**BỘ GIÁO DỤC & ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐIỆN – ĐIỆN TỬ**

**BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN**

**-----------------⸙∆⸙-----------------**



**BÁO CÁO CUỐI KỲ**

**MÔN HỌC: ĐIỀU KHIỂN THÔNG MINH**

**ĐỀ TÀI: THIẾT KẾ BỘ ĐIỀU KHIỂN THÔNG MINH CHO HỆ CÁNH TAY ROBOT 2 BẬC TỰ DO**

**GVHD: TS. Trần Đức Thiện**

**SVTH: Nguyễn Xuân Trà 19151299**

**Nguyễn Đức Mạnh 19151253**

**Nguyễn Ngọc Thiện 19151292**

**Tp. Hồ Chí Minh tháng 6 năm 2022**

# Mục lục

[Mục lục i](#_Toc107408735)

[Danh sách hình ảnh ii](#_Toc107408736)

[Danh sách bảng iii](#_Toc107408737)

[Chương 1. THIẾT KẾ BỘ ĐIỀU KHIỂN PD MỜ 1](#_Toc107408738)

[1.1 Phương trình vi phân của hệ robot 2 bậc tự do 1](#_Toc107408739)

[1.2 Cơ sở lý thuyết của bộ điều khiển mờ Fuzzy 2](#_Toc107408740)

[1.3 Xây dựng bộ điều khiển mờ trên Fuzzy Toolbox 3](#_Toc107408741)

[1.4 Xây dựng file mô phỏng trên Matlab Simulink 5](#_Toc107408742)

[1.5 Thu thập và trình bày kết quả 8](#_Toc107408743)

[1.6 Phân tích kết quả mô phỏng 11](#_Toc107408744)

[Chương 2. THIẾT KẾ MẠNG THẦN KINH XẤP XỈ BỘ ĐIỀU KHIỂN MỜ 13](#_Toc107408745)

[2.1 Cơ sở lý thuyết về mạng thần kinh nhân tạo 13](#_Toc107408746)

[2.2 Thu thập dữ liệu 14](#_Toc107408747)

[2.3 Thiết kế bộ điều khiển 16](#_Toc107408748)

[2.4 Kết quả mô phỏng 19](#_Toc107408749)

[Chương 3. SỬ DỤNG GIẢI THUẬT GA ĐỂ XÁC ĐỊNH CÁC THÔNG SỐ BỘ ĐIỀU KHIỂN PID TỐI ƯU 22](#_Toc107408750)

[3.1 Lưu đồ giải thuật di truyền 22](#_Toc107408751)

[3.2 Chương trình giải thuật di truyền bộ thông số PID 28](#_Toc107408752)

[3.3 Mô phỏng Simulink và kết quả 31](#_Toc107408753)

[TÀI LIỆU THAM KHẢO 36](#_Toc107408754)

# Danh sách hình ảnh

[Hình 1 Hình robot hai bậc tự do 1](#_Toc107408669)

[Hình 2 Bộ điều khiển mờ trên Fuzzy Toolbox 3](#_Toc107408670)

[Hình 3 Hàm liên thuộc ngõ vào sai số E 3](#_Toc107408671)

[Hình 4 Hàm liên thuộc ngõ vào sai số vi phân DE 4](#_Toc107408672)

[Hình 5 Hàm liên thuộc ngõ ra U 4](#_Toc107408673)

[Hình 6 Bộ điều kiển PD mờ cho hệ cánh tay máy 2 bậc trên Matlab Simulink 5](#_Toc107408674)

[Hình 7 Đối tượng điều khiển cánh tay máy 2 bậc tự do 5](#_Toc107408675)

[Hình 8 Khối Fuzzy Logic Control 6](#_Toc107408676)

[Hình 9 Khối Saturation 7](#_Toc107408677)

[Hình 10 Khối Transfer Fcn 7](#_Toc107408678)

[Hình 11 Khối Setpoint 8](#_Toc107408679)

[Hình 12 Đáp ứng ngõ ra tại khớp thứ nhất của Robot 2 DoF 9](#_Toc107408680)

[Hình 13 Tín hiệu điều khiển tại khớp thứ nhất của Robot 2 DoF 9](#_Toc107408681)

[Hình 14 Giá trị sai số tại khớp thứ nhất của Robot 2 DoF 10](#_Toc107408682)

[Hình 15 Đáp ứng ngõ ra tại khớp thứ hai của Robot 2 DoF 10](#_Toc107408683)

[Hình 16 Tín hiệu điều khiển tại khớp thứ hai của Robot 2 DoF 11](#_Toc107408684)

[Hình 17 Giá trị sai số tại khớp thứ hai của Robot 2 DoF 11](#_Toc107408685)

[Hình 18 Mô hình mạng thần kinh kiểu dự báo 13](#_Toc107408686)

[Hình 19 Lưu đồ giải thuật cho quá trình xây dựng một mạng thần kinh nhân tạo 13](#_Toc107408687)

[Hình 20 Cấu trúc hệ thu thập dữ liệu 14](#_Toc107408688)

[Hình 21 Thông số khối workspace 14](#_Toc107408689)

[Hình 22 Các điểm được thu thập dữ liệu 15](#_Toc107408690)

[Hình 23 Dữ liệu được thu thập từ quá trình mô phỏng 15](#_Toc107408691)

[Hình 24 Các biến ở workspace 17](#_Toc107408692)

[Hình 25 Bảng huấn luyện mạng neural ở khớp 1 17](#_Toc107408693)

[Hình 26 Bảng huấn luyện mạng neural ở khớp 2 18](#_Toc107408694)

[Hình 27 Kết quả huấn luyện ở simulink 18](#_Toc107408695)

[Hình 28 Sơ đồ kết nối với bộ điều khiển mạng neural 19](#_Toc107408696)

[Hình 29 Đáp ứng ngõ ra của khớp 1 19](#_Toc107408697)

[Hình 30 Đáp ứng ngõ ra của khớp 2 20](#_Toc107408698)

[Hình 31 Tín hiệu điều khiển của khớp 1 20](#_Toc107408699)

[Hình 32 Tín hiệu điều khiển của khớp 2 20](#_Toc107408700)

[Hình 33 Lưu đồ giải thuật di truyền 22](#_Toc107408701)

[Hình 34 Simulink mô phỏng bộ điều khiển PID 31](#_Toc107408702)

[Hình 35 Bộ điều khiển PID khớp 1 31](#_Toc107408703)

[Hình 36 Bộ điều khiển PID khớp 2 31](#_Toc107408704)

[Hình 37 Khối qui hoạch quĩ đạo hình sin cho cả hai khớp 32](#_Toc107408705)

[Hình 38 Thông số khối sin khớp 1 Hình 39 Thông số khối sin của khớp 2 32](#_Toc107408706)

[Hình 40 Kết quả các thông số PID khi chạy xong chương trình giải thuật GA 33](#_Toc107408707)

[Hình 41 Thay đổi hàm thích nghi qua các thế hệ 33](#_Toc107408708)

[Hình 42 Đồ thị đáp ứng của khớp 1, tín hiệu điều khiển tor1, và sai số so với tín hiệu đặt 1 34](#_Toc107408709)

[Hình 43 Đồ thị đáp ứng của khớp 2, tín hiệu điều khiển tor2, và sai số so với tín hiệu đặt 2 34](#_Toc107408710)

# Danh sách bảng

[Bảng 1 Các qui tắc mờ cho bộ điều khiển cách tay máy 1 bậc tự do 4](#_Toc107258903)

# THIẾT KẾ BỘ ĐIỀU KHIỂN PD MỜ

## Phương trình vi phân của hệ robot 2 bậc tự do

Ta xét robot trong trạng thái bỏ qua ma sát giữa các khớp, và bỏ qua tác động lực vào khớp cuối của robot.



Hình 1 Hình robot hai bậc tự do

Thông số vật lý của cánh tay máy mô phỏng:

: moment quán tính của hai khớp .

: khối lượng của hai khớp .

: chiều dài hai khớp .

: khoảng cách từ trọng tâm của mỗi khớp đến trục quay .

: gia tốc trọng trường .

Động lực học robot 2 DOF dưới dạng tiêu chuẩn:



Với: là ma trận quán tính, là vector hướng tâm/ Coriolis,  là vector trọng lực.

Chúng ta viết động lực học của robot 2 DOF dưới dạng vector như sau:





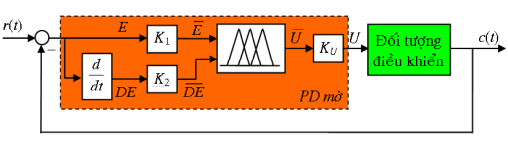




Trong đó: , ,, 

## Cơ sở lý thuyết của bộ điều khiển mờ Fuzzy

Bộ điều khiển PD mờ dùng hệ quy tắc Mandani:



Hình 1. 1 Bộ điều khiển PD mờ

Bộ điều khiển trên là bộ điều khiển PD mờ vì tín hiệu ra của bộ điều khiển mờ cơ bản phụ thuộc vào tín hiệu vào và vi phân của tín hiệu vào. Bộ điều khiển PD mờ có thể điều khiển vô sai trong các trường hợp sau đây:

* Đối tượng có khâu tích phân lí tưởng.
* Ổn định hóa trạng thái của đối tượng xung quanh điểm cân bằng.

Nếu đối tượng không có khâu tích phân lí tưởng (như hệ lò nhiệt, bồn nước,…) thì sử dụng bộ điều khiển PD mờ không thể triệt tiêu sai số xác lập.

Đối tượng điều khiển cánh tay máy hai bậc tự do là hệ MIMO - hai vào hai ra nhưng ta vẫn có thể sử dụng bộ điều khiển mờ SISO – một vào một ra theo nguyên tắc tách rời, có nghĩa là mỗi bộ điều khiển mờ sẽ kiểm soát hoạt động cho mỗi khớp liên kết của Robot.

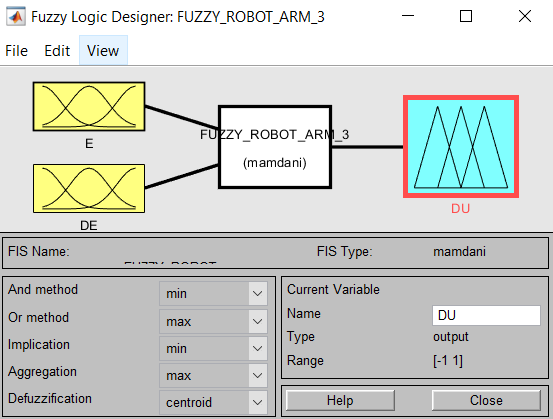
## Xây dựng bộ điều khiển mờ trên Fuzzy Toolbox

Như đã nói ở trên ta vẫn sẽ sử dụng bộ điều khiển mờ SISO theo nguyên tắc tách rời để điều khiển cho hệ cánh tay máy hai bậc tự do MIMO. Vì vậy ta chỉ cần thiết kế một bộ điều khiển mờ để điều khiển cho hai khớp liên kết của Robot.

Ở đây ta xây dựng bộ điều khiển mờ trên Fuzzy Toolbox theo hệ qui tắc Mandani. Bộ điều khiển có hai tín hiệu vào là sai số E và sai số vi phân DE, có một tín hiệu ra là vi phân điện áp điều khiển DU.

