

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

KHOA ĐIỆN - ĐIỆN TỬ

BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

LUẬN VĂN TỐT NGHIỆP

ĐỀ TÀI:

**ĐIỀU KHIỂN TRƯỢT TỔNG HỢP CÂN BẰNG VÀ BÁM QUỲ ĐẠO HỆ
CART AND POLE KẾT HỢP GIẢM CHATTERING**

Giảng viên hướng dẫn: PGS.TS Nguyễn Minh Tâm

Sinh viên thực hiện: Hồng Gia Bảo - 16151113

TP. HỒ CHÍ MINH – 3/2020

TRƯỜNG ĐH SPKT TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN-ĐIỆN TỬ

ĐỘC LẬP - TỰ DO - HẠNH PHÚC

BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN

----o0o----

Tp. HCM, ngày tháng năm

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên: Hồng Gia Bảo

MSSV: 16151113

Chuyên ngành: Công nghệ Kỹ thuật Điều khiển và Tự động hóa

Hệ đào tạo: Đại học chính quy

Khóa: K2016 Lớp: 161512B

I. TÊN ĐỀ TÀI:

ĐIỀU KHIỂN TRƯỢT TỔNG HỢP CÂN BẰNG VÀ BÁM QUỲ ĐẠO HỆ
CART AND POLE KẾT HỢP GIẢM CHATTERING

II. NHIỆM VỤ

1. Các số liệu ban đầu:

(ghi những thông số, tập tài liệu tin hiệu, hình ảnh,...)
.....
.....

2. Nội dung thực hiện:

- Xây dựng phương trình toán học hệ Cart and Pole, nhận diện thông số động cơ, xây dựng bộ điều khiển trượt kết hợp giảm chattering.
- Mô phỏng hệ Cart and Pole và bộ điều khiển trượt trên Matlab/Simulink.
- Xây dựng phần cơ khí và điện tử cho hệ Cart and Pole.
- Lập trình điều khiển hệ Cart and Pole cân bằng và bám theo quỹ đạo sử dụng STM32F407.

III. NGÀY GIAO NHIỆM VỤ: 3/3/2020

IV. NGÀY HOÀN THÀNH NHIỆM VỤ: 20/07/2020

V. HỌ VÀ TÊN CÁN BỘ HƯỚNG DẪN: PSG.TS Nguyễn Minh Tâm

CÁN BỘ HƯỚNG DẪN

BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN

TRƯỜNG ĐH SPKT TP. HỒ CHÍ MINH

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

KHOA ĐIỆN-ĐIỆN TỬ

ĐỘC LẬP - TỰ DO - HẠNH PHÚC

BỘ MÔN TỰ ĐỘNG ĐIỀU KHIỂN

----o0o----

Tp. HCM, ngày tháng năm

LỊCH TRÌNH THỰC HIỆN ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Hồng Gia Bảo

MSSV: 16151113

Tên đề tài: Điều khiển trượt tổng hợp cân bằng và bám quỹ đạo hệ Cart and Pole kết hợp giảm chattering.

<i>Ngày</i>	<i>Nội dung</i>	<i>Xác nhận GVHD</i>
3/3 - 8/3	Xây dựng phương trình toán học hệ Cart and Pole	
10/3 - 15/3	Thiết kế bộ điều khiển Sliding Mode ConTrol	
17/3 - 22/3	Nhận diện thông số động cơ	
24/3 - 29/3	Nghiên cứu phương pháp giảm chattering	
1/4 - 6/4	Nhận diện thông số động cơ	
8/4 - 13/4	Đo đạc lấy các thông số hệ thống (cân nặng con lắc, xe, ...)	
15/4 - 20/4	Tối ưu thông số trượt bằng Genertic Algorithm	
22/4 - 27/4	Mô phỏng hệ Cart and Pole và bộ điều khiển trượt trên Matlab/Simulink	
29/4 - 11/5	Xây dựng phần cơ khí cho hệ Cart and Pole	
13/5 - 6/6	Lập trình điều khiển hệ Cart and Pole trên STM32F4	
8/6 - 1/7	Viết báo cáo đồ án tốt nghiệp	

GV HƯỚNG DẪN

(Ký và ghi rõ họ và tên)

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ
THUẬT THÀNH PHỐ HỒ CHÍ MINH**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT
NAM**

Độc lập - Tự do - Hạnh phúc

ĐỀ CƯƠNG LUẬN VĂN TỐT NGHIỆP

Họ tên sinh viên:

Hồng Gia Bảo

Ngày tháng năm sinh:

23/10/1998

Khóa học:

2016-2020

Số điện thoại liên hệ:

0359868282

Tên đề tài luận văn:

Điều khiển trượt tổng hợp cân bằng và
bám quỹ đạo hệ Cart and pole kết hợp
giảm chattering.

Chuyên ngành:

Kỹ thuật điều khiển và tự động hóa

Giảng viên hướng dẫn:

PGS. TS Nguyễn Minh Tâm

Ý KIẾN CỦA NGƯỜI HƯỚNG DẪN VÀ HỘI ĐỒNG PHẢN BIỆN

1. Ý kiến của người hướng dẫn

.....

.....

.....

.....

.....

Tp.HCM, ngày, tháng..... , năm 2020

Người hướng dẫn

2. Ý kiến của Hội đồng khoa học Khoa

.....

.....

.....

.....

.....

.....

Tp.HCM, ngày..., tháng..., năm 2020

T/M Hội đồng khoa học Khoa

(Ký và ghi rõ họ tên)

CHƯƠNG 1

TỔNG QUAN VỀ HỆ THỐNG CART AND POLE

1.1 Đặt vấn đề

Cart and Pole, còn gọi là hệ Inverted Pendulum hay Con lắc ngược trên xe [1] thường được sử dụng nhiều trong các phòng nghiên cứu, thí nghiệm về lý thuyết điều khiển vì tính bất ổn định cao của nó. Mô hình gọn, thuận tiện di chuyển, cùng với đó là một hệ phi tuyến bất ổn định nên rất phù hợp trong việc dạy và học về các giải thuật điều khiển cân bằng ở các trường Đại học không chỉ ở Việt Nam mà còn trên toàn thế giới. Đây là một đại diện rất tốt cho các mô hình cần điều khiển ổn định như tên lửa (bám quỹ đạo), như tàu thuyền cân bằng trên biển hoặc như những trục máy cần được điều khiển để ổn định kịp thời,...

Hệ thống Cart and Pole gồm các thành phần như sau: một mô hình xe di chuyển qua lại trên thanh ray trượt, một động cơ DC đầu thanh ray để kéo thanh ray (thiết kế bằng dây curoa) làm xe di chuyển. Trên xe gắn một Encoder quang với 1 trục tự do, trục này được gắn với con lắc đơn đồng chất. Encoder này có tác dụng sẽ feedback về vị trí của con lắc đơn (sau quá trình tính toán, quy đổi) sau đó gửi dữ liệu về vi xử lý STM32F4 để thực hiện quá trình so sánh và đưa ra tín hiệu xử lý phù hợp. Con lắc ở trạng thái bình thường sẽ hướng xuống, nhưng vì chiều dài con lắc khá dài nên sinh viên quyết định cho nằm chéo và sử dụng góc offset để bù vào. Sau khi đưa con lắc lên vị trí thẳng đứng (TOP position), hệ sẽ bắt đầu tự điều khiển. Trong quá trình này, cả xe và con lắc đơn sẽ được điều khiển chuyển động theo tín hiệu sinh viên thiết lập. Trong đề tài luận văn này, sinh viên sẽ điều khiển hệ với 4 trường hợp: Cân bằng, bám quỹ đạo xung vuông, bám quỹ đạo xung sin và bám quỹ đạo xung vuông kết hợp tần số cao (bám quỹ đạo: Xe sẽ được di chuyển theo quỹ đạo cho trước nhưng con lắc vẫn sẽ được giữ thẳng đứng).

Nhiều năm qua, trên các trường đại học trên thế giới ở các cấp bậc bao gồm cả sinh viên, nghiên cứu sinh đã thực hiện các đề tài liên quan đến hệ thống under-

actuated system (hay còn gọi là SIMO – simple input multiple output - system) [2] [3]. Mỗi bài báo khoa học hay luận văn đều liên quan góp phần giải quyết nhiều khía cạnh trong ngành khoa học này. Tuy nhiên, việc giải quyết tất cả các vấn đề cơ bản liên quan đến hệ con lắc ngược xe vẫn còn hạn chế.

