Institute of Technology, 2000.
- [13] Simon Haykin, "Neural Networks A Comprehensive Foundation", Prentice Hall 1999
- [14] Y.G Leu, T.T. Lee, W.Y. Wang, "Robust adaptive fuzzy-neural controllers for uncertain nonlinear systems", IEEE Trans. Robotics Automat. 15 (1999) 805- 817.

III. Các bài báo:

- [1] Dorin Popescu, Dan Selisteanu and Cosmin Ionete, "On Adaptive and Neural Control of a Robot Arm", University of Craiova, Faculty of Automation, Computer and electronics, Craiova.
- [2] Jafa Tavoosi, Majid Alaei, Behrouz Jahani, "Neuron – Fuzzy Controller for Position Control of Robot Arm", Faculty of Electrical and Computer Engineering University of Tabriz, Tabriz, Iran, 2011.
- [3] Tamer A. Alzohairy, "Direct Adaptive Control of Unknown Nonlinear Systems Using Radial Basis Function Networks with Gradient Descent and K-means", International Journal of Computer Theory and Engineering, Vol. 3, No. 6, December 2011.

PHỤ LỤC

1. Chương trình m file của mô hình cánh tay robot:

```
function [sys,x0,str,ts] = mht_Robot(t,x,u,flag,m1,m2, mt,a1,a2)
%m1=2.5;m2=1.0; mt=0.5;a1=0.25;a2=0.15;
switch flag,
case 0
    [sys,x0,str,ts]=mdlInitializeSizes;
case 3
    sys=mdlOutputs(t,x,u,m1,m2,mt,a1,a2);
case { 1, 2, 4}
    sys=[];
case { 9 }
    sys=[];

otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [sys,x0,str,ts] = mdlInitializeSizes()
    sizes = simsizes;
    sizes.NumContStates = 0;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 4;
    sizes.NumInputs = 2;
    sizes.DirFeedthrough = 1;
    sizes.NumSampleTimes = 1; % at least one sample time is needed
    sys = simsizes(sizes);
    str = [];
    x0 = [];
    ts = [0.001 0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%clc
global Xo Ts qo1 qo2
    dq1 = 0; dq2 = 0;
    Xo = [qo1;dq1;qo2;dq2]; Ts = 0.001;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function sys = mdlOutputs(t,x,u,m1,m2,mt,a1,a2)
global Xo Ts
    C1 = cos(Xo(1)); C2 = cos(Xo(3)); C12 = cos(Xo(1)+Xo(3));
    S1 = sin(Xo(1)); S2 = sin(Xo(3)); S12 = sin(Xo(1)+Xo(3));
% Ma trận moment quán tính
    H11 = (m1+m2+mt)*a1^2 + (m2+mt)*a2^2 + 2*(m2+mt)*a1*a2*C2;
```

```

H12 = (m2+mt)*a2^2 + (m2+mt)*a1*a2*C2;
H21 = H12;
H22 = (m2+mt)*a2^2 ;
H = [H11,H12;H21,H22];
% moment nhót và hướng tâm
v(1) = -(m2+mt)*a1*a2*S2*Xo(4)^2 - 2*(m2+mt)*a1*a2*S2*Xo(2)*Xo(4);
v(2) = (m2+mt)*a1*a2*S2*Xo(2)^2 ;
V = [v(1);v(2)];
% moment trọng lực
g1 = (m1+m2+mt)*9.81*a1*C1+(m2+mt)*9.81*a2*C12;
g2 = (m2+mt)*9.81*a2*C12;
G = [g1;g2];
% Nghịch đảo H
Hinv = inv(H);
% Vận tốc khớp
DX = -Hinv*(V+G) + Hinv*u;
% Phương trình trạng thái
dx(1) = Xo(2);
dx(2) = DX(1);
dx(3) = Xo(4);
dx(4) = DX(2);

Xo = Xo + Ts*dx';
Px = a1*C1 + a2*C12;
Py = a1*S1 + a2*S12;
sys = [Xo(1); Px; Py; Xo(3)];
%=====