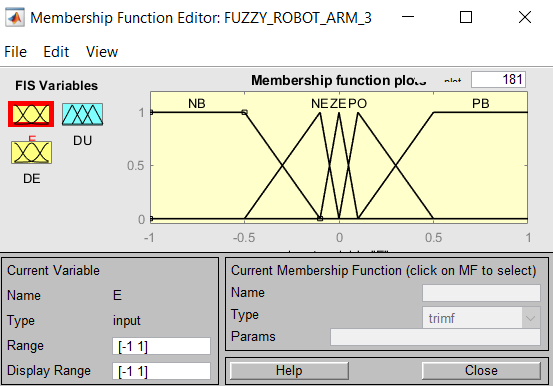
Phương pháp suy diễn MAX-MIN.

Phương pháp giải mờ trung bình trọng số.

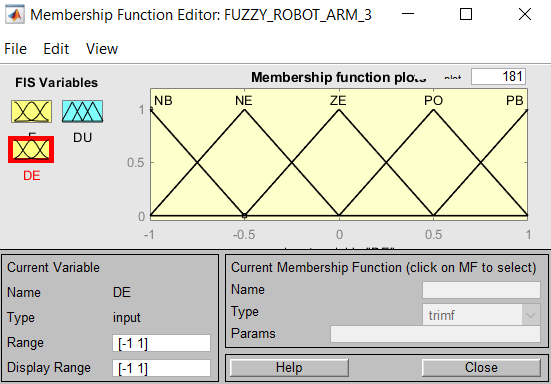


Hình 2 Bộ điều khiển mờ trên Fuzzy Toolbox

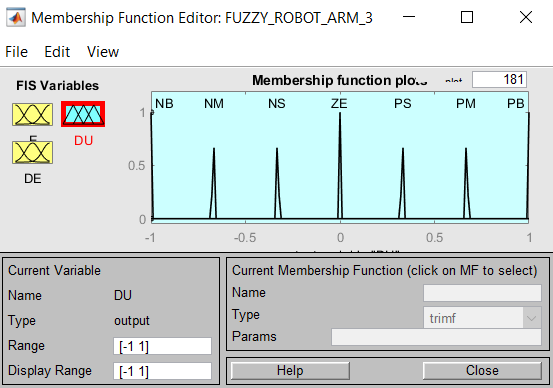
Hệ có tính phi tuyến cao, nên ta chọn số tập mờ cho mỗi biến vào là 5 tập mờ và biến ngõ ra là 7 tập mờ.



Hình 3 Hàm liên thuộc ngõ vào sai số E



Hình 4 Hàm liên thuộc ngõ vào sai số vi phân DE



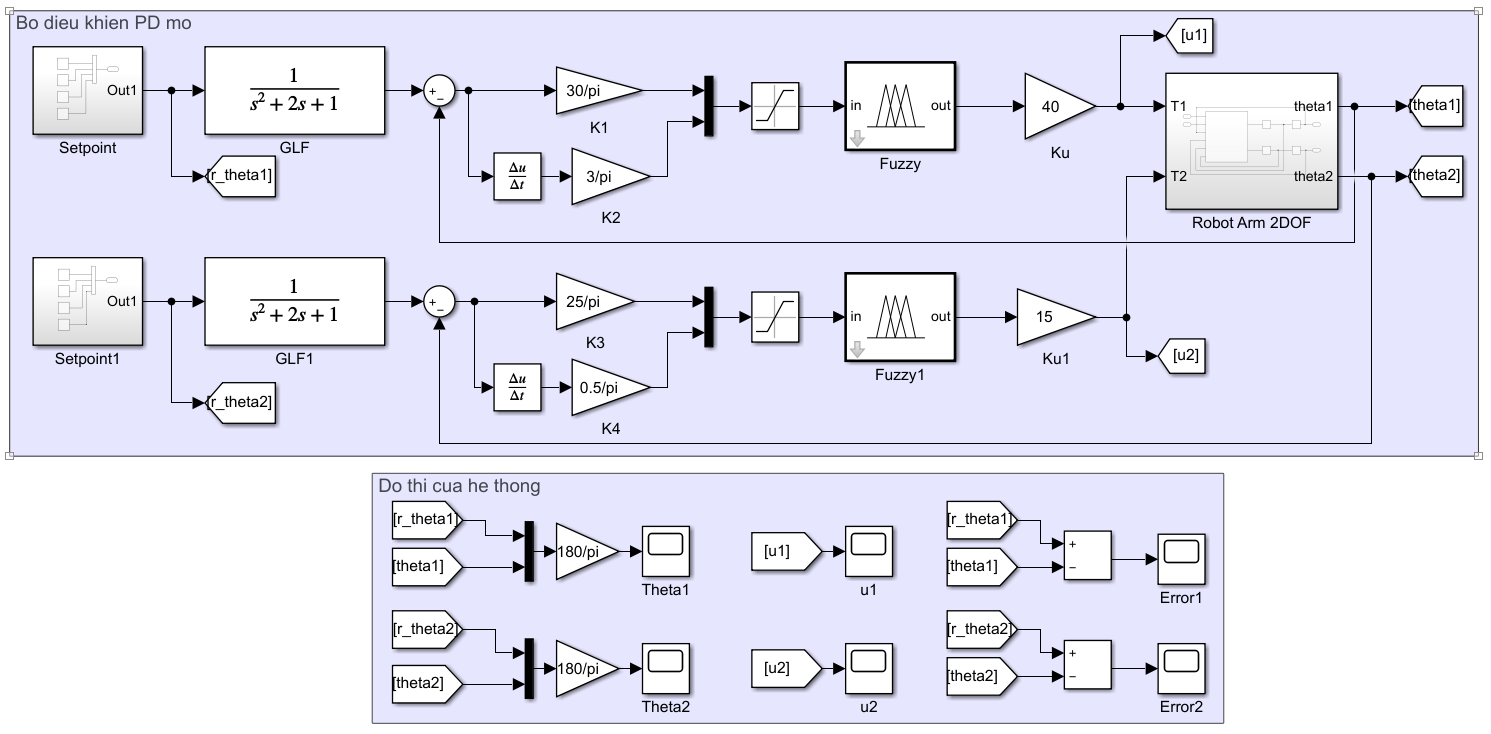
Hình 5 Hàm liên thuộc ngõ ra U

Hệ qui tắc mờ điều khiển được đưa ra dựa vào kinh nghiệm:

*Bảng 1 Các qui tắc mờ cho bộ điều khiển cách tay máy 1 bậc tự do*

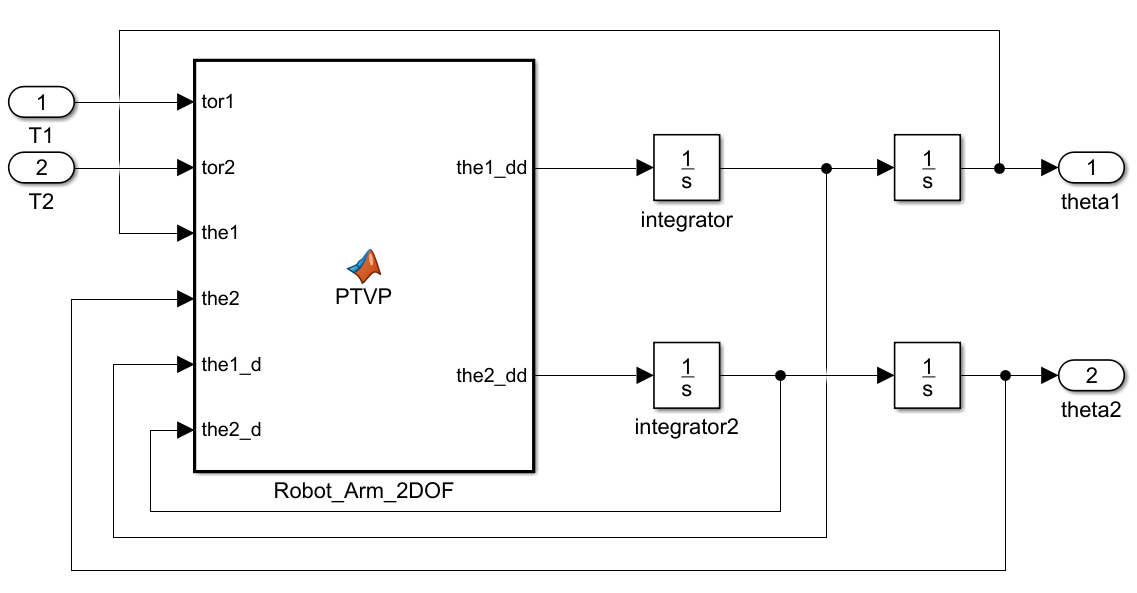
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| DU | | E | | | | |
| NB | NE | ZE | PO | PB |
| DE | NB | NB | NB | NM | NS | ZE |
| NE | NB | NM | NS | ZE | PS |
| ZE | NM | NS | ZE | PS | PM |
| PO | NS | ZE | PS | PM | PB |
| PB | ZE | PS | PM | PB | PB |

## Xây dựng file mô phỏng trên Matlab Simulink



Hình 6 Bộ điều kiển PD mờ cho hệ cánh tay máy 2 bậc trên Matlab Simulink

* Mô hình đối tượng điều khiển cánh tay máy 2 bậc tự do (Robot Arm 2DOF):



Hình 7 Đối tượng điều khiển cánh tay máy 2 bậc tự do

Chương trình của khối Robot\_Arm\_2DOF:

function [the1\_dd,the2\_dd] = PTVP(tor1,tor2,the1,the2,the1\_d,the2\_d)

J1=0.05;

J2=0.05;

m1=1;

m2=1;

m3=0;%Khoi luong cua hai khop

l1=1;

l2=1;

lc1=0.4;

lc2=0.4;

g=9.81;

H11=J1+J2+m1\*(lc1^2)+m2\*((l1^2)+(lc2^2)+2\*l1\*lc2\*cos(the2))+m3\*(l1\*l1+l2\*l2+2\*l1\*l2\*cos(the2));

H22=J2+m2\*lc2\*lc2+m3\*l2\*l2;

H12=J2+m2\*(lc2\*lc2+l1\*lc2\*cos(the2))+m3\*(l2\*l2+2\*l1\*l2\*cos(the2));

H21=H12;

H=[H11 H12;

H21 H22];

h=m2\*l1\*lc2\*sin(the2);

V=[-h\*the2\_d -h\*the1\_d - h\*the2\_d;

-h\*the1\_d 0];

g1=m1\*lc1\*g\*cos(the1)+m2\*g\*(lc2\*cos(the1+the2)+l1\*cos(the1));

g2=m2\*lc2\*g\*cos(the1+the2);

G=[g1;

g2];

tor=[tor1;

tor2];

the\_dd=pinv(H)\*(tor-G-V\*[the1\_d;the2\_d]);