Việt Nam đang ngày càng phát triển ngành khoa học điều khiển, với mong muốn tìm hiểu rõ hơn về các giải thuật điều khiển phi tuyến cũng như nắm bắt được các phương pháp tiên tiến đang ngày càng được hoàn thiện và đổi mới, ta cần phải nắm được những giải thuật kinh điển được áp dụng rộng rãi ngày nay [4] [5] [6] – do đó, sinh viên chọn một trong số đó là giải thuật điều khiển phi tuyến Sliding Mode Control sử dụng trong đề tài luận văn tốt nghiệp lần này. Và để củng cố hơn khả năng điều khiển và tính ổn định của giải thuật, đề tài luận văn sẽ thực hiện điều khiển bám quỹ đạo cho hệ xe con lắc ngược trên xe – Cart and Pole (C&P). Tuy nhiên, giải thuật trên có một hạn chế thường gặp là chattering (được đề cập kỹ hơn trong chương 3). Trong giới hạn đề tài luận văn tốt nghiệp, sinh viên đưa ra phương pháp dùng Linear Regression để giảm thiểu một phần tồn thất của vấn đề trên. Ngoài ra, các thông số của giải thuật trượt được tối ưu hoá bằng giải thuật di truyền GA.

1.2 Mục tiêu đề tài

- Sử dụng phương trình Euler- Lagrange tìm phương trình động lực học của hệ Cart and Pole.
- Mô phỏng hệ thống C&P trên Matlab/Simulink
- Xây dựng bộ điều khiển trượt tổng hợp dựa theo tiêu chuẩn Lyapunov để giữ cân bằng và điều khiển hệ thống theo quỹ đạo đặt trước.
- Dùng Linear Regression xây dựng phương trình giảm chattering.
- Kiểm chứng bộ điều khiển trượt tổng hợp trên hệ thống thực tế.

1.3 Nội dung công việc thực hiện

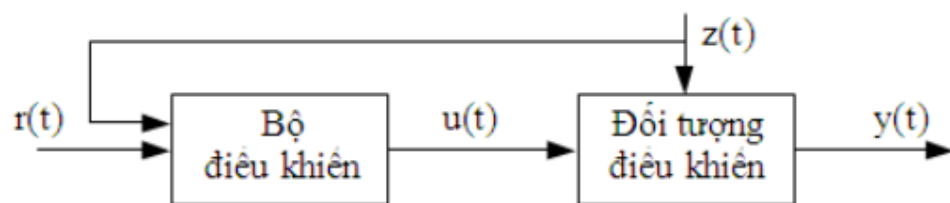
- Xây dựng phương trình toán học mô tả hệ thống Cart and Pole.

- Xây dựng giải thuật điều khiển trượt tổng hợp điều khiển ổn định hệ thống Cart and Pole và điều khiển bám theo quỹ đạo đặt trước.
- Dùng giải thuật di truyền GA để tìm tham số giải thuật trượt.
- Xây dựng phương trình tìm nuy dựa vào Linear Regression giảm thiểu hiện tượng chattering.
- Mô phỏng hệ thống Cart and Pole và các bộ điều khiển trên Matlab/Simulink.
- Lập trình điều khiển hệ thống Cart and Pole trên mô hình thực tế.

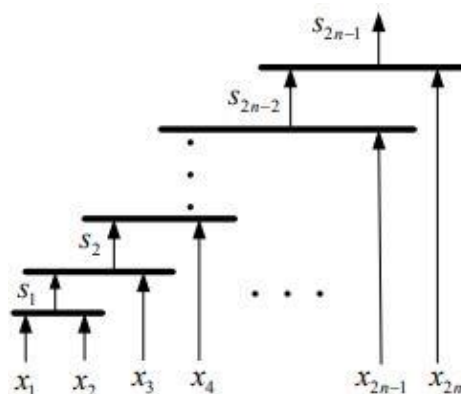
1.4 Giới thiệu về đối tượng và phương pháp nghiên cứu

1.4.1 Đối tượng nghiên cứu

- Nghiên cứu phương trình toán học của hệ thống Cart and Pole. Nghiên cứu giải thuật trượt điều khiển hệ thống.



Hình 1 Sơ đồ khối tổng quát bộ điều khiển hệ thống Cart and Pole



Hình 2 Cấu trúc mặt trượt tổng hợp gồm n hệ thống con

- Nghiên cứu giải thuật di truyền (GA) tìm và tối ưu thông số điều khiển của bộ điều khiển trượt.
- Nghiên cứu phương pháp Linear Regression (một phương pháp cơ bản của Machine Learning).
- Nghiên cứu lập trình STM32 trên Matlab/Simulink để điều khiển hệ thống Cart and Pole thực tế.

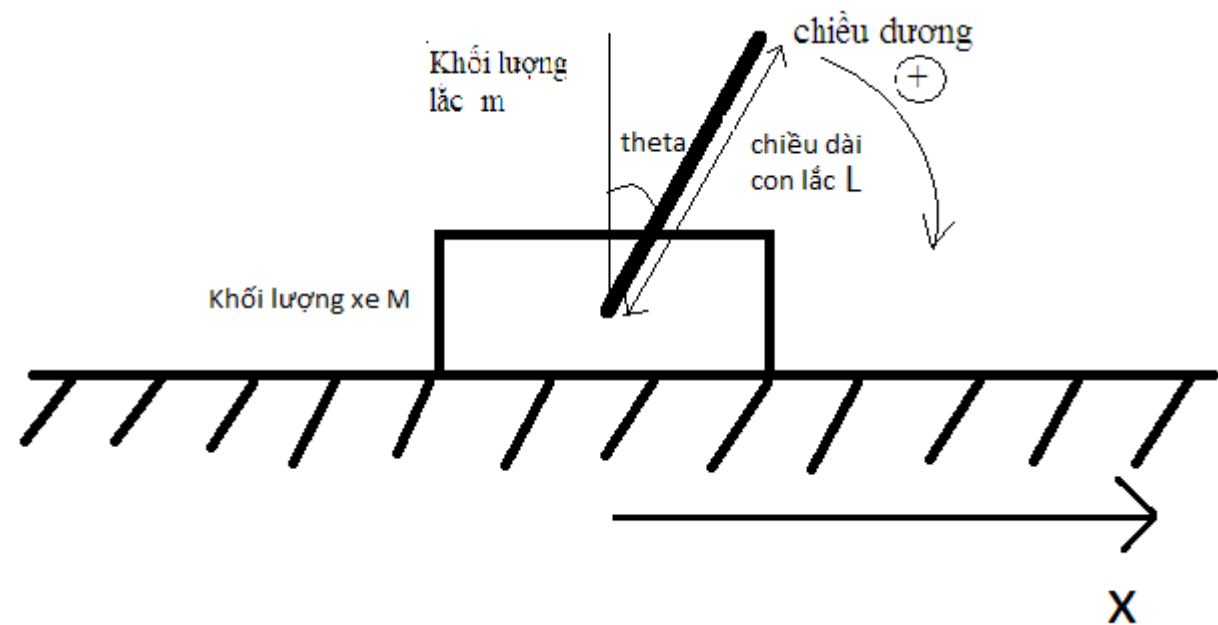
1.5 Ý nghĩa khoa học

Như sinh viên đã trình bày trên mục 1.1, với việc tiếp cận với các công nghệ 4.0 và hơn nữa, khi thế giới đang dần đi đến tự động hoá, thì việc tối ưu trong phương pháp điều khiển vẫn đang rất được ưu tiên. Các bài báo đầu ngành trên thế giới hằng năm vẫn cho ra các nguyên cứu khoa học trong việc cải tiến và phát triển. Ở Việt Nam, chúng ta cũng nên dần tiếp cận với việc nghiên cứu các phương pháp điều khiển, nâng cao kiến thức và củng cố nền tảng cho sinh viên trước khi tiếp nhận việc nghiên cứu ra nhưng cải tiến hơn trong điều khiển tự động. Có rất nhiều ví dụ về tự động điều khiển trên thế giới như: Tên lửa, xe tự hành, tàu tự hành, máy bay không người lái UAV, Turtlebot, điều khiển vị trí, vận tốc motor,... [7] [8] [9]. Dựa vào đó, sinh viên mong muốn tiếp cận và nghiên cứu về giải thuật phi tuyến Sliding mode control để kiểm chứng khả năng điều khiển của các giải thuật này trên mô hình bất ổn định, từ đó phát triển và áp dụng vào các mô hình lớn hơn và mang tính chất quan trọng hơn ngoài thực tế.

Chương 2: Mô hình toán học và giải thuật điều khiển con lắc ngược trên xe

2.1 Xây dựng phương trình toán học

Hệ thống Cart and Pole được diễn tả như Hình 3. Từ đó, các phương trình toán học của hệ được trình bày tuần tự từ (1) đến (21).