```

2. Chương trình m file của mô hình mạng neural điều khiển:

```

function [sys,x0,str,ts] = neuron_s(t,x,u,flag,no,n1,n2,eta,Ts,lr)
switch flag,
case 0
    [sys,x0,str,ts]=mdlInitializeSizes(n1,n2,Ts);
case 3
    sys=mdlOutputs(t,x,u,n1,n2,eta,lr);
case { 1, 2, 4 }
    sys=[];
case { 9 }
    sys=[];
otherwise
    error(['Unhandled flag = ',num2str(flag)]);
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [sys,x0,str,ts] = mdlInitializeSizes(n1,n2,Ts)

```

```

sizes = simsizes;
sizes.NumContStates = 0;
sizes.NumDiscStates = 0;
sizes.NumOutputs = n2;
sizes.NumInputs = n1;
sizes.DirFeedthrough = 1;
sizes.NumSampleTimes = 1; % at least one sample time is needed
sys = simsizes(sizes);

str = [];
x0 = [];
ts = [Ts 0];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global W c sigma b2
load('trong_so.mat','W','c','sigma','b2')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function sys = mdlOutputs(t,x,u,n1,n2,eta,lr)
global W c sigma b2
for n=1:n2
    a_1(n,:) = sqrt(((u - c(n,:))'*(u - c(n,:))));
    a_2(:,n) = (sigma(n,:))';
    net_q(:,n) = a_1(n,:)./a_2(:,n);
end
zq = radbas(net_q);
for n=1:n2
    uk_net(n,:) = (W(n,:)*zq(:,n)) + b2(n);
end
sys = uk_net;
erro = [u(2) + 0.3*u(3), u(5) + 0.25*u(4)];
if 0.5*sum(erro.^2) > lr
    for n=1:n2
        errh(n,:) = erro(n)*W(n,:)*(a_1(n,:)/(a_2(:,n).^2))*zq(:,n);
        c(n,:) = c(n,:)+eta(n)*errh(n,:);
        errh1(n,:) = erro(n)*W(n,:)*((a_1(n,).^2)/(a_2(:,n).^3))*zq(:,n);
        sigma(n,:) = sigma(n,:)+ eta(n)* errh1(n,:);

        W(n,:) = W(n,:) + (eta(n)*erro(n)* zq(:,n))';
        b2(n) = 400*eta(n)*erro(n);
    end
end
end

```

Chương trình m file cập nhật tâm, hệ số trái, trọng số:

```
clc
load('input.mat','PR')
load('output.mat','Y')
x =PR';
d = Y';
d1=Y(1,:);% bkhop 1 them vo
d2 = Y(2,:); % bkhop 2 them vo
N = size(d,1);
load('trong_so_1.mat','W_T','C_T','SIGMA_T')
eta = 0.3;
epoch = 0;
while epoch < 10
    for bk=1: N
        a3(bk,:) = sqrt((x(bk,:)' - C_T)'*(x(bk,:)' - C_T));
        a4(:,bk) = (SIGMA_T);
        net_q_T(:,bk) = a3(bk,:)./a4(:,bk); % 4 hang 500 cot
        zq_T(:,bk) = radbas(net_q_T(:,bk));% matran 4 hang 500 cot
        y(:,bk) = W_T*zq_T(:,bk); % 2 hang 500 cot
        for p1 = 1:2
            erro_T(bk,p1) = d(bk,p1) - y(p1,bk)'; % ma tran 500 hang 2 cot
        end
        saiso(bk,:) = erro_T(bk,1) + erro_T(bk,2);%500 hang 1 cot
        J(bk,:) = 0.5*(erro_T(bk,1)^2 + erro_T(bk,2)^2);
        test1(bk,:) = erro_T(bk,:)*W_T;
        test2(bk,:) = test1(bk,:)*net_q_T(:,bk);
        test2_1(:,bk) = test2(bk,:);
        test2_2(bk,:) = zq_T(:,bk)';
        test2_1(:,bk)*test2_2(bk,:);
        errh_T = test2_1(:,bk)*test2_2(bk,:);
        C_T = C_T+eta*test1(bk,:)*(a3(bk,:)./(a4(:,bk).^2))*zq_T(:,bk)';
        errh_T1 = test1(bk,:)*((a3(bk,:).^2)./(a4(:,bk).^3))*zq_T(:,bk)';
        SIGMA_T = SIGMA_T+ eta* errh_T1;
        test3(bk,:)= eta*erro_T(bk,:);
        test3_1(:,bk)= test3(bk,:);
        test3_2(bk,:) = zq_T(:,bk)';
        test3_1(:,bk)*test3_2(bk,:);
        W_T = W_T + test3_1(:,bk)*test3_2(bk,:); %ma tran 500 hang 4 cot
    end
    epoch = epoch + 1;
end
SIGMA_T;
C_T;
```

```

W_T;
bk;
erro_T;
saiso;
J;
T = 0.2:0.2:100;
subplot(411), plot(T,d1,'k',T,y(1:),'r');grid;legend('d1','y1')
subplot(412), plot(T,d2,'k',T,y(2:),'r');grid;legend('d2','y2')
subplot(413), plot(T,erro_T(:,1),'r');grid;legend('e1')
subplot(414), plot(T,erro_T(:,2),'r');grid;legend('e2')
disp('nhận dạng xong')