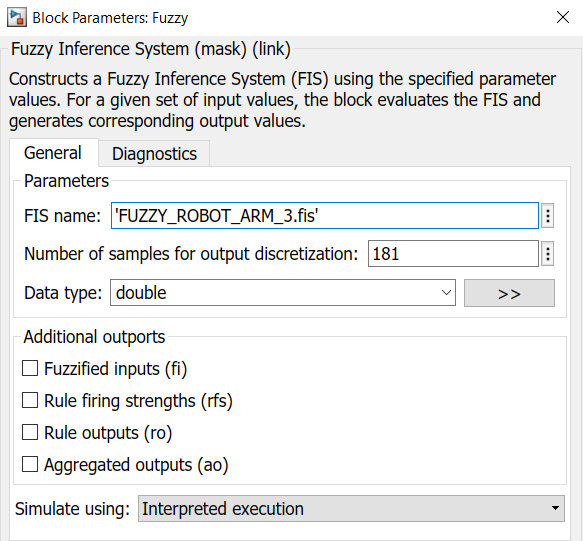
%-diag([1;1])\*[the1\_d;the2\_d]

the1\_dd=the\_dd(1);

the2\_dd=the\_dd(2);

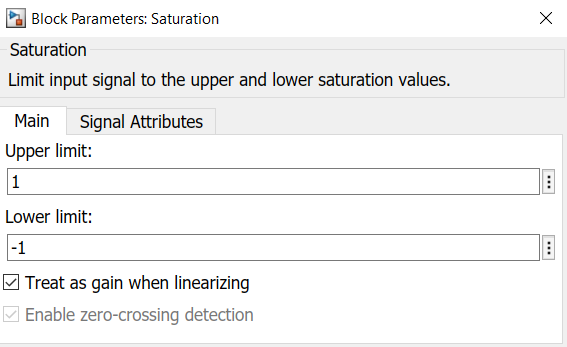
end

* Khối Fuzzy Logic Control:



Hình 8 Khối Fuzzy Logic Control

* Khối Saturation:



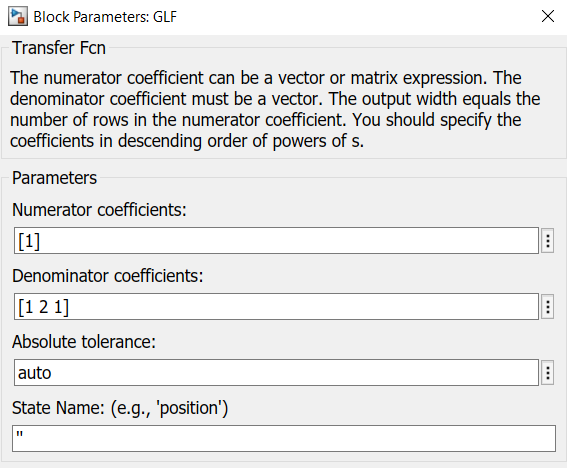
Hình 9 Khối Saturation

* Khối Transfer Fcn:

Thiết kế bộ lọc thông thấp: ta có các bộ lọc sau

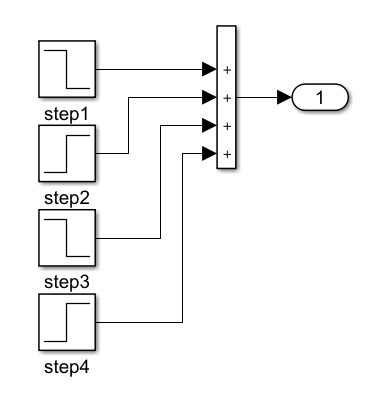


Qua quá trình mô phỏng với các bộ lọc trên, ta thấy bộ lọc  cho ra đáp ứng tốt hơn so với các bộ lọc khác.



Hình 10 Khối Transfer Fcn

* Khối Setpoint giá trị đặt:



Hình 11 Khối Setpoint

* Khối Gain: K1, K2, Ku là thông số dùng để điều khiển Torque1. K3, K4, Ku1 là thông số dùng để điều khiển Torque2. Bằng phương pháp chỉnh định thử sai ta có thể chọn các thông số chuẩn hóa như sau:





## Thu thập và trình bày kết quả

Phiên bản Matlab 2019a

Thời gian lấy mẫu là: 0.01s

Thời gian mô phỏng là: 40s

**Kết quả thu được tại khớp thứ nhất của cánh tay máy 2 bậc tự do**

* Đồ thị đáp ứng ngõ ra của hệ thống tại khớp liên kết thứ nhất của Robot:



Hình 12 Đáp ứng ngõ ra tại khớp thứ nhất của Robot 2 DoF

* Đồ thị tín hiệu điều khiển của hệ thống tại khớp liên kết thứ nhất của Robot:



Hình 13 Tín hiệu điều khiển tại khớp thứ nhất của Robot 2 DoF

* Đồ thị giá trị sai số giữa ngõ ra với giá trị đặt tại khớp liên kết thứ nhất của Robot:



Hình 14 Giá trị sai số tại khớp thứ nhất của Robot 2 DoF

**Kết quả thu được tại khớp thứ hai của cánh tay máy 2 bậc tự do**

* Đồ thị đáp ứng ngõ ra của hệ thống tại khớp liên kết thứ hai của Robot:



Hình 15 Đáp ứng ngõ ra tại khớp thứ hai của Robot 2 DoF

* Đồ thị tín hiệu điều khiển của hệ thống tại khớp liên kết thứ hai của Robot:



Hình 16 Tín hiệu điều khiển tại khớp thứ hai của Robot 2 DoF

* Đồ thị giá trị sai số giữa ngõ ra với giá trị đặt tại khớp liên kết thứ hai của Robot:



Hình 17 Giá trị sai số tại khớp thứ hai của Robot 2 DoF

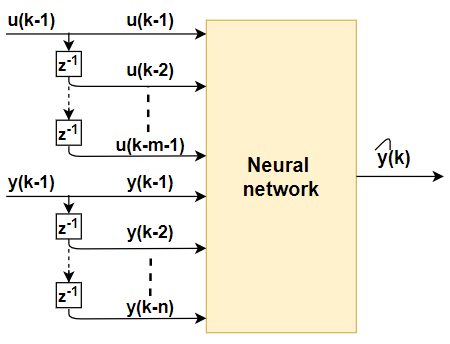
## Phân tích kết quả mô phỏng

* Từ các đồ thị trên ta thấy đáp ứng ngõ ra góc quay theta1 và theta2 tại 2 khớp liên kết của Robot bám sát với giá trị đặt trước, hầu như không có vọt lố, thời gian lên và thời gian xác lập khoảng 5s tuy không nhanh nhưng vẫn chấp nhận được, điều này là do ta chọn bộ lọc thông thấp cho bộ điều khiển vẫn chưa tối ưu nhất. Vì vậy có thể xem bộ điều khiển trượt đã đạt yêu cầu. Tuy nhiên từ đồ thị thu được tín hiệu điều khiển ta thấy hệ thống vẫn còn dao động nhưng không đáng kể. Từ đồ thị giá trị sai số của hệ thống ta thấy giá trị sai số vẫn còn dao động quanh giá trị 0, điều này có thể do bộ điều khiển PD mờ chưa phải là bộ điều khiển tối ưu nhất đối với hệ cánh tay máy 2 bậc tự do, mà còn có các bộ điều khiển khác chẳng hạn như PID mờ sẽ tối ưu hơn cho hệ thống này.

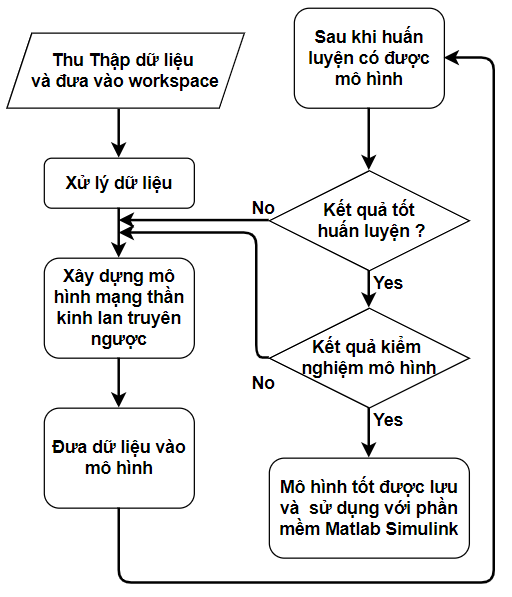
# THIẾT KẾ MẠNG THẦN KINH XẤP XỈ BỘ ĐIỀU KHIỂN MỜ

## Cơ sở lý thuyết về mạng thần kinh nhân tạo

Bộ điều khiển thần kinh được tạo ra là dựa vào cơ chế truyền nhận và xử lý tín hiệu của bộ não con người. Việc xây dựng bộ điều khiển thần kinh chính là tạo một mạng thần kinh nhân tạo bắt chước quá trình làm việc của bộ não người. Bản chất mạng thần kinh nhân tạo là mạng tính toán phân bố song song. Trái với các mô hình tính toán thông thường, hầu hết các mạng thần kinh phải được huấn luyện trước khi sử dụng. Ở đây nhóm em sẽ trình bày các bước tạo một mạng thần kinh để điều khiển vị trí góc quay của cánh tay máy 2 bậc tự do xấp xỉ với bộ điều khiển mờ.