Hình 3 Mô hình mô tả hệ thống con lắc ngược trên xe

Các thông số của hệ thống con lắc ngược trên xe được trình bày như trên Bảng 1:

Bảng 1

Thông số	Mô tả thông số	Đơn vị	Giá trị
L	Chiều dài con lắc	m	0.48
m	Khối lượng con lắc	Kg	0.23
M	Khối lượng xe	Kg	0.39
J_1	Moment quán tính con lắc	Kgm^2	$\frac{1}{12}(m^2 + 0.003^2)$
θ	Góc con lắc so với phương thẳng đứng	rad	

x	Vị trí chiếc xe	m	
τ	Lực tác dụng vào hệ thống	N	

Như trên Hình 3, con lắc sẽ xoay quanh trục một encoder được gắn trên xe. Mục tiêu của hệ là giữ cho con lắc luôn nằm ở vị trí thẳng bằng (thẳng đứng hướng lên – TOP position), ngoài ra, vị trí của xe sẽ đi theo một quỹ đạo cho trước - điều khiển bám quỹ đạo - mà không làm con lắc bị rơi xuống. Lúc này, bộ điều khiển hệ thống sẽ hoạt động để đảm bảo các yêu cầu trên.

Dựa vào phương trình Euler – Lagrange, phương trình toán học hệ con lắc ngược trên xe được tìm như sau:

$$\frac{d\left(\frac{\partial L}{\partial \dot{q}}\right)}{dt} - \frac{\partial L}{\partial q} = Q \quad (1)$$

Với : $L = T - V$ là toán tử Lagrange, T = động năng hệ thống, V = thế năng, Q = tổng ngoại lực, $q = \begin{bmatrix} x \\ \theta \end{bmatrix}$, $Q = \begin{bmatrix} F \\ 0 \end{bmatrix}$

Vị trí con lắc ngược theo 2 phương x và y là:

$$r_{1x} = L \sin \theta + x \quad (2)$$

$$r_{1y} = L \cos \theta \quad (3)$$

Từ (2) - (3), vận tốc con lắc theo 2 phương là:

$$\dot{r}_{1x} = L\dot{\theta} \cos \theta + \dot{x} \quad (4)$$

$$\dot{r}_{1y} = -L\dot{\theta} \sin \theta \quad (5)$$

Từ đó, ta có bình phương vận tốc con lắc và vận tốc xe như phương trình (6) – (7):

$$v_1^2 = \dot{r}_{1x}^2 + \dot{r}_{1y}^2 \quad (6)$$

$$v_2^2 = \dot{x}^2 \quad (7)$$

Động năng hệ thống:

$$T = T_{pole} + T_{cart} \quad (8)$$

Trong đó, động năng con lắc và động năng xe lần lượt như sau:

$$T_{pole} = \frac{1}{2} m v_1^2 + \frac{1}{2} J_1 \dot{\theta}^2 \text{ (động năng truyền thẳng và động năng quay)} \quad (9)$$

$$T_{cart} = \frac{1}{2} M v_2^2 \quad (10)$$

Từ (8), (9) và (10), động năng hệ thống có dạng sau:

$$\begin{aligned} T &= \frac{1}{2} m (\dot{r}_{1x}^2 + \dot{r}_{1y}^2) + \frac{1}{2} J_1 \dot{\theta}^2 + \frac{1}{2} M \dot{x}^2 \\ &= \frac{1}{2} m (\dot{x}^2 + 2L\dot{\theta}\dot{x}\cos\theta + L^2\dot{\theta}^2) + \frac{1}{2} J_1 \dot{\theta}^2 + \frac{1}{2} M \dot{x}^2 \end{aligned} \quad (11)$$

Thế năng của hệ thống sẽ được tính theo thế năng của con lắc, vì gốc thế năng được đặt ở vị trí xe nên xem như xe không có thế năng:

$$V = m r_{1y} g = m L g \cos \theta \quad (12)$$

Từ động năng và thế năng hệ thống được trình bày ở trên, toán tử Lagrange trở thành:

$$L = T - V = \frac{1}{2} m (\dot{x}^2 + 2L\dot{\theta}\dot{x}\cos\theta + L^2\dot{\theta}^2) + \frac{1}{2} J_1 \dot{\theta}^2 + \frac{1}{2} M \dot{x}^2 \quad (13)$$

Từ phương trình trên, các thành phần của phương trình (1) trở thành:

$$\frac{\partial L}{\partial x} = 0 \quad (14)$$

$$\frac{\partial L}{\partial \dot{x}} = (m + M) \dot{x} + mL\dot{\theta} \cos \theta \quad (15)$$

$$\frac{\partial L}{\partial \dot{x}} = (m + M) \dot{x} + mL\dot{\theta} \cos \theta \quad \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) = (m + M) \ddot{x} + mL\ddot{\theta} \cos \theta - mL\dot{\theta}^2 \sin \theta \quad (16)$$

$$\frac{\partial L}{\partial \theta} = -mL\dot{\theta} \sin \theta + mLg \sin \theta \quad (17)$$

$$\frac{\partial L}{\partial \dot{\theta}} = mL\dot{x} \cos \theta + (J_1 + mL^2) \dot{\theta} \quad (18)$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = mL\ddot{x} \cos \theta - mL\dot{x} \sin \theta + (J_1 + mL^2) \ddot{\theta} \quad (19)$$

Thay các phương trình từ (14) đến (19) vào (1), hệ phương trình trạng thái của hệ con lắc ngược trên xe với ngõ vào là lực do động cơ gây ra được trình bày:

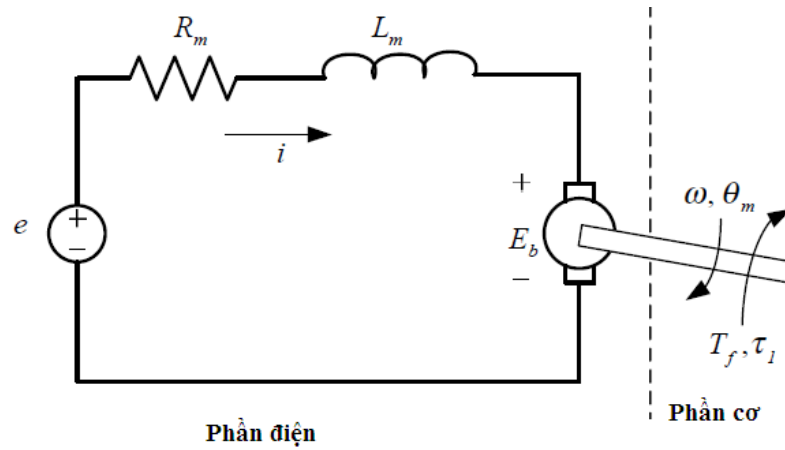
$$\begin{cases} (m + M) \ddot{x} + mL\ddot{\theta} \cos \theta - mL\dot{\theta}^2 \sin \theta = F \\ mL\ddot{x} \cos \theta + (J_1 + mL^2) \ddot{\theta} - mLg \sin \theta = 0 \end{cases} \quad (20)$$

Phương trình trên được viết dưới dạng ma trận như sau:

$$M(q)\ddot{q} + V_m(q, \dot{q}) + G(q) = \begin{bmatrix} F & 0 \end{bmatrix}^T \quad (21)$$

Trong đó: $M(q) = \begin{bmatrix} m + M & mL \cos \theta \\ mL \cos \theta & J_1 + mL^2 \end{bmatrix}$; $V_m(q, \dot{q}) = \begin{bmatrix} 0 & -mL\dot{\theta}^2 \sin \theta \\ 0 & 0 \end{bmatrix}$; $G(q) = \begin{bmatrix} 0 \\ -mLg \sin \theta \end{bmatrix}$

Tuy nhiên, trên phương trình (20) hay (21) thì tín hiệu được cấp vào để điều khiển hệ thống là lực của động cơ, việc này nếu chỉ mô phỏng thì có thể được sử dụng, nhưng nếu muốn điều khiển trên thực tế, ta cần phải chuyển đổi lực này sang tín hiệu điện áp cấp vào cho động cơ. Do vậy, việc tính toán phương trình động học của động cơ cũng như việc tìm thông số cho động cơ là cần thiết. Theo tài liệu [10], động cơ được chia thành các phần như sau: điện và cơ.



Hình 4. Sơ đồ động cơ

Trong động cơ, các thông số ta cần chú ý khi chuyển đổi từ Torque sang điện áp cấp vào động cơ u:

R_m : điện trở động cơ (ohm).

L_m : hệ số điện kháng (H).