```

3. Chương trình m file tạo giao diện:

```

function varargout = giaodien(varargin)
    gui_Singleton = 1;
    gui_State = struct('gui_Name',    mfilename, ...
        'gui_Singleton', gui_Singleton, ...
        'gui_OpeningFcn', @giaodien_OpeningFcn, ...
        'gui_OutputFcn', @giaodien_OutputFcn, ...
        'gui_LayoutFcn', [], ...
        'gui_Callback', []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback = str2func(varargin{1});
    end
    if nargout
        [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State, varargin{:});
    end
%-----
function giaodien_OpeningFcn(hObject, eventdata, handles, varargin)
    handles.output = hObject;
    guidata(hObject, handles);
%-----
function varargout = giaodien_OutputFcn(hObject, eventdata, handles)
    varargout{1} = handles.output;
    global r loai_ve a1 a2 m dta_2 dta_1 q
    loai_ve = 0; m = 0; dta_2 = []; dta_1 = []; q = [];
    a1 = 0.25; a2 = 0.15;
    L1 = link([0 a1 0 0 0]);
    L2 = link([0 a2 0 0 0]);
    r = robot( {L1, L2}, 'Robot 2D') ;%n = r.n
    gioi_han(handles);
%-----

```

```

function gioi_han(handles)
    global a1 a2
    clc
    t = linspace(-2*pi,2*pi);
    x = sin(t); y = cos(t);
    xc1 = (a1+a2)*x; yc1=(a1+a2)*y; xc2 =(a1-a2)*x;yc2 = (a1-a2)*y;
    axes(handles.axes_dat)
    set(handles.axes_dat,'NextPlot','Add')
    plot(xc1,yc1,'linewidth',3)% hold on
    plot(xc2,yc2,'linewidth',3)
%-----
function p_ve_Callback(hObject, eventdata, handles)
    global loai_ve
    axes(handles.axes_dat)
    set(handles.axes_dat,'NextPlot','ReplaceChildren')
    plot(0,0)
    gioi_han(handles);
%-----
    xy = [];n = 0; but = 1; xy_L = []; m = 0; xys_L = [];
    while (but == 1&loai_ve~=0)
        [xi,yi,but] = ginput(1);
        q = teta(handles,[xi,yi]);
        if isempty(find(imag(q(:,1))))==1 & isempty(find(imag(q(:,2))))==1
            plot(xi,yi,'ro')
            n = n+1;
            xy(:,n) = [xi;yi];
            %%%%%%%%%%
            if(loai_ve==2)
                m = m+1;
                xy_L(:,m) = [xi;yi];
                if(mod(n,2)==0)
                    xys = linear(handles,xy_L);
                    plot(xys(:,1),xys(:,2),'k','linewidth',5);
                    m = 0;
                    xys_L = [xys_L; xys];
                    xys = xys_L;
                end
            end
        end
        %%%%%%%%%%
    end
end
if(loai_ve==1) % nội suy phi tuyến
    xys = cubic(handles,xy,n);