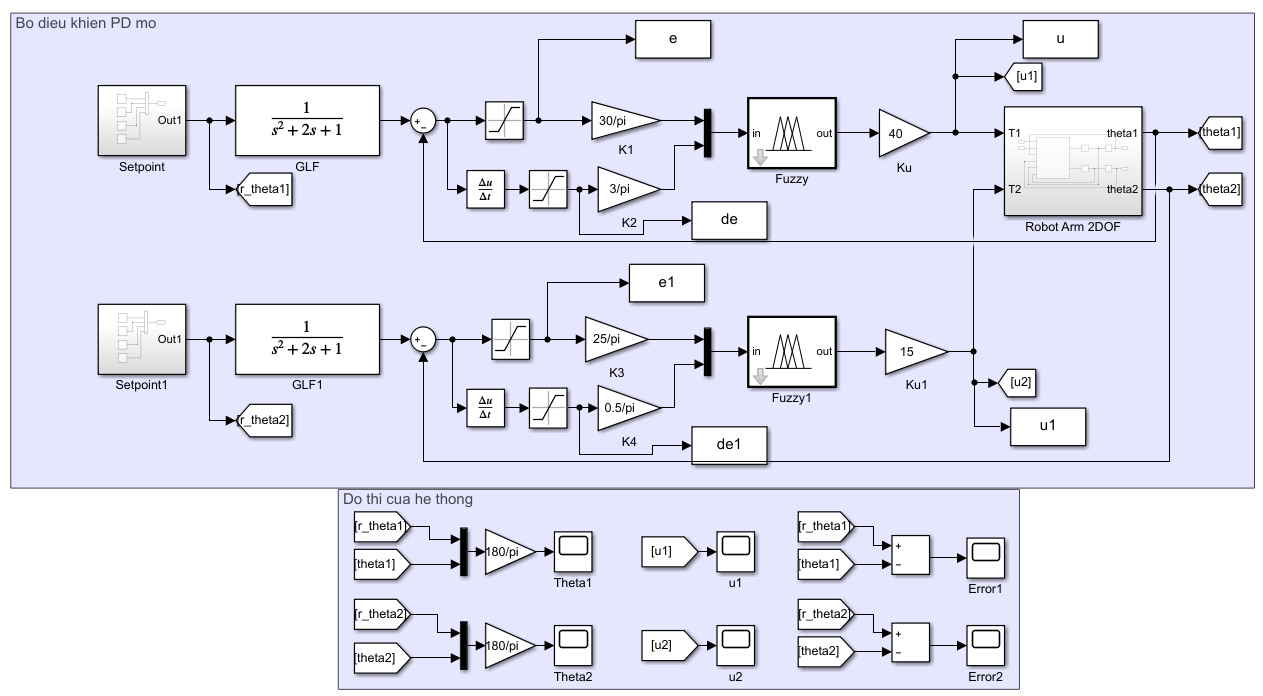


Hình 18 Mô hình mạng thần kinh kiểu dự báo



Hình 19 Lưu đồ giải thuật cho quá trình xây dựng một mạng thần kinh nhân tạo

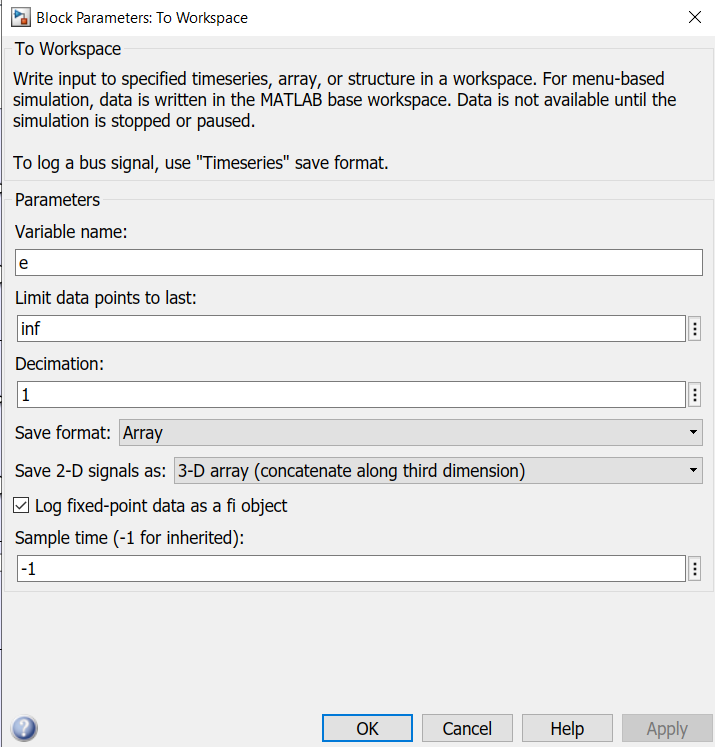
## Thu thập dữ liệu



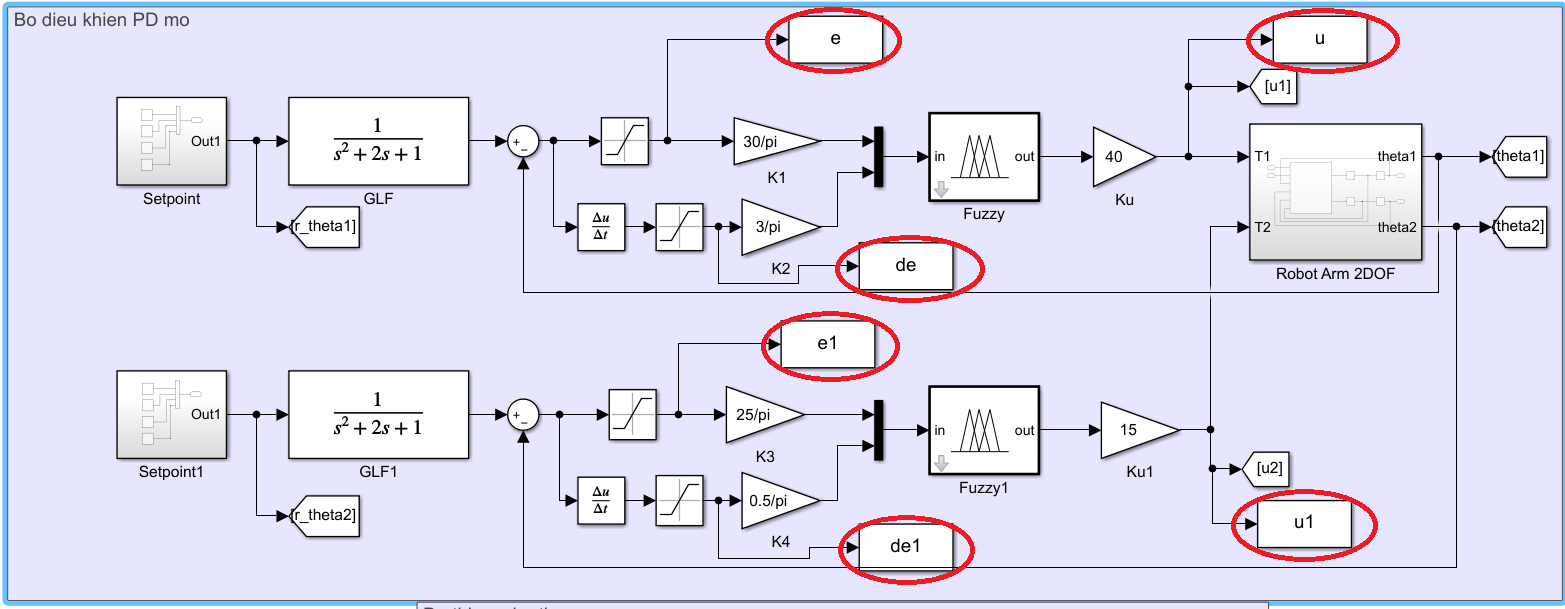
Hình 20 Cấu trúc hệ thu thập dữ liệu

* Dữ liệu cần thu thập để huấn luyện mạng neural được lấy tại các điểm đặt khối “to workspace”.

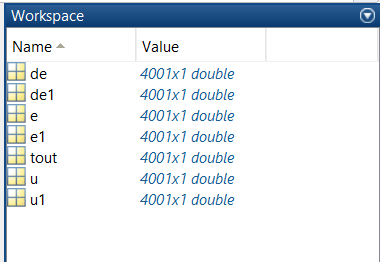
Trong khối “to workspace”:



Hình 21 Thông số khối workspace



Hình 22 Các điểm được thu thập dữ liệu



Hình 23 Dữ liệu được thu thập từ quá trình mô phỏng

Sau khi việc thu thập dữ liệu đã thực hiện xong, ta tiến hành xây dựng mạng thần kinh nhân tạo với cấu trúc:

* Mạng có 3 ngõ vào và 1 ngõ ra ứng với mỗi khớp.
* Khớp 1: Có 3 lớp vào, 20 lớp ẩn thứ 1 với các tế bào thần kinh có hàm kích hoạt dạng S lưỡng cực(tansig), 5 lớp ẩn thứ 2 với các tế bào thần kinh có hàm kích hoạt dạng S đơn cực (logsig), 1 ngõ ra với tế bào thần kinh có hàm kích hoạt tuyến tính (purelin).
* Khớp 2: Có 3 lớp vào, 20 lớp ẩn thứ 1 với các tế bào thần kinh có hàm kích hoạt dạng S lưỡng cực(tansig), 5 lớp ẩn thứ 2 với các tế bào thần kinh có hàm kích hoạt dạng S đơn cực (logsig), 1 ngõ ra với tế bào thần kinh có hàm kích hoạt tuyến tính (purelin).
* Tốc độ học mỗi khớp đều có hệ số là 0.05, và cho học 200000 lần, và sai số dừng là 10e-7.
* Lớp ra gồm 2 tế bào thần kinh có hàm kích hoạt tuyến tính (purelin).

## Thiết kế bộ điều khiển

Chương trình tạo mạng neural:

%% Chuong trinh mang neural khop 1

K = length(u);

X = [e(2:K)'; de(2:K)'; u(1:K-1)']; %vecto dau vao

D = [u(2:K)']; %vecto ket qua

N = 20; %so lop an

mynet = newff(minmax(X),[N 5 1] ,{'tansig' 'logsig' 'purelin'},'trainrp'); %ham tong hop va ham tac dong

mynet.trainparam.show = 50;

mynet.trainparam.epochs=200000; %so lan chay

mynet.trainparam.lr=0.05;

mynet.trainparam.goal=1e-7;

Mynet = train(mynet,X,D); %hinh thanh mang neuron

gensim(Mynet);

a=sim(Mynet,X);

%%

plot(a)

%% Chuong trinh mang neural khop 2

K1 = length(u1);

X1 = [e1(2:K1)'; de1(2:K1)'; u1(1:K1-1)']; %vecto dau vao

D1 = [u1(2:K1)']; %vecto ket qua

N1 = 20; %so lop an

mynet1 = newff(minmax(X1),[N1 5 1] ,{'tansig' 'logsig' 'purelin'},'trainrp'); %ham tong hop va ham tac dong

mynet1.trainparam.show = 50;

mynet1.trainparam.epochs=200000; %so lan chay

mynet1.trainparam.lr=0.05; %he so hoc

mynet1.trainparam.goal=1e-7;

Mynet1 = train(mynet1,X1,D1); %hinh thanh mang neuron

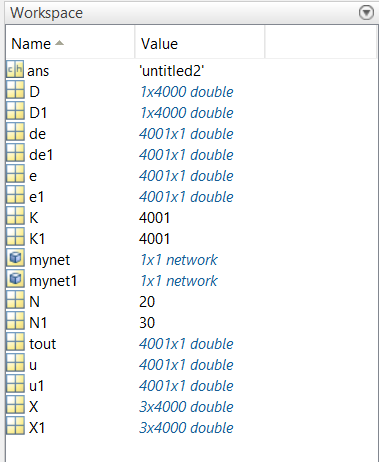
gensim(Mynet1);

%%

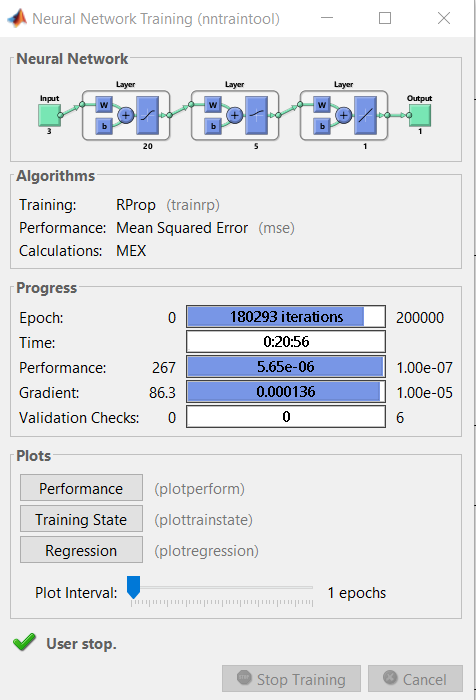
a1=sim(Mynet1,X1);

plot(a1)

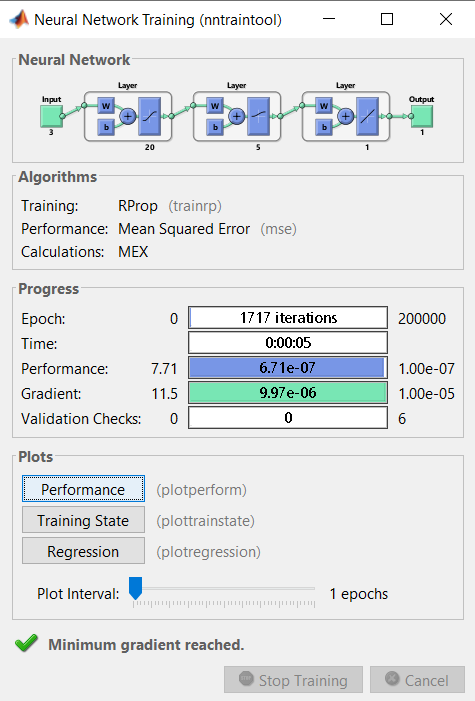
Kết quả chạy chương trình:



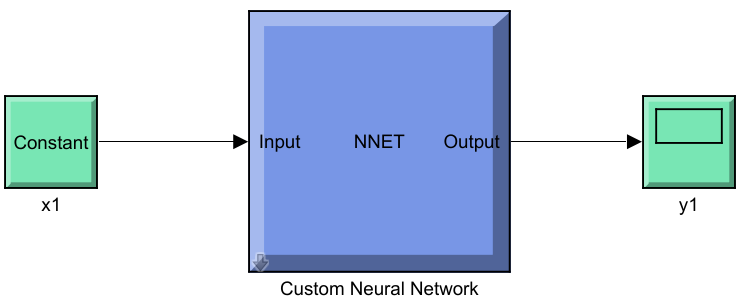
Hình 24 Các biến ở workspace



Hình 25 Bảng huấn luyện mạng neural ở khớp 1



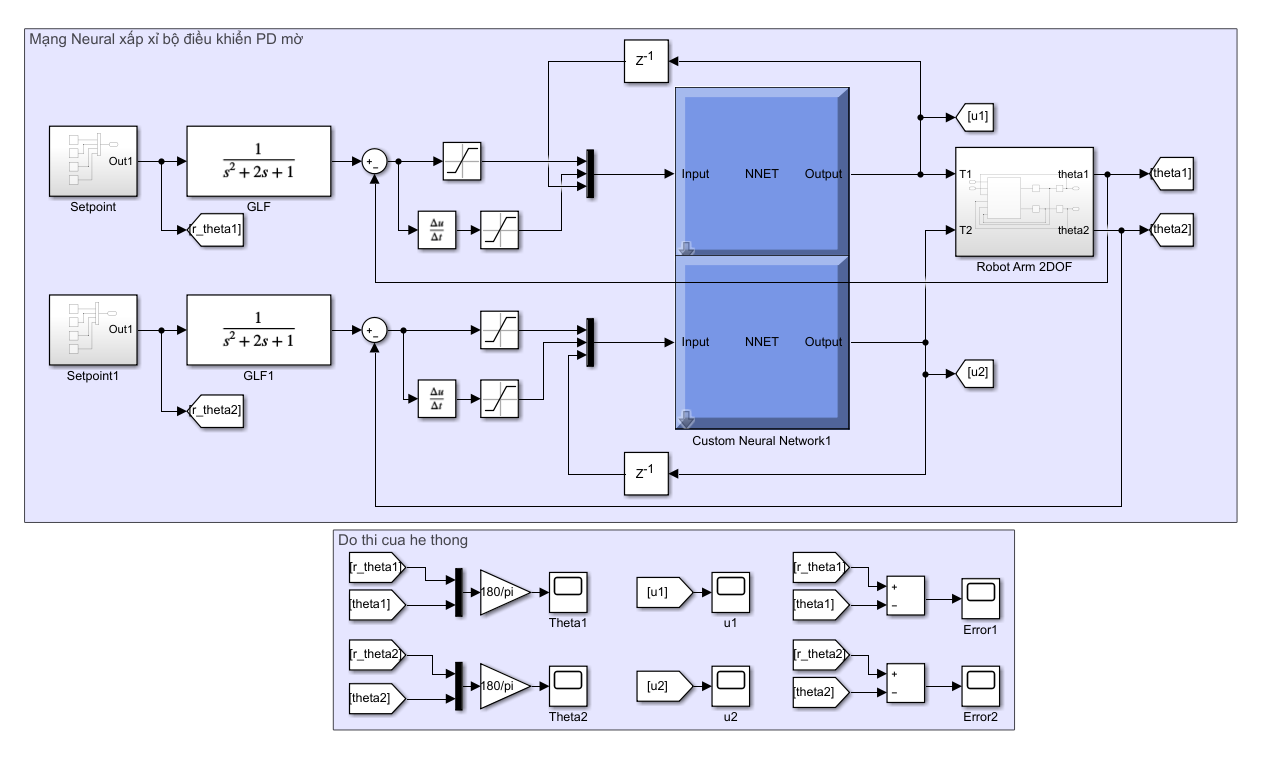
Hình 26 Bảng huấn luyện mạng neural ở khớp 2



Hình 27 Kết quả huấn luyện ở simulink

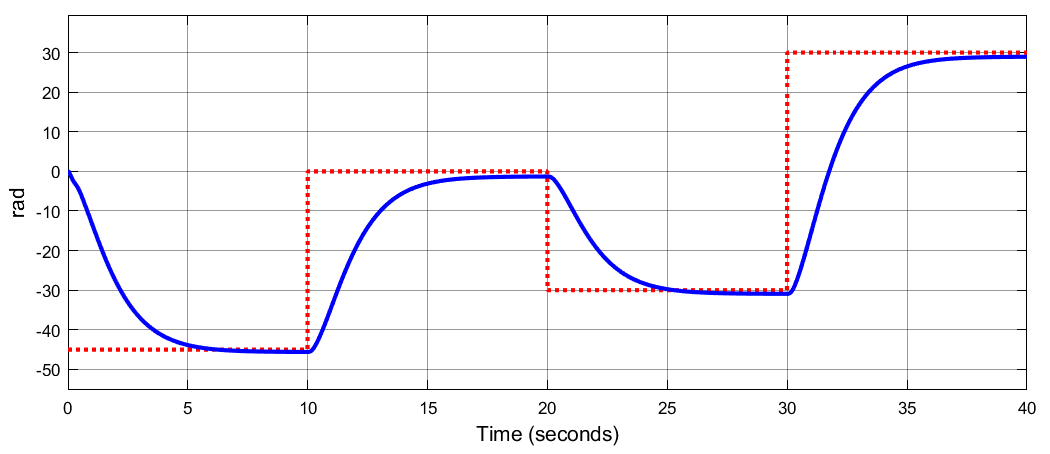
Sau khi quá trình huấn luyện mạng kết thúc, một giao diện thể hiện các thông số của mạng thần kinh (hình 9) và một khối nơ ron được tạo ra (hình 10), có khả năng điều khiển vị trí góc quay của robot 2 bậc tự do xấp xỉ khối mờ (fuzzy logic controller). Khi đó ta tiến hành lấy khối nơ ron được tạo và mô phỏng so sánh đáp ứng ngõ ra của hai bộ điều khiển như hình 11.

Ta có bộ điều khiển bằng mạng neural:

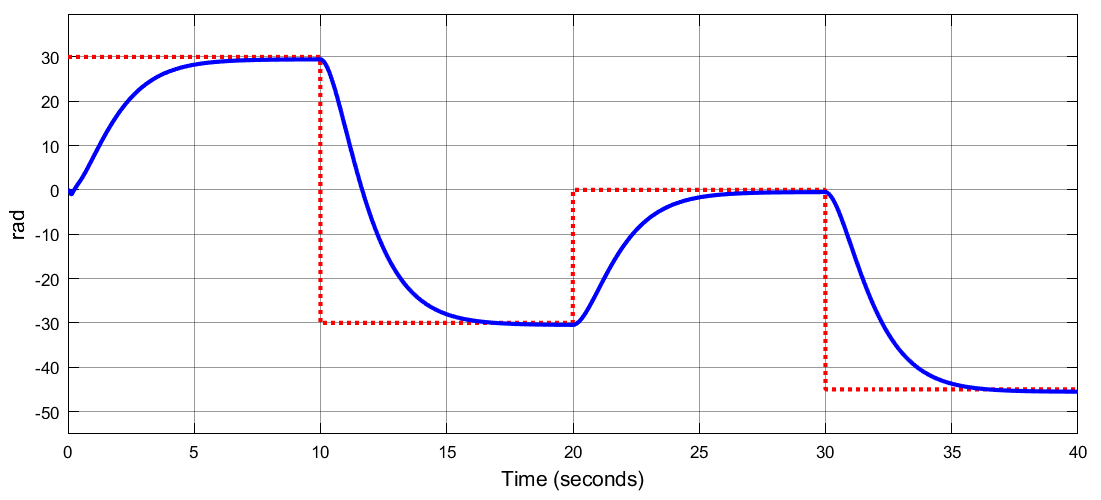


Hình 28 Sơ đồ kết nối với bộ điều khiển mạng neural

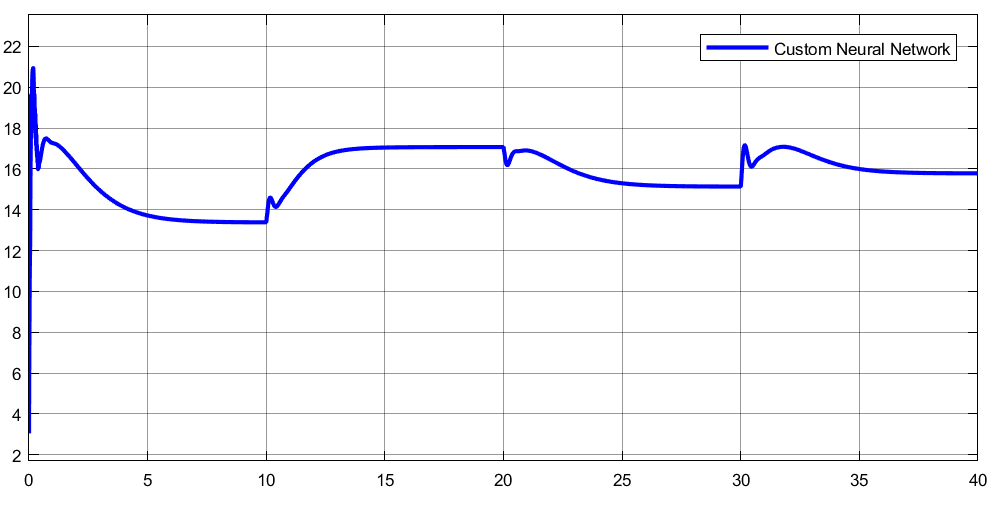
## Kết quả mô phỏng



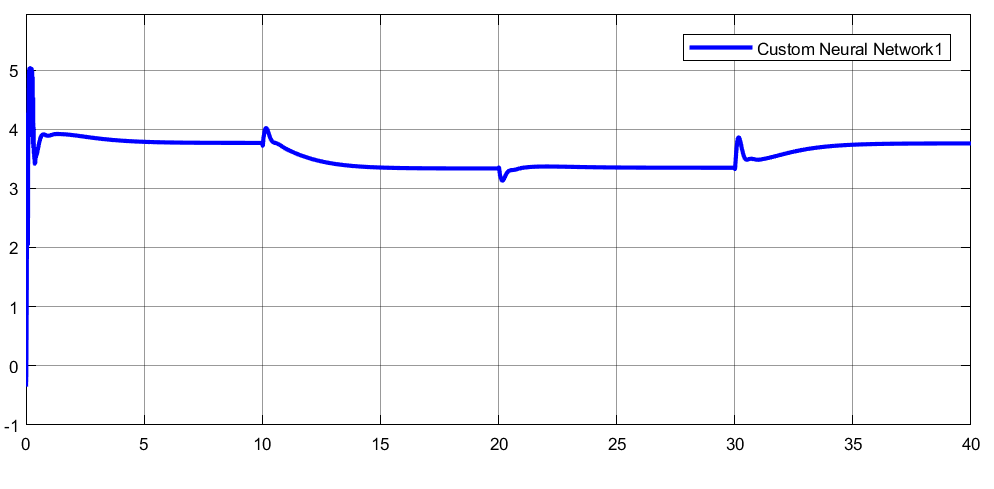
Hình 29 Đáp ứng ngõ ra của khớp 1



Hình 30 Đáp ứng ngõ ra của khớp 2



Hình 31 Tín hiệu điều khiển của khớp 1



Hình 32 Tín hiệu điều khiển của khớp 2

**Nhận xét:**

Dựa vào các hình trên ta có thể thấy được đáp ứng ngõ ra bộ điều khiển thần kinh chưa được tốt so với bộ điều khiển fuzzy. Bới vì phương pháp huấn luyện ở đây nhóm em dùng phương pháp thử sai nên tín hiệu điều khiển robot 2 DOF không tốt dẫn đến hệ chỉ có thể tiến về xấp xỉ giá trị đặt mà chưa được bám sát tốt.