E_b : Sức điện động (V).

J_m : moment quán tính của rotor (kgm^2).

K_b : Hằng số phản điện (V/(rad/sec)).

K_t : Hằng số moment (Nm/A).

τ_m : mô men xoắn nội (Nm).

θ_m : góc xoay trục động cơ (rad).

T_f : momen ma sát (Nm).

τ_l : momen xoắn cản (Nm)

ω : vận tốc motor (rad/s).

C_m : hệ số ma sát nhớt (Nm/(rad/sec)).

Giá trị các thông số này được trình bày trong Bảng 2.

Áp dụng định luật Kirchoff cho mạch điện phản ứng:

$$E = L_m \frac{di}{dt} + R_m i + E_b \quad (22)$$

Với: $E_b = K_b \omega$

Áp dụng định luật Newton cho chuyển động quay của trục động cơ:

$$J_m \frac{d\omega}{dt} + T_f + C_m \omega + \tau_1 = \tau_m \quad (23)$$

Với: $\tau_m = K_t i$

$$\Rightarrow J_m \frac{d\omega}{dt} = K_t i - T_f - C_m \omega - \tau_1 \quad (24)$$

Ta có, công suất điện và công suất động cơ lần lượt như sau:

$$P_e = E_b i \quad (25)$$

$$P_m = \tau_m \omega \quad (26)$$

Áp dụng định luật bảo toàn năng lượng, ta có được phương trình (27):

$$P_e = P_m \quad (27)$$

$$\Rightarrow E_b i = \tau_m \omega \quad (28)$$

$$\Rightarrow K_b \omega i = K_t i \omega \quad (29)$$

$$\Rightarrow K_b = K_t \quad (30)$$

Lưu ý: Nếu K_b và K_t dùng đơn vị MKS (K_b có đơn vị V/rad/sec và K_t có đơn vị Nm/A). Khi đó, $K_b = K_t$.

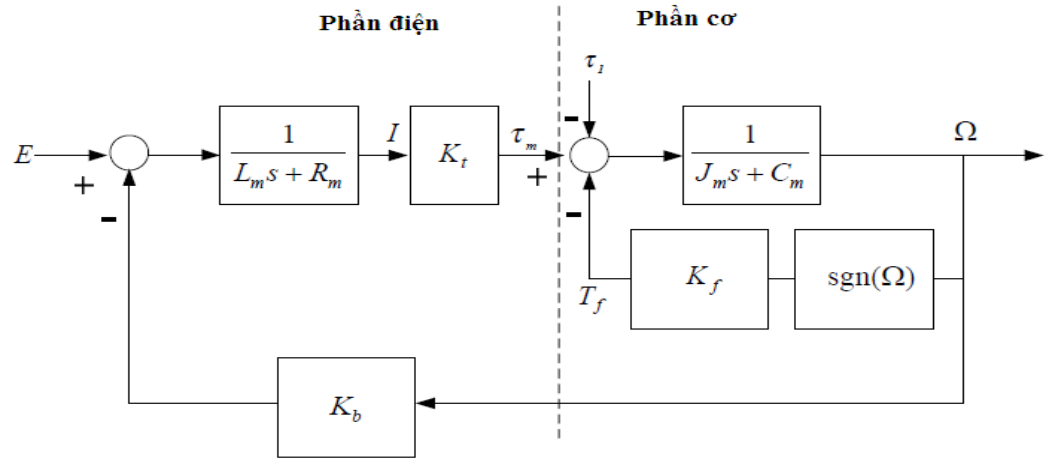
Biến đổi Laplace phương trình (22) và (23) theo tần số như sau:

$$(L_m s + R_m) I(s) + K_b \Omega(s) = E(s) \quad (31)$$

$$K_t I(s) - (J_m s + C_m) \Omega(s) = T_f(s) + \tau_1(s) \quad (32)$$

Ta giả thiết T_f là hằng số và $T_f = K_f \text{sgn}(\Omega)$, $K_f = \text{const}$ và

$$\text{sgn}(\Omega) = \begin{cases} 1 & \Omega > 0 \\ 0 & \Omega = 0 \\ -1 & \Omega < 0 \end{cases}$$



Hình 5 Sơ đồ khối động cơ DC theo hàm truyền

Động cơ sử dụng trong đề tài luận văn tốt nghiệp này được dùng trong nhiều trường hợp khởi động, hãm, đảo chiều động cơ ngay lập tức,... Trong những trường hợp này, vì điện cảm trong động cơ nhỏ nên có thể bỏ qua:

Do đó, ta có thể bỏ qua thành phần $L_m \frac{di}{dt}$. Do đó, (22) trở thành:

$$E = R_m i + K_b \omega \quad (33)$$

$$\rightarrow i = \frac{U - K_b \omega}{R_m} \quad (E = U). \text{ Từ đó, ta có:}$$

$$\tau_m = K_t i = \frac{K_t}{R_m} (U - K_b \omega) = \frac{K_t}{R_m} U - \left(\frac{K_t K_b}{R_m} \right) \omega \quad (34)$$

Với cấu trúc truyền động từ trục động cơ sang bánh xe, vị trí xe được tính như sau:

$$x = \frac{R \theta_m}{d_l} \quad (35)$$

Với: R là bán kính bánh xe và d_l là hệ số tỉ lệ bánh răng truyền động.

Từ (35), ta có $\dot{x} = \frac{R\omega}{d_l}$. Thay kết quả trên vào (34), ta được :

$$\tau_m = \frac{K_t}{R_m} U - \frac{K_b K_t}{R_m R} \dot{x} \quad (36)$$

Bỏ qua T_f , ta tiến hành thay (36) vào (24), ta có :

$$\tau_1 = -\frac{J_m d_l}{R} \ddot{x} - d_l \left(\frac{C_m}{R} + \frac{K_b K_t}{R_m R} \right) \dot{x} + \frac{K_t}{R_m} U \quad (37)$$

Lực tác dụng lên xe được tính như sau:

$$F = \frac{d_l \tau_1}{R} = \frac{d_l}{R} \left[\frac{K_t}{R_m} U - d_l \left(\frac{K_b K_t}{R_m R} + \frac{C_m}{R} \right) \dot{x} - \frac{J_m d_l}{R} \ddot{x} \right] \quad (38)$$

Đặt $k_1 = \frac{d_l K_t}{R_m R}$; $k_2 = \frac{d_l^2 K_t K_b}{R^2 R_m} + \frac{d_l^2 C_m}{R^2}$; $k_3 = \frac{d_l^2 J_m}{R^2}$. Qua cách đặt trên, (38) trở thành

$$F = k_1 U - k_2 \dot{x} - k_3 \ddot{x} \quad (39)$$

Với $e = u$ là điện áp cấp cho động cơ

Từ các kết quả trên, phương trình động lực học hệ xe con lắc ngược có dạng sau :

$$\begin{cases} (m+M)\ddot{x} + mL\ddot{\theta} \cos \theta - mL\dot{\theta}^2 \sin \theta = k_1 U - k_2 \dot{x} - k_3 \ddot{x} \\ mL\ddot{x} \cos \theta + (J_1 + mL^2)\ddot{\theta} - mLg \sin \theta = 0 \end{cases} \quad (40)$$

$$M_f(q)\ddot{q} + V_{mf}(q, \dot{q})\dot{q} + G_f(q) = \begin{bmatrix} k_1 u \\ 0 \end{bmatrix}$$

$$\text{Với : } M_f(q) = \begin{bmatrix} m+M+k_3 & mL \cos \theta \\ mL \cos \theta & J_1 + mL^2 \end{bmatrix} ; V_{mf} = \begin{bmatrix} k_2 & -mL\dot{\theta}^2 \sin \theta \\ 0 & 0 \end{bmatrix} ; G_f = \begin{bmatrix} 0 \\ -mLg \sin \theta \end{bmatrix}$$

Từ (42), hệ phương trình biến trạng thái của hệ C&P được viết lại như sau

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(x) + g_1(x)u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(x) + g_2(x)u \end{cases} \quad (41)$$

$$\text{Với : } x = [x_1 \quad x_2 \quad x_3 \quad x_4]^T = [x \quad \dot{x} \quad \theta \quad \dot{\theta}]^T ;$$

Bảng 2. Thông số động cơ

Thông số	Đơn vị	Giá trị
Kb	V/rad/s	0.053
Kt	Nm/A	0.053
Rm	Ohm	2.7
Jm	Kg.m ²	4.76*10 ⁻⁵
Cm	Nm/(rad/s)	7.636*10 ⁻⁵
Radius	m	0.015

Nhận diện thông số động cơ:

Với các trường hợp người nghiên cứu là sinh viên có kinh tế không tốt, thì với những động cơ cũ đã qua sử dụng và không được cung cấp các thông số cần thiết cho việc nghiên cứu điều khiển, theo [10], ta có phương pháp để nhận diện động cơ như sau:

Ta có:

$$\frac{di(t)}{dt} \approx \frac{i(t_{n+1}) - i(t_n)}{\Delta t} \quad (42)$$

$$\frac{d\omega(t)}{dt} \approx \frac{\omega(t_{n+1}) - \omega(t_n)}{\Delta t} \quad (43)$$

Với: $\Delta t = t_{n+1} - t_n$

Từ phương trình (22), ta có:

$$e(t_n) = R_m \cdot i(t_n) + L_m \frac{i(t_{n+1}) - i(t_n)}{\Delta t} + K_b \cdot \omega(t_n) \quad (44)$$

$$\Rightarrow \Delta t \cdot e(t_n) = \Delta t \cdot R_m \cdot i(t_n) + L_m (i(t_{n+1}) - i(t_n)) + \Delta t \cdot K_b \cdot \omega(t_n) \quad (45)$$

$$\Rightarrow L_m \cdot i(t_{n+1}) = (L_m - R_m \cdot \Delta t) \cdot i(t_n) - K_b \cdot \Delta t \cdot \omega(t_n) + \Delta t \cdot e(t_n) \quad (46)$$

$$\Rightarrow i(t_{n+1}) = \frac{(L_m - R_m \cdot \Delta t)}{L_m} \cdot i(t_n) - \frac{K_b \cdot \Delta t}{L_m} \cdot \omega(t_n) + \frac{\Delta t}{L_m} \cdot e(t_n) \quad (47)$$

Từ phương trình (23), ta có:

$$K_b.i(t_n) = J_m \frac{\omega(t_{n+1}) - \omega(t_n)}{\Delta t} + C_m \omega(t_n) + K_f \operatorname{sgn}(\omega) \quad (48)$$

$$\Rightarrow K_b.\Delta t.i(t_n) = J_m.\omega(t_{n+1}) - J_m.\omega(t_n) + C_m.\Delta t.\omega(t_n) + K_f \operatorname{sgn}(\omega).\Delta t \quad (49)$$

$$\Rightarrow J_m.\omega(t_{n+1}) = (J_m - C_m.\Delta t).\omega(t_n) + K_b.\Delta t.i(t_n) - K_f \operatorname{sgn}(\omega).\Delta t \quad (50)$$

$$\Rightarrow \omega(t_{n+1}) = \frac{J_m - C_m.\Delta t}{J_m}.\omega(t_n) + \frac{K_b.\Delta t}{J_m}.i(t_n) - \frac{K_f.\operatorname{sgn}(\omega).\Delta t}{J_m} \quad (51)$$

Sau khi ta suy ra được hai phương trình (47), (51), ta làm gọn như sau:

$$i(n+1) = a_1.i(n) + a_2.\omega(n) + a_3.e(n) \quad (52)$$

$$\omega(n+1) = b_1.i(n) + b_2.\omega(n) + b_3.\operatorname{sgn}(\omega) \quad (53)$$

$$\text{Với : } a_1 = \frac{L_m - R_m.\Delta t}{L_m}; a_2 = \frac{-K_b.\Delta t}{L_m}; a_3 = \frac{\Delta t}{L_m};$$

$$b_1 = \frac{K_b.\Delta t}{J_m}; b_2 = \frac{J_m - C_m.\Delta t}{J_m}; b_3 = \frac{-K_f.\Delta t}{J_m}$$

Từ đây, ta sẽ thu thập 2 ma trận n chiều như sau:

$\begin{bmatrix} \omega(2) \\ \omega(3) \\ \vdots \\ \vdots \\ \vdots \\ \omega(n+1) \end{bmatrix} = \begin{bmatrix} i(1)\omega(1) & \operatorname{sgn}(\omega(1)) \\ i(2)\omega(2) & \operatorname{sgn}(\omega(2)) \\ & \vdots \\ & \vdots \\ & \vdots \\ i(n) & \omega(n) & \operatorname{sgn}(\omega(n)) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$	(54)
$\begin{bmatrix} i(2) \\ i(3) \\ \vdots \\ \vdots \\ \vdots \\ i(n+1) \end{bmatrix} = \begin{bmatrix} i(1)\omega(1) & e(1) \\ i(2)\omega(2) & e(2) \\ & \vdots \\ & \vdots \\ & \vdots \\ i(n) & \omega(n) & e(n) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$	(55)

Sinh viên đặt các biến như sau: $I = \begin{bmatrix} i(2) \\ i(3) \\ \vdots \\ \vdots \\ \vdots \\ i(n+1) \end{bmatrix}$; $\Omega = \begin{bmatrix} \omega(2) \\ \omega(3) \\ \vdots \\ \vdots \\ \vdots \\ \omega(n+1) \end{bmatrix}$; $A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$; $B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

$$P_m = \begin{bmatrix} i(1) & \omega(1) & e(1) \\ i(2) & \omega(2) & e(2) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ i(n) & \omega(n) & e(n) \end{bmatrix}; Q_m = \begin{bmatrix} i(1) & \omega(1) & \text{sgn}(\omega(1)) \\ i(2) & \omega(2) & \text{sgn}(\omega(2)) \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ i(n) & \omega(n) & \text{sgn}(\omega(n)) \end{bmatrix}$$

Từ hai phương trình trên, ta tìm được ma trận A, B:

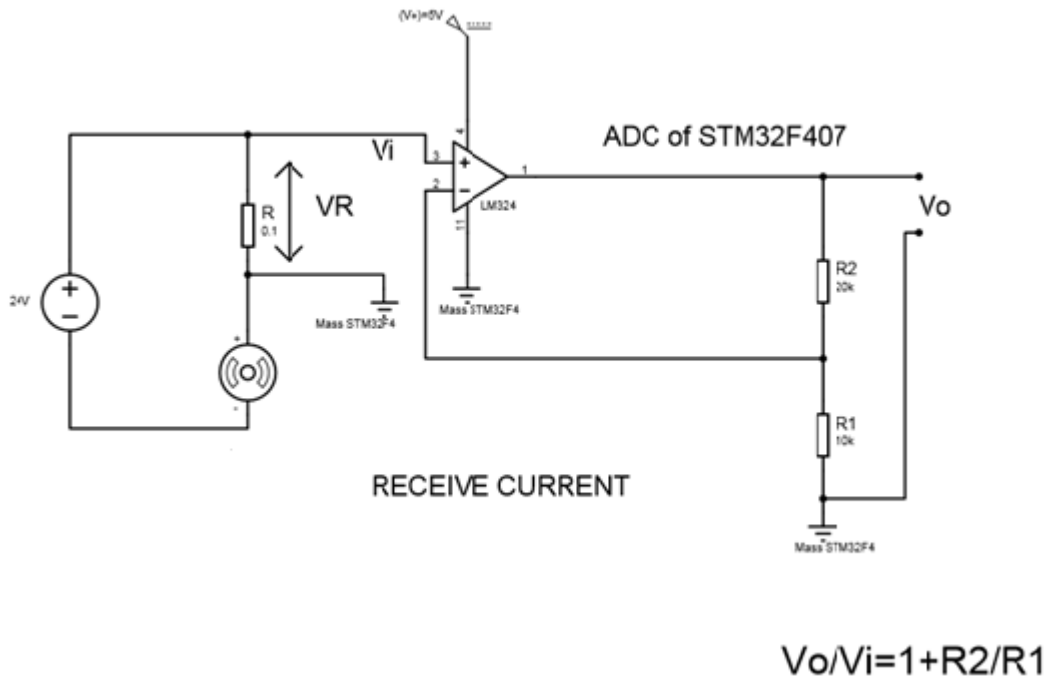
$$\hat{A} = (P_m^T \cdot P_m)^{-1} \cdot P_m^T \cdot I \quad (56)$$