```

```

        plot(xys(:,1),xys(:,2),'color','k','linewidth',5);
    end
    while (but == 1 & loai_ve == 0)
        [xi,yi,but] = ginput(1);
        q = teta(handles,[xi,yi]);
        plot(xi,yi,'ro')
        n = n+1;
        xy(:,n) = [xi,yi];
    end
    if(n<2)
        errordlg('POINT >= 2','ERROR')
        return
    end

    if(loai_ve == 0) % nội suy vòng tròn
        xys = circle(handles,xy);
        plot(xys(:,1),xys(:,2),'color','k','linewidth',4);
    end
    DHD(handles,xys)
    %-----
    function p_close_Callback(hObject, eventdata, handles)
        closereq
    %-----
    function p_sim_Callback(hObject, eventdata, handles)
        global q r
        clc
        if isempty(q)
            errordlg('STOP','ERROR')
        return
        end
        h = figure(1);
        set(h,'numbertitle','off','name','Mo hình Robot 2D',
            'menubar','none','color','w','position',[10 300 500 400])
        title('Quỹ đạo đầu công tác','color','r','Fontname','VNI-Times','FontSize',20)
        xlabel('x','color','r','Fontname','VNI-Times','FontSize',20)
        ylabel('y','color','r','Fontname','VNI-Times','FontSize',20)
        zlabel('z','color','r','Fontname','VNI-Times','FontSize',20)
        plot(r,[q;[0,0]])
    %-----
    function [nPx,nPy]=DHT(q)
        global a1 a2
        teta1 = q(:,1); teta2 = q(:,2);
        nPx = a1*cos(teta1) + a2*cos(teta1 + teta2) ;

```

```

nPy = a1*sin(teta1) + a2*sin(teta1 + teta2) ;
%-----
function DHD(handles,xys)
    global q t q1 q2
    q = teta(handles,xys);
    if find(imag(q(:,1)))~=0 | find(imag(q(:,2)))~=0
        error('STOP','ERROR')
    return
end
% Vẽ quỹ đạo khớp
q1 = q(:,1); q2 = q(:,2);
axes(handles.axes_q1)
t = (1:length(q1));
plot(t,q(:,1),'linewidth',3,'color','r') % Vẽ nội suy
legend('\theta_1^ (rad/s)',4)
axes(handles.axes_q2)
plot(t,q(:,2),'linewidth',3,'color','b') % Vẽ nội suy
legend('\theta_2^ (rad/s)',4)
set(gca,'Fontname','VNI-Times','FontSize',13)
% Lưu file
control_file(xys);
%-----
function q = teta(handles,xys)
    global a1 a2
% Động học đảo
Px = xys(:,1);
Py = xys(:,2);
d_2 = Px.^2 + Py.^2 ;
% Tính teta_2
teta_2 = acos((d_2-a1^2-a2^2)/(2*a1*a2)) ;
% Tính teta_1
gama = atan2(-Py./sqrt(d_2),-Px./sqrt(d_2));
teta_1 = gama + acos((-d_2+a1^2-a2^2)/(2*a1*sqrt(d_2)));
q = [teta_1,teta_2];
%-----
function r_cubic_Callback(hObject, eventdata, handles)
    global loai_ve
    loai_ve = 1;
%-----
function r_linear_Callback(hObject, eventdata, handles)
    global loai_ve
    loai_ve = 2;
%-----

```

```

function p_open_Callback(hObject, eventdata, handles)
    global qo1 qo2 Px Py
    clc
    load toa_do_q1.mat sig1
    load toa_do_q2.mat sig2
    q1 = sig1(2,:); q2 = sig2(2,:);
    qo1=q1(1,1); qo2=q2(1,1);
    t = (1:length(q1));
    Len = length(t);
    q = [q1;q2]';
    [Px,Py]=DHT(q);
    axes(handles.axes_dat)
    plot(Px,Py,'ko','linewidth',2);
% Vẽ quỹ đạo khớp
    q1 = q(:,1); q2 = q(:,2);
    axes(handles.axes_q1)
    plot(t,q(:,1),'linewidth',3,'color','r') % Vẽ nội suy
    set(handles.axes_q1,'NextPlot','Add')
    plot(t,q(:,2),'linewidth',3,'color','b') % Vẽ nội suy
    set(handles.axes_q1,'NextPlot','ReplaceChildren')
    legend('\theta_1^ (rad/s)', '\theta_2^ (rad/s)',4)
    set(gca,'Fontname','VNI-Times','FontSize',13)

    set_param('neuron_controller', 'StopTime', num2str(Len))
    set_param('neuron_controller', 'StartTime', num2str(0.001))
%-----
function r_circle_Callback(hObject, eventdata, handles)
    global loai_ve
    loai_ve = 0;
%-----
function p_s_Callback(hObject, eventdata, handles)
    sim('neuron_controller')
%-----
% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton8 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```