# SỬ DỤNG GIẢI THUẬT GA ĐỂ XÁC ĐỊNH CÁC THÔNG SỐ BỘ ĐIỀU KHIỂN PID TỐI ƯU

Giải thuật di truyền (Genetic Algorithm - GA) là giải thuật tìm kiếm lời giải tối ưu trên nguyên tắc phỏng đoán theo quá trình tiến hóa và quy luật di truyền của sinh vật trong tự nhiên. Trong giải thuật di truyền cho bài toán tối ưu, ý tưởng là một tập hợp của các nhiễm sắc thể trong các cá thể sẽ được cho tiến triển theo hướng chọn lọc những giải pháp tốt dần hơn.

## Lưu đồ giải thuật di truyền



*Hình 33 Lưu đồ giải thuật di truyền*

* ***Mã hóa:*** trong quần thể mỗi cá thể sẽ được mã hóa thành 1 chuỗi nhiễm sắc thể (NST). Tùy theo phương pháp mà chuỗi NST có thể là chuỗi số nhị phân, chuỗi số thập phân, chuỗi số tự nhiên hay chuỗi số thực.

Có 3 cách thường dùng để mã hóa bài toán tìm cực trị hàm là mã hóa nhị phân, mã hóa thập phân và mã hóa số thực. Nhưng trong bài toán này nhóm đã sử dụng mã hóa thập phân.

*Code chương trình Mã hóa thập phân:*

function pop=Encode\_Decimal(par,sig,dec)

%Biến vào

%par: mmảng hai chiều lưu[Nxd] giá trị quầnn thể của mã hóa

%sig:vector hàng[1xd] lưu chữ số có nghĩa tương ứng với mỗi thông số

%dec:vector hàng [1xd] Lưu vị trí dấu chấm thập phân

if(nargin<3)

error('Thieu doi so.Cú pháp: pop=encode\_decimal(par,sig,dec)');

end

if(size(sig)~=size(dec))

error('??i S? không phù h?p v?i nhau');

end

[N,d]=size(par);

for pop\_index=1:N %quét t? ??u ??n cu?i qu?n th?

gene\_index=1;

for par\_index=1:d

%gen ??u tiên mã hóa d?u c?a thông s?

if par(pop\_index,par\_index)<0

pop(pop\_index,gene\_index)=0;

else

pop(pop\_index,gene\_index)=9;

end

gene\_index=gene\_index+1;

%c?c gien ti?p theo là ch? s? có ngh?a

temp(par\_index)=abs(par(pop\_index,par\_index))/10^dec(par\_index);

for count=1:sig(par\_index),

temp(par\_index)=temp(par\_index)\*10;

pop(pop\_index,gene\_index)=temp(par\_index)-...

rem(temp(par\_index),1);

temp(par\_index)=temp(par\_index)-...

pop(pop\_index,gene\_index);

gene\_index=gene\_index+1;

end

end

end

end

* ***Chọn lọc:*** chọn ra những cá thể có độ thích nghi cao. Cá thể nào có độ thích nghi càng cao thì càng có nhiều cơ hội để tồn tại và bắt cặp với 1 cá thể khác để sinh ra thế hệ con có xu hướng thích nghi tốt hơn so với thế hệ bố mẹ. Vì vậy phép toán chọn lọc có vai trò trọng đối với chât lượng của GA.

Các phương pháp chọn lọc.

* Chọn lọc tỉ lệ.
* Chọn lọc đấu vòng.
* Chọn lọc cắt.
* Chọn lọc sắp hạng tuyến tính.
* Chọn lọc sắp hạng lũy thừa.
* Cường độ chọn lọc.



Trong đó, 𝑀 và 𝑀∗ là độ thích nghi trung bình của quần thể trước và sau chọn lọc,  là phương sai của độ thích nghi trước chọn lọc.

Ở bài này nhóm đã sử phương pháp chọn lọc sắp hạng tuyến tính để chọn lọc quần thể.

*Chương trình matlab chọn lọc sắp hạng tuyến tính*

function parent=select\_Linear\_Ranking(pop,fitness,eta,elitism,bestchorm)

if(nargin<5)

error('Thiếu đối Số');

end

N=length(fitness);%Kích thước quần thể

[fitness,order]=sort(fitness); %sắp xếp tăng đân theo độ thích nghi

for k=1:N

p(k)=(eta+(2-eta)\*(k-1)/(N-1))/N;

end

s=zeros(1,N+1);

for k=1:N

s(k+1)=s(k)+p(k);

end

for k=1:N

if (elitism==1)&(order(k)==bestchorm)

%eitism=1 và NST đang xét là NST tốt nhấtt thì NST đó được chon

parent(order(k),:)=pop(order(k),:);

else

%nếu eitism==0 và NST đang xét không phải là NST tốt nhất thì NST

%dó đc chon xác xuất tỉ lệ hạng với nó

r=rand\*s(N+1);

index=find(s<r); j=index(end);

parent(order(k),:)=pop(order(j),:);

end

end

end

* ***Lai ghép:*** sau khi được chọn lọc thì các NST sẽ được bắt cặp ngẫu nhiên 2 NST để lai ghép với nhau tạo ra 2 NST con, mỗi NST con thừa hưởng 1 phần gien của bố, 1 phần gien của mẹ với triển vọng tạo ra NST con tốt hơn so với bố mẹ.

Các phương pháp lai ghép:

* Lai ghép một điểm.
* Lai ghép nhiều điểm.
* Lai ghép đều.

Nhóm em chọn phương pháp lai ghép nhiều điêm là 2 điềm.

*Chương trình matlab lai ghép 2 điểm*

function child=Cross\_Twopoint(parent,Pc,elitism,bestchorm)

if(nargin<4)

error('Thiếu đoi Số');

end

[N,L]=size(parent);

for p1=1:N

%nêu cờ elitimi=1 và p1 là cá thể siêu việt thì không lai ghép

if(elitism==1)&(p1==bestchorm)

child(p1,:)=parent(p1,:);

else

if Pc>rand %Lai ghép xảy ra với xác xuất Pc

p2=p1; %chọn ngẫu nhiên cá thể cha mẹ 2

while p2==p1 %khác cá thể cha mẹ 1

p2=rand\*N;

p2=p2-rem(p2,1)+1;

end

k1=rand\*(L-1); %chọn ngẫu nhiên điểm lai ghép 1

k1=k1-rem(k1,1)+1;

k2=k1;

while k2==k1

k2=rand\*(L-1); %chọn ngẫu nhiên điểm lai ghép 2, khác điểm lai ghép 1

k2=k2-rem(k2,1)+1;

end

%nếu k2<k1 thì chuyển đỏii k1 và k2 để đảm bảo rằng k1<k2

if k1>k2, t=k2;k2=k1;k1=t; end;

%cá thể con thừa hưởng gien cha mẹ

child(p1,1:k1)=parent(p1,1:k1);

child(p1,k1+1:k2)=parent(p2,k1+1:k2);

child(p1,k2+1:L)=parent(p1,k2+1:L);

else

child(p1,:)=parent(p1,:);

end

end

end

end

* ***Đột biến:*** phép toán đột biến thay đổi ngẫu nhiên 1 hoặc nhiều gien của 1 cá thể để làm tăng sự đa dạng về cấu trúc trong quần thể. Vai trò của đột biến là khôi phục chất liệu di truyền bị mất hoặc chưa khai phá để ngăn hiện tượng GA hội tụ sớm.

Các phương pháp đột biến.

* Đột biến một điểm.
* Đột biến đa điểm.

Trong bài này nhóm áp dụng phương pháp đột biến đa điểm là đột biến đều.

function newpop=Mutata\_Uniform(pop,Pm,elitism,bestchorm)

if(nargin<4)

error('Thieu doi so');

end

[N,L]=size(pop);

newpop=pop;

for pop\_index=1:N

if(elitism==0)||(elitism==1&&pop\_index~=bestchorm)

for gene\_index=1:L

if Pm>rand

rand\_gene=rand\*10;

while(pop(pop\_index,gene\_index)==...

rand\_gene-rem(rand\_gene,1)||rand\_gene==10)

rand\_gene=rand\*10;

end

newpop(pop\_index,gene\_index)=rand\_gene-rem(rand\_gene,1);

end

end

end

end

end

* *Giải mã*: giải mã các đoạn NST sau khi đã lai ghép.

*Chương trình matlab giải mã.*

function par=Decode\_Decimal(pop,sig,dec)

if(nargin<3)

error('Thi?u ??i s?.Cú pháp: pop=encode\_decimal(par,sig,dec)');

end

if(size(sig)~=size(dec))

error('??i S? không phù h?p v?i nhau');

end

[N,L]=size(pop);

d=length(sig);

if L~=sum(sig)+d

error('chi?u dài chu?i NST không phù h?p v?i sig');

end

par=zeros(N,d);

for pop\_index=1:N

start\_gene=1;

for par\_index=1:d

for count=1:sig(par\_index)

gene\_index=start\_gene+count;

weight=dec(par\_index)-count;

par(pop\_index,par\_index)=par(pop\_index,par\_index)+...

(pop(pop\_index,gene\_index))\*10^weight;

end

if pop(pop\_index,start\_gene)<5

par(pop\_index,par\_index)=-par(pop\_index,par\_index);

end

start\_gene=start\_gene+1+sig(par\_index);

end

end

end

* *Đánh giá:* tìm ra những cá thể thích nghi cao và loại bỏ những cá thể kém thích nghi. Sau đó kiểm tra xem có hội tụ hay không, nếu có thì kết thúc quá trình, ngược lại thì quay lại quá trình mã hóa và cứ tiếp tục vòng kín cho đến khi hội tụ.
* *Nhận xét:* Kết quả là qua quá trình tiến hóa được lặp đi lặp lại từ thế hệ này sang thế hệ khác, thế hệ sau có xu hướng thích nghi tốt hơn thế hệ trước, điều đó có nghĩa là lời giải của bài toán tiến dần đến lời giải tối ưu.