$$\hat{B} = (Q_m^T \cdot Q_m)^{-1} \cdot Q_m^T \cdot \Omega \quad (57)$$

Từ đó, ta tìm được giá trị của $a_1, a_2, a_3, b_1, b_2, b_3$

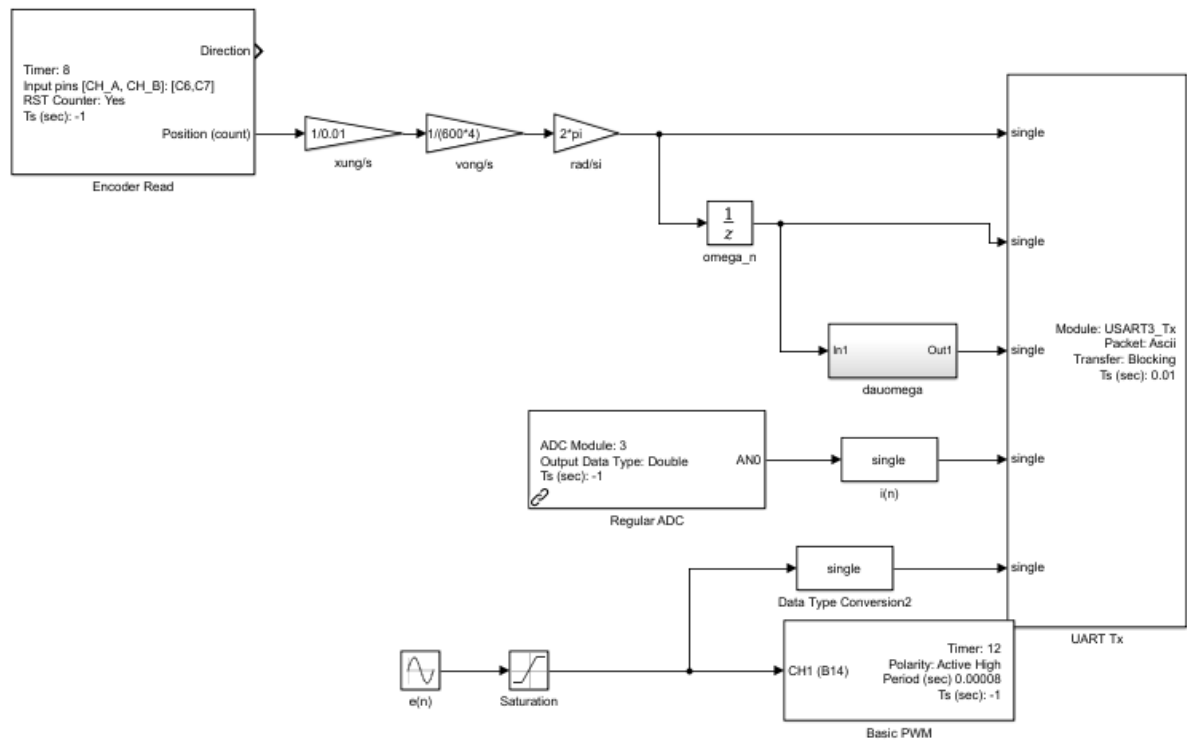
$$\Rightarrow L_m = \frac{\Delta t}{a_3}; K_b = -\frac{a_2}{a_3}; R_m = \frac{(1-a_1)}{a_3}; |T_f| = K_f = \frac{b_3 a_2}{b_1 a_3}; J_m = -\frac{a_2 \Delta t}{a_3 b_1}; C_m = \frac{a_2 (b_2 - 1)}{a_3 b_1}$$

Như vậy có 3 đại lượng cần thu thập Ω, i và e . Trong đó Ω có thể được thu thập thông qua encoder của động cơ, ta sẽ suy ra được vận tốc góc Ω . Điện áp tiêu thụ sẽ có dạng sóng sine từ -24V đến 24V được điều khiển thông qua chân PWM và khối sine trong Matlab/Simulink. Đồng thời, dòng điện tiêu thụ được đọc thông qua mạch như sau:



Hình 6. Sơ đồ mạch điện thu thập giá trị dòng điện đi qua động cơ

Chương trình thu thập dữ liệu trong Matlab/Simulink:



Hình 7. Chương trình thu thập dữ liệu thông qua Matlab/Simulink

Chương 3: Thiết kế bộ điều khiển trượt tổng hợp kết hợp Linear Regression

3.1 Thiết kế giải thuật trượt tổng hợp bám quỹ đạo

3.1.1 Tiêu chuẩn ổn định tiệm cận Lyapunov

Tiêu chuẩn này còn được gọi là tiêu chuẩn thứ 2 của Lyapunov [1'], ngày nay được sử dụng khá rộng rãi trong lĩnh vực điều khiển tự động.

Phát biểu tiêu chuẩn:

Xét hệ thống được mô tả bởi phương trình trạng thái: $\dot{X} = f(x_1, x_2, x_3, \dots, x_n)$

Nếu tìm được một hàm $V(x)$ sao cho với mọi biến trạng thái $x_1, x_2, x_3, \dots, x_n$ là một hàm xác định dương và đạo hàm của nó theo thời gian $\frac{dV(x)}{dt}$ là một hàm xác định âm thì hệ thống sẽ ổn định. Ta gọi lúc này hệ thống ổn định Lyapunov. Và hàm $V(x)$ được gọi là hàm Lyapunov.

- Khi $V(x) \cdot \dot{V}(x) < 0$: hệ thống ổn định tiệm cận (các biến trạng thái hội tụ về 0).
- Khi $V(x) \cdot \dot{V}(x) = 0$: hệ thống ổn định.
- Khi $V(x) \cdot \dot{V}(x) > 0$: hệ thống không ổn định.

3.1.2 Bổ đề Barbalat

Theo [11], bổ đề Barbalat cho biết:

Nếu $f(t)$ có một giới hạn hữu hạn khi $t \rightarrow \infty$ và nếu \dot{f} là liên tục đều (hoặc \dot{f} bị chặn) thì $\dot{f}(t) \rightarrow 0$ khi mà $t \rightarrow \infty$.

Chúng ta biết rằng trong trường hợp của các hệ thống tự hành (thời gian bất biến), nếu \dot{V} là nửa xác định âm (NSD), thì cũng có thể biết được các hành vi tiệm cận

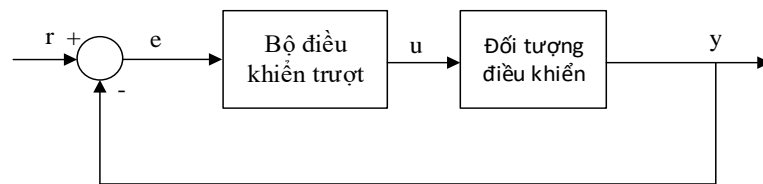
bằng cách sử dụng định lý bất biến nào đó. Tuy nhiên, sự linh hoạt này không sẵn có cho các hệ thống thời gian biến đổi. Đây là nơi "bổ đề Barbalat" được sử dụng tới. Nó nói rằng:

Nếu $V(x, t)$ thỏa mãn những điều kiện sau:

- $V(x, t)$ có một hữu hạn khi $t \rightarrow \infty$.
- $\dot{V}(x, t)$ là nửa xác định âm (tức ≤ 0) (NSD).
- $\dot{V}(x, t)$ là liên tục và đều theo thời gian (được thỏa mãn nếu \ddot{V} là hữu hạn).

Thì $\dot{V}(x, t) \rightarrow 0$ khi $t \rightarrow \infty$.

3.1.1 Lý thuyết bộ điều khiển trượt cơ bản [3']



Hình 8. Sơ đồ bộ điều khiển trượt

Xét hệ thống phi tuyến được biểu diễn bởi phương trình vi phân:

$$\begin{cases} \dot{x}^{(n)} = f(X) + G(X)u \\ y = x \end{cases} \quad (58)$$

Trong đó: $X = [x \quad \dot{x} \quad \dots \quad x^{(n-1)}]^T$ là vector biến trạng thái

u – tín hiệu điều khiển

y – tín hiệu ra

n – bậc hệ thống

Các hàm $f = f(X)$, $g = g(X)$ là các hàm phi tuyến không biết trước nhưng biết trước các chặn trên và dưới của chúng.

$$f_{\min} \leq f \leq f_{\max}, 0 < g_{\min} \leq g \leq g_{\max} \quad (59)$$

Gọi r là tín hiệu đặt. Giả thiết r có đạo hàm theo t đến cấp n .

Định nghĩa:

$$X_d = \begin{bmatrix} x_d \\ x_d^{(1)} \\ \dots \\ x_d^{(n-1)} \end{bmatrix} = \begin{bmatrix} r_d \\ r^{(1)} \\ \dots \\ r^{(n-1)} \end{bmatrix}; E = X - X_d = \begin{bmatrix} x - x_d \\ \dot{x} - \dot{x}_d \\ \dots \\ x^{(n-1)} - x_d^{(n-1)} \end{bmatrix} = \begin{bmatrix} e \\ e^{(1)} \\ \dots \\ e^{(n-1)} \end{bmatrix}$$

Mục tiêu điều khiển là xác định luật điều khiển u sao cho sai lệch E tiến tới 0 khi $t \rightarrow \infty$

Định nghĩa hàm trượt:

$$S = e^{(n-1)} + a_{n-2}e^{(n-1)} + \dots + a_1e^{(1)} + a_0e \quad (60)$$

Các hệ số a_0, a_1, \dots, a_{n-2} phải được chọn sao cho đa thức đặc trưng của phương trình vi phân (60):

$$p^{n-1} + a_{n-2}p^{n-2} + \dots + a_1p + a_0 = 0 \quad (61)$$

có tất cả các nghiệm nằm bên trái mặt phẳng phức (thỏa tiêu chuẩn ổn định Hurwitz) [2']. Khi đó, các nghiệm đặc trưng nằm bên trái mặt phẳng phức, nên $e(t)$ sẽ tiến đến không khi t tiến đến vô cùng.

Trong không gian trạng thái n chiều, phương trình $S=0$ xác định một mặt cong được gọi là mặt trượt (*sliding surface*). Luật điều khiển u được xác định sao cho $S \rightarrow 0$ trong khoảng thời gian hữu hạn. Các quỹ đạo pha của hệ thống được đưa về mặt trượt, bên trên mặt trượt, quỹ đạo pha bám theo điểm X_d một cách tiệm cận.

Để xác định luật điều khiển, đạo hàm (60) theo thời gian, ta có:

$$\begin{aligned}
\dot{S} &= e^{(n)} + a_{n-2}e^{(n-1)} + \dots + a_1e^{(2)} + a_0e^{(1)} \\
&= \left(x^{(n)} - x_d^{(n)}\right) + a_{n-2}\left(x^{(n-1)} - x_d^{(n-1)}\right) + \dots + a_1\left(x^{(1)} - x_d^{(2)}\right) + a_0\left(x^{(1)} - x_d^{(1)}\right) \\
&= f(X) + g(X)u - x_d^{(n)} + a_{n-2}\left(x^{(n-1)} - x_d^{(n-1)}\right) + \dots \\
&\quad + a_1\left(x^{(1)} - x_d^{(2)}\right) + a_0\left(x^{(1)} - x_d^{(1)}\right)
\end{aligned} \tag{62}$$

Sử dụng phương pháp Lyapunov, chọn một hàm xác định dương V có dạng:

$V = \frac{1}{2}S^2$	(63)
----------------------	------

Suy ra:

$\dot{V} = S\dot{S}$	(64)
----------------------	------

Để \dot{V} xác định âm cần chọn luật điều khiển u sao cho: Khi $S > 0$ thì $\dot{S} < 0$, Khi $S < 0$ thì $\dot{S} > 0$, Khi $S = 0$ thì $\dot{S} = 0$

Có thể chọn:

$$\dot{S} = -k \text{sign}(S) \tag{65}$$

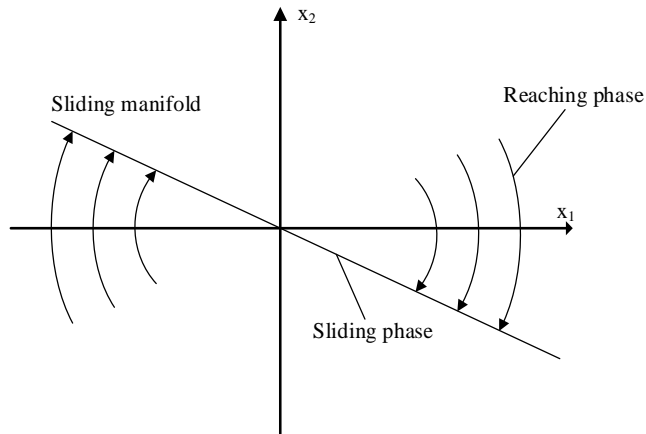
$$\text{Với } k > 0 \text{ và } \text{sign}(S) = \begin{cases} 1, S > 0 \\ 0, S = 0 \\ -1, S < 0 \end{cases}$$

Thay (65) vào (62) ta được luật điều khiển:

$$u = \frac{1}{g(X)} \left[f(X) - x_d^{(n)} + a_{n-2}\left(x^{(n-1)} - x_d^{(n-1)}\right) + \dots + a_1\left(x^{(1)} - x_d^{(2)}\right) + a_0\left(x^{(1)} - x_d^{(1)}\right) + k \text{sign}(S) \right] \tag{66}$$

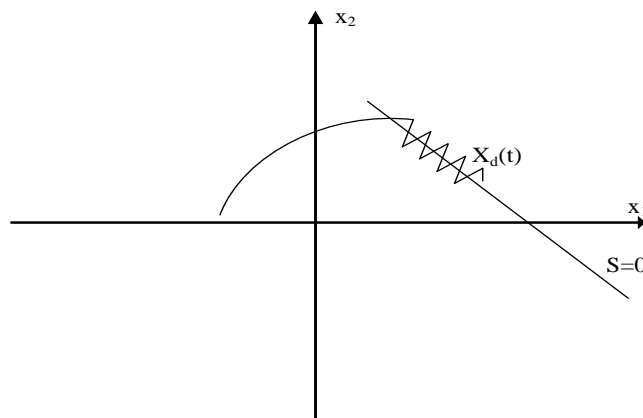
Với k là hằng số dương. Giá trị của k được chọn sao cho trong trường hợp xấu nhất $S\dot{S} < 0$ khi $S \neq 0$ vẫn luôn thỏa mãn.

Luật điều khiển u (66) đưa quỹ đạo pha của hệ thống về mặt trượt và duy trì một cách bền vững trên mặt trượt này.



Hình 9 Giải thuật trượt

Điều khiển trượt lý tưởng đòi hỏi luận điều khiển phải thay đổi tức thời ngay tại thời điểm quỹ đạo pha của hệ thống vừa chạm vào mặt trượt để đảm bảo khi $S = 0$ thì $\dot{S} = 0$. Trong thực tế, điều này không thể thực hiện được do thời gian trễ hay quán tính của khâu chấp hành. Kết quả là quỹ đạo pha tiếp tục vượt qua khỏi mặt trượt sau khi chạm vào nó và gây nên hiện tượng quỹ đạo pha dao động quanh mặt trượt (hiện tượng chattering) như Hình 10.



Hình 10 Quỹ đạo thực tế của hệ trượt

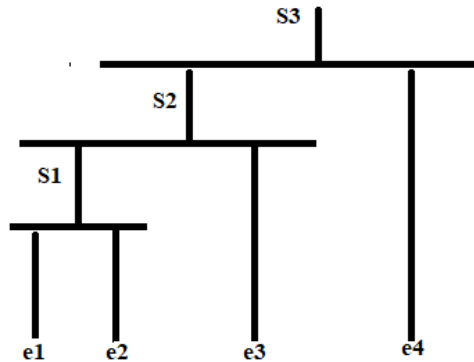
Hiện tượng chattering là không mong muốn vì nó gây ra sai số điều khiển, làm phát nóng mạch điện tử, mài mòn các bộ phận cơ khí, kích động các mode tần số cao không được mô hình hóa, làm giảm chất lượng điều khiển hoặc mất ổn định.

Người ta luôn tìm nhiều biện pháp khác nhau để làm giảm thiểu hoặc loại trừ hiện tượng này. Một số cách để khắc phục hiện tượng chattering là sử dụng hàm

saturation, hàm sigmoid hoặc hàm sat-PI thay thế cho hàm sign; dùng logic mờ (*fuzzy logic*) để giảm hiện tượng chattering.

3.1.2 Thiết kế bộ điều khiển trượt cho hệ Cart and Pole

Dựa vào tài liệu [12], các mặt trượt được định nghĩa như sau:



Hình 11 Cấu trúc của các mặt trượt

Điều khiển trượt bám quỹ đạo hệ xe con lắc ngược, ta cần đặt ra các trạng thái mẫu cho hệ thống: $X_d = [x_{1d} \ x_{2d} \ x_{3d} \ x_{4d}]^T$.

Sai số giữa trạng thái thực và trạng thái mẫu của hệ là:

$$e_i, (i = 1, 2, 3) = X_i - X_d$$

Dựa vào **Error! Reference source not found.**, các mặt trượt sẽ lần lượt được đặt như sau:

$$s_1 = c_1 e_1 + c_2 e_2 \quad (67)$$

$$s_2 = s_1 + c_3 e_3 \quad (68)$$

$$s_3 = s_2 + c_4 e_4 \quad (69)$$

Với c_1, c_2, c_3, c_4 là các hằng số của giải thuật trượt thoả tiêu chuẩn Hurwitz (sinh viên tìm bằng giải thuật di truyền – General Algorithm).