## Chương trình giải thuật di truyền bộ thông số PID

Chương trình giải thuật di truyền tìm bộ thông số PID

clc;

close all;

clear all

rand('state',sum(100\*clock));

%% Khai báo các bi?n

%?i?u ki?n d?ng: GA d?ng khi ?ã ch?y ???c 200 th? h? ho?c

%Trong 50 th? h? liên ti?p có hàm thích nghi thay ??i không quá 0.0001

max\_generation=200;

max\_stall\_generation=50;

eplilon=0.0001;

N=20; %Kich thuoc quan the

npar=6; %Co 6 thong so kp, kd, ki

range=[ 0 0 0 0 0 0;...

70 70 70 70 70 70]; %tam khoi dong cua thong so PID

dec=[2 2 2 2 2 2]; %VI tri so thap phan 2

sig=[5 5 5 5 5 5]; %có 5 cho so có ngh?a, vi trí dau cham thap phan là 2

Pc=0.8; %Xác xuat lai ghép là 0.9

Pm=0.1; %Xác xuat dot bien là 0.1

rho=0.02;

elitism=1; %L?u tr? cá th? t?t nh?t trong quá trình ti?n hóa

%% Kh?i ??ng ng?u nhiên th? h? ??u tiên

par=Init(N,npar,range);

terminal=0; %c? báo ch?a k?t thúc thu?t toán

generation=0; %Bi?n ??m s? th? h?

stall\_generation=0; %Bi?n ??m s? th? h? hàm thích nghi không ??i

for pop\_index=1:N

Kp1=par(pop\_index,1);

Ki1=par(pop\_index,2);

Kd1=par(pop\_index,3);

Kp2=par(pop\_index,4);

Ki2=par(pop\_index,5);

Kd2=par(pop\_index,6);

sim('GA\_PID\_2DOF');

if (length(e1)>4000)

J=(e1'\*e1)+(e2'\*e2)+rho\*(u1'\*u1+u2'\*u2);

fitness(pop\_index)=1/(J+eps);

else

J=10^100;

fitness(pop\_index)=1/(J+eps);

end

end

%%

[bestfit0,bestchorm]=max(fitness);

while~terminal % chay dieu kien dung

generation=generation+1;

disp(['generation #' num2str(generation) ' of maximum ' num2str(max\_generation)]);

%Ma hoa thap phan

pop=Encode\_Decimal(par,sig,dec);

%Chon loc xep hang tuyen tinh

parent=select\_Linear\_Ranking(pop,fitness,0.5,elitism,bestchorm);

%LAI GHEP HAI DIEM

child=Cross\_Twopoint(parent,Pc,elitism,bestchorm);

%DOT BIEN DEU

pop=Mutata\_Uniform(child,Pm,elitism,bestchorm);

%Gia ma

par=Decode\_Decimal(pop,sig,dec);

% ?ánh giá l?i ?? thích nghi

for pop\_index=1:N

Kp1=par(pop\_index,1);

Ki1=par(pop\_index,2);

Kd1=par(pop\_index,3);

Kp2=par(pop\_index,4);

Ki2=par(pop\_index,5);

Kd2=par(pop\_index,6);

sim('GA\_PID\_2DOF');

if (length(e1)>4000)

J=(e1'\*e1)+(e2'\*e2)+rho\*(u1'\*u1+u2'\*u2);

fitness(pop\_index)=1/(J+eps);

else

J=10^100;

fitness(pop\_index)=1/(J+eps);

end

end

[bestfit(generation),bestchrom]=max(fitness);

% Kiem tra dieu kien dung

if generation == max\_generation

terminal = 1;

elseif generation>1

if abs(bestfit(generation)-bestfit(generation-1))<eplilon,

stall\_generation=stall\_generation+1;

if stall\_generation == max\_stall\_generation, terminal = 1;end

else

stall\_generation=0;

end

end

end %While

%%

plot(1./bestfit) % ve do thi ham thich nghi

Kp1=par(bestchrom,1) % hien thi cac nhiem sac the sau khi chay xong

Ki1=par(bestchrom,2) % hien thi cac bo thong so PID

Kd1=par(bestchrom,3)

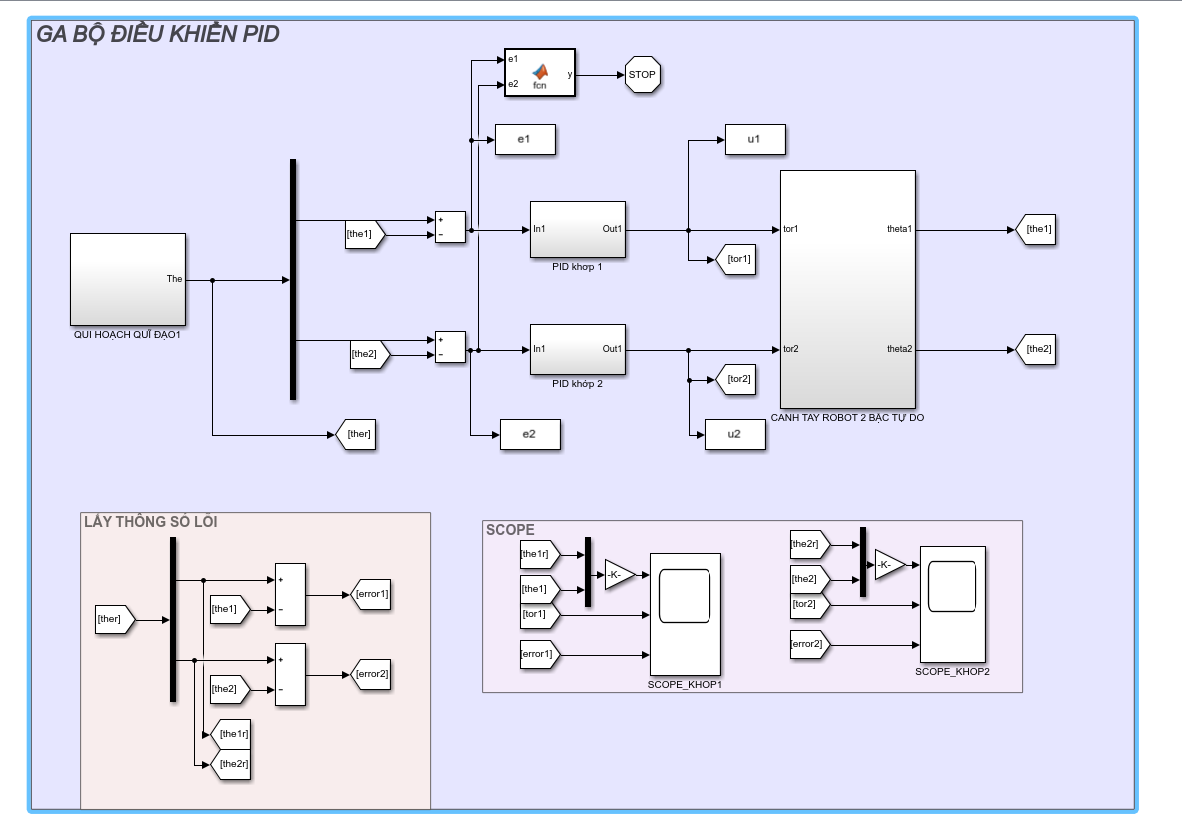
Kp2=par(bestchrom,4)

Ki2=par(bestchrom,5)

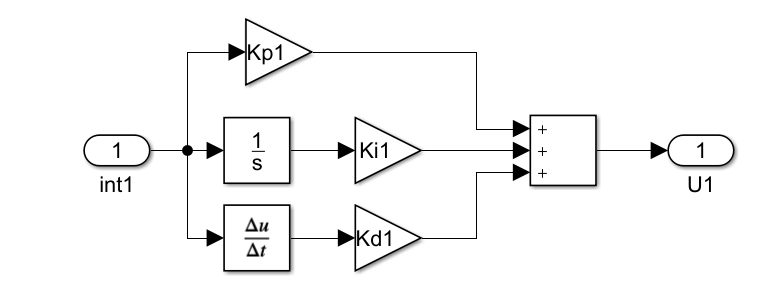
Kd2=par(bestchrom,6)

sim('GA\_PID\_2DOF');

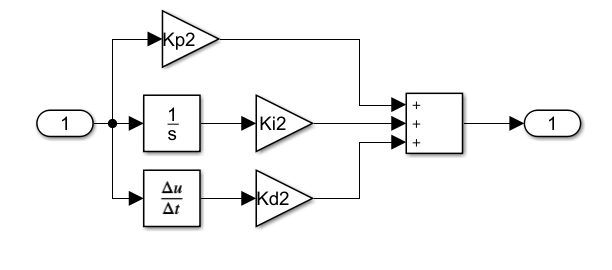
## Mô phỏng Simulink và kết quả



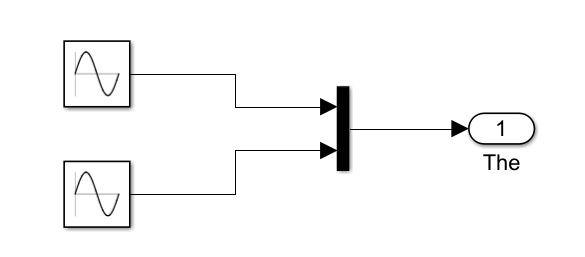
*Hình 34 Simulink mô phỏng bộ điều khiển PID*



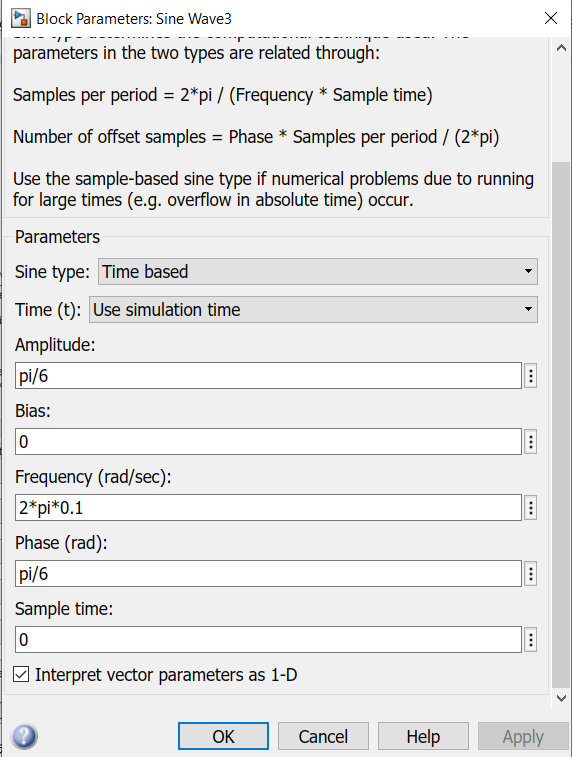
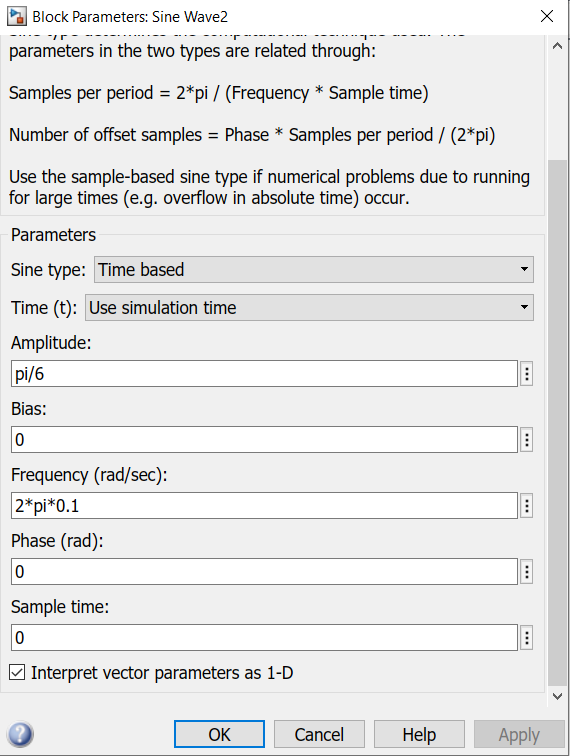
*Hình 35 Bộ điều khiển PID khớp 1*



*Hình 36 Bộ điều khiển PID khớp 2*

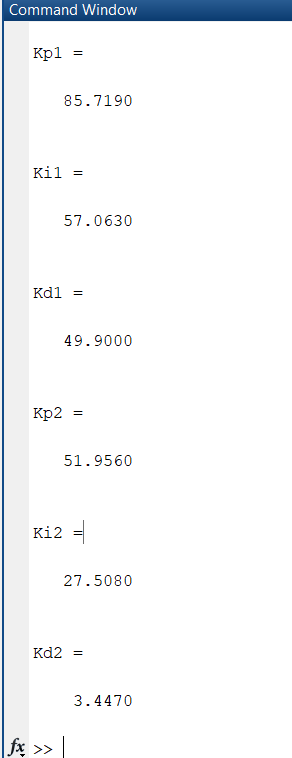


*Hình 37 Khối qui hoạch quĩ đạo hình sin cho cả hai khớp*

**

*Hình 38 Thông số khối sin khớp 1 Hình 39 Thông số khối sin của khớp 2*

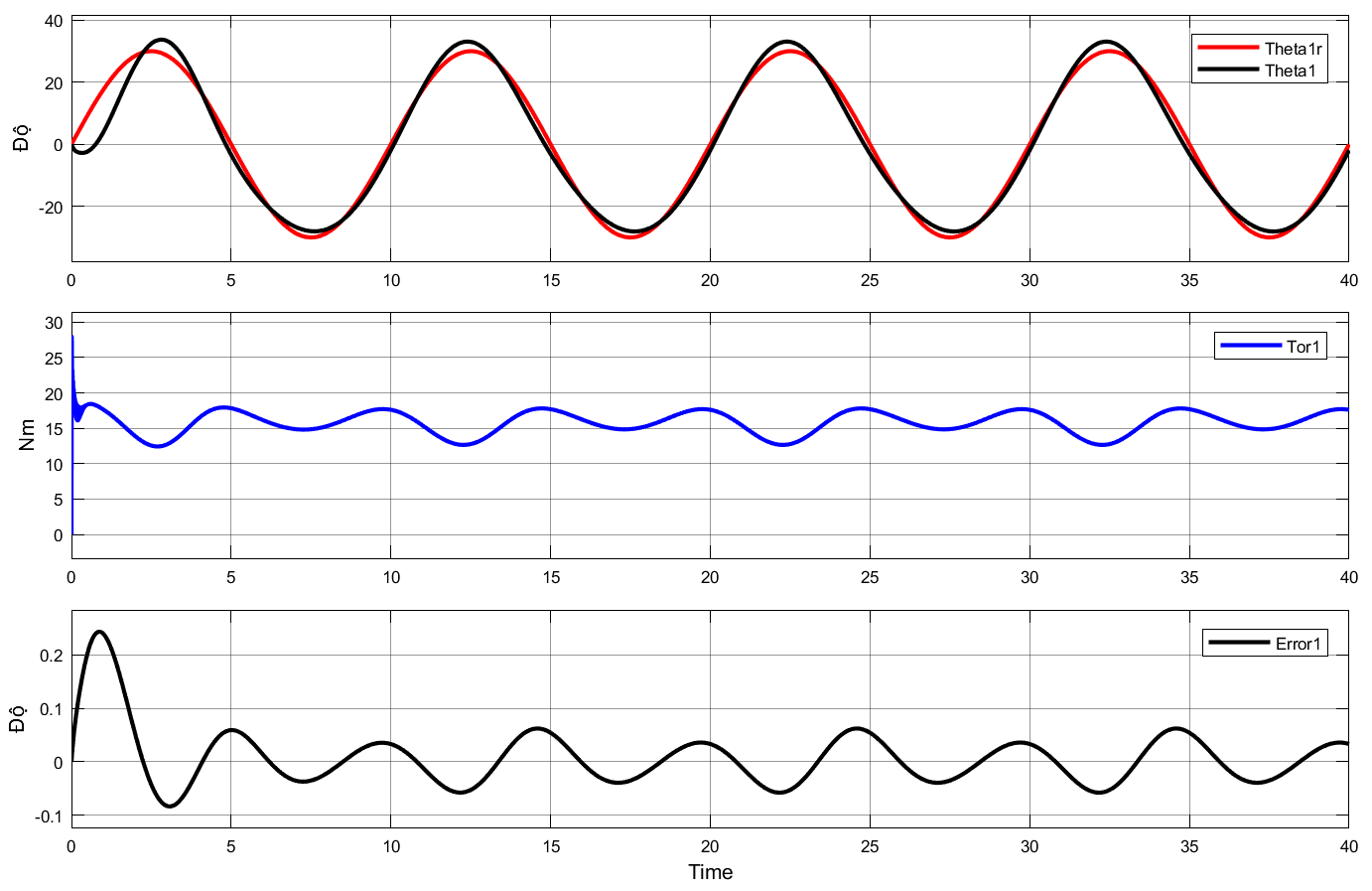
Sau khi chạy chương trình ta thu được thông số bộ điều khiển PID như sau:

**

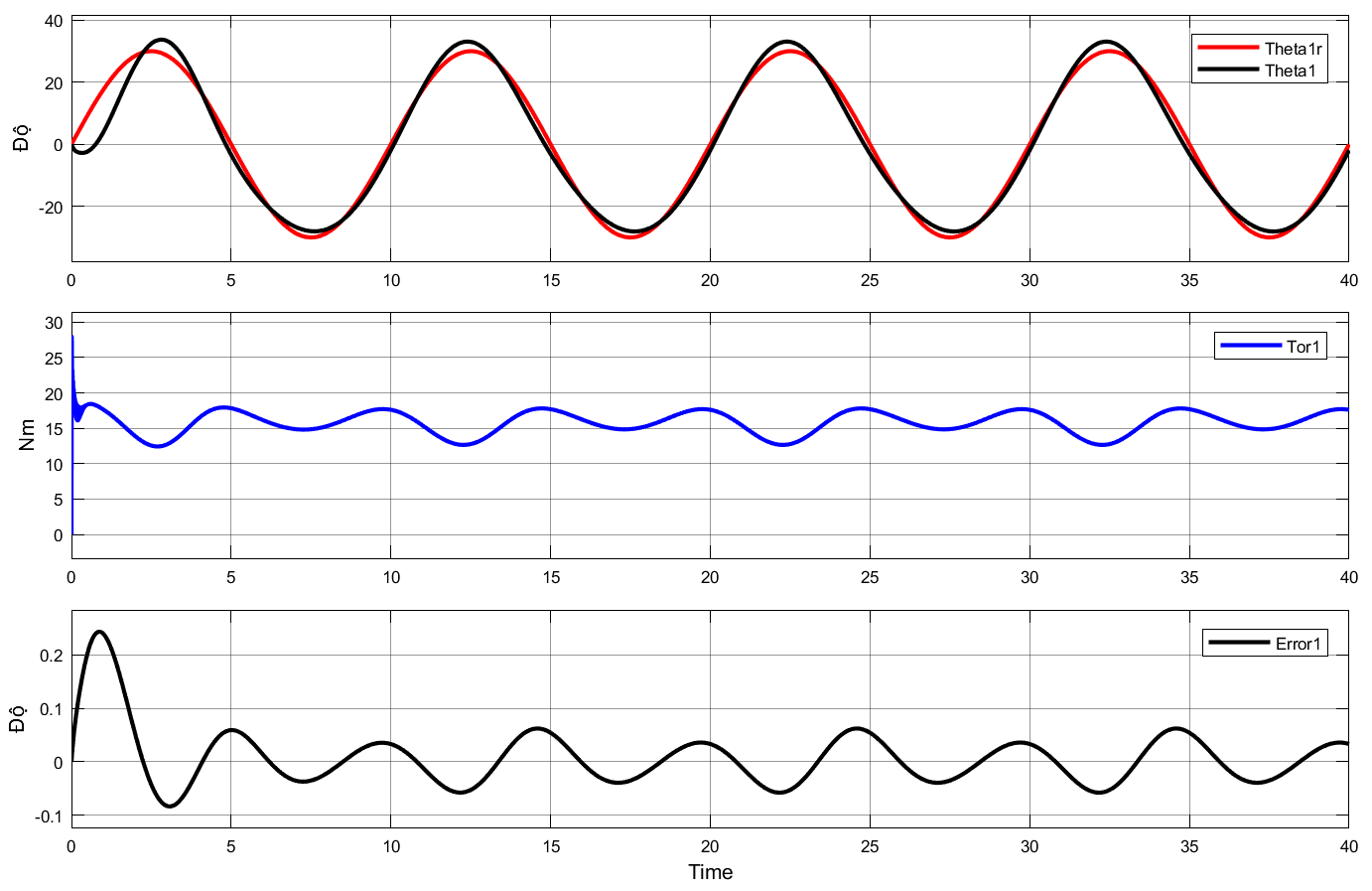
*Hình 40 Kết quả các thông số PID khi chạy xong chương trình giải thuật GA*



*Hình 41 Thay đổi hàm thích nghi qua các thế hệ*

**

*Hình 42 Đồ thị đáp ứng của khớp 1, tín hiệu điều khiển tor1, và sai số so với tín hiệu đặt 1*

**

*Hình 43 Đồ thị đáp ứng của khớp 2, tín hiệu điều khiển tor2, và sai số so với tín hiệu đặt 2*

***Nhận xét:***

Với KP1 , KI1, KD1, KP2 ,KI2 ,KD2 thu được sau khi chạy chương trình giải thuật di truyền, dựa vào hình 42 và hình 43, ta thấy tín hiệu ra đáp ứng tốt, bám theo giá trị đặt, sai số dao động quanh 0, nhỏ tín hiệu điều khiển tốt. Giá trị của hàm thích nghi trong quá trình lai ghép từ thế hệ cũng hội tụ.

# TÀI LIỆU THAM KHẢO

[1] PGS. Ts. Huỳnh Thái Hoàng, Giáo trình điều khiển thông minh.

[2] PGS. Ts. Huỳnh Thái Hoàng, Hệ thống điều khiển thông minh, NXB Đại Học Quốc Gia.

[3] PGS. Ts. Nguyễn Thị Phương Hà, Lý Thuyết Điều Khiển Hiện Đại, NXB Đại Học Quốc Gia

[4] PGS. Ts. Dương Hoài Nghĩa, Công nghệ tính toán mềm.

Tài liệu nguồn trên Web Youtube:

[5] <https://www.youtube.com/watch?v=zJdgWz-UbTI&list=PLv2J3W-wi7ogaIIP3IZiP9EuIvLfXc8NU&index=2>