Đạo hàm các mặt trượt theo thời gian:

$$\begin{aligned}\dot{s}_1 &= c_1 \dot{e}_1 + c_2 \dot{e}_2 \\ &= c_1 (\dot{x}_1 - \dot{x}_{1d}) + c_2 (f_1(x) + g_1(x)u_1 - \dot{x}_{2d})\end{aligned}\quad (70)$$

$$\begin{aligned}\dot{s}_2 &= \dot{s}_1 + c_3 \dot{e}_3 \\ &= c_1 (\dot{x}_1 - \dot{x}_{1d}) + c_2 (f_1(x) + g_1(x)u_2 - \dot{x}_{2d}) + c_3 (x_4 - \dot{x}_{3d})\end{aligned}\quad (71)$$

$$\begin{aligned}\dot{s}_3 &= \dot{s}_2 + c_4 \dot{e}_4 \\ &= c_1 (\dot{x}_1 - \dot{x}_{1d}) + c_2 (f_1(x) + g_1(x)u_3 - \dot{x}_{2d}) + \\ &\quad c_3 (x_4 - \dot{x}_{3d}) + c_4 (f_2(x) + g_2(x)u_3 - \dot{x}_{4d})\end{aligned}\quad (72)$$

Yêu cầu đặt ra của bài toán bám quỹ đạo cho hệ con lắc ngược trên xe là chiếc xe sẽ di chuyển theo quỹ đạo cho trước trong khi giữ cho con lắc luôn ở vị trí cân bằng $x_{1d} = x_{2d} = \dot{x}_{1d} = 0$.

Điện áp điều khiển cho từng mặt trượt được định nghĩa như sau

$$u_i = u_{eq(i)} + u_{sw(i)} \quad (73)$$

Với $u_{eq(i)}$ là điện áp giữ các trạng thái trên mặt trượt tương ứng.

$u_{sw(i)}$ là điện áp kéo các trạng thái về mặt trượt tương ứng.

$i = 1, 2, 3$.

Cho $\dot{s}_i = 0$, từ [12], ta thu được:

$$u_{eq1} = \frac{-(c_1 x_2 + c_2 f_1(x))}{c_2 g_1(x)} \quad (74)$$

$$u_{eq2} = \frac{-(c_1 (x_2 - \dot{x}_{1d}) + c_2 f_1(x) + c_3 (x_4 - \dot{x}_{3d}))}{c_2 g_1(x)} \quad (75)$$

$$u_{eq3} = \frac{-(c_1 x_2 + c_2 f_1(x) + c_3 (x_4 - \dot{x}_{3d}) + c_4 (f_2(x) - \dot{x}_{4d}))}{c_2 g_1(x) + c_4 g_2(x)} \quad (76)$$

Tín hiệu điều khiển u_i trong [12] được tìm theo nguyên lý ổn định Lyapunov.

Định nghĩa hàm Lyapunov. Nhìn vào các phương trình (74) đến (76), ta nhận thấy rằng tín hiệu điện áp ở phương trình (76) sẽ điều khiển bao gồm tất cả các trạng thái của hệ thống (bao quát hơn u_{eq2} và u_{eq3}), nên hàm Lyapunov sinh viên chọn sẽ liên quan đến mặt trượt s_3 :

$$V = \frac{1}{2} s_3^2 \quad (77)$$

Đạo hàm V theo thời gian, ta được:

$$\begin{aligned} \dot{V} &= s_3 \dot{s}_3 \\ \dot{s}_3 &= c_1 \dot{e}_1 + c_2 \dot{e}_2 + c_3 \dot{e}_3 + c_4 \dot{e}_4 \\ &= c_1 x_2 + c_2 \left(f_1(x) + g_1(x)(u_{sw3} + u_{eq3}) \right) \\ &\quad + c_3 (x_4 - x_{4d}) + c_4 \left(\left(f_2(x) + g_2(x)(u_{sw3} + u_{eq3}) \right) - \dot{x}_{4d} \right) \\ &= c_1 (x_2 - \dot{x}_{1d}) + c_2 f_1(x) + c_3 (x_4 - x_{4d}) + c_4 (f_2(x) - \dot{x}_{4d}) \\ &\quad + (c_2 g_1(x) + c_4 g_2(x)) u_{eq3} + (c_2 g_1(x) + c_4 g_2(x)) u_{sw3} \end{aligned} \quad (78)$$

Thay thế u_{eq3} trong (76) vào (78)

$\dot{s}_3 = (c_2 g_1(x) + c_4 g_2(x)) u_{sw3}$	(79)
---	------

Theo nguyên lý ổn định của Lyapunov, do V là hàm xác định dương, nên \dot{V} bắt buộc phải là hàm xác định âm để thỏa điều kiện ổn định của tiêu chuẩn. Do đó, ta đặt:

$\dot{s}_3 = -k s_3 - \eta \text{sign}(s_3)$	(80)
--	------

Với k, η là những hằng số dương (Tuy nhiên, trong đề tài này, để giảm thiểu chattering, sinh viên sẽ tận dụng giá trị của nuy thay đổi theo thời gian). Kết quả là:

$\begin{aligned} \dot{V}(t) &= s_3 (-k s_3 - \eta \text{sign}(s_3)) \\ &= -k s_3^2 - \eta s_3 \leq 0 \end{aligned}$	(81)
---	------

thỏa mãn nguyên lý ổn định của Lyapunov. Khi ấy, $u_{sw3} = \frac{-k s_3 - \eta \text{sign}(s_3)}{c_2 g_1(x) + c_4 g_2(x)}$

Có thể thấy, $u_3 = u_{eq3} + u_{sw3}$ đảm bảo cho mặt trượt s_3 tiến về không và ổn định khi $t \rightarrow \infty$. Khi ấy các mặt trượt s_2 và s_1 cũng tiến về không và ổn định. Có nghĩa là, các sai số quỹ đạo $e_i (i = 1..4)$ cũng tiến về không.

Phân tích tính ổn định

Định lý: Xem xét một hệ thống bất ổn định under –actuated system với những sliding surface được định nghĩa như (67), (68), (69). Tất cả các thông số của mặt trượt đều là constant và giả sử các $f_i(X)$, $b_i(X)$ và X là sát với thực tế. Lúc này, tất cả các trạng thái của hệ thống sẽ hội tụ về trạng thái cân bằng (sai số $e \rightarrow 0$ và vận tốc $\dot{e} \rightarrow 0$) và ổn định tiệm cận.

Chứng minh: Ta đạo hàm hàm Lyapunov V theo thời gian t trong phương trình (77), ta thu được:

$\begin{aligned}\dot{V} &= s_3 \cdot \dot{s}_3 \\ &= s_3 \cdot (-k \cdot s_3 - \eta \cdot \text{sgn}(s_3)) \\ &= -k_1 s_3^2 - \eta s_3 \leq 0 \quad \forall s_3\end{aligned}$	(82)
---	------

Từ phương trình (82) nguyên hàm 2 vế như sau:

$\begin{aligned}\int_0^t \dot{V} dt &= \int_0^t (-k \cdot s_3^2 - \eta \cdot s_3) dt \\ \Leftrightarrow V(0) &= V(t) + \int_0^t (k \cdot s_3^2 + \eta \cdot s_3) dt \Rightarrow \int_0^t (k \cdot s_3^2 + \eta \cdot s_3) dt\end{aligned}$	(83)
--	------

Do đó:

$\lim_{t \rightarrow \infty} \int_0^t (k \cdot s_3^2 + \eta \cdot s_3) dt \leq V(0) < \infty$	(84)
---	------

Từ phương trình (83), (84), áp dụng bổ đề Barbalat, tồn tại:

$\lim_{t \rightarrow \infty} (k \cdot s_3^2 + \eta \cdot s_3) = 0$	(85)
--	------

Phương trình trên cũng cho biết mặt trượt s_3 có $\lim_{t \rightarrow \infty} s_3 = 0$. Nói cách khác mặt trượt s_3 sẽ ổn định tiệm cận, do đó các biến trạng thái sẽ hội tụ về 0 khi $t \rightarrow \infty$.

Do đó, sinh viên chọn tín hiệu điều khiển cho hệ con lắc ngược trên xe là:

$u = u_{eq} + u_{sw}$	(86)
-----------------------	------

Với:

$$U_{eq} = \frac{-1}{c_4 g_2 + c_2 g_1} (c_4 f_2 - c_4 \dot{x}_{4d} + c_3 e_4 + c_2 f_1 - c_2 \dot{x}_{2d} + c_1 e_2) \quad (87)$$

$$U_{sw} = \frac{-(\eta \text{sign}(s_3) + k s_3)}{c_4 g_2 + c_2 g_1} \quad (88)$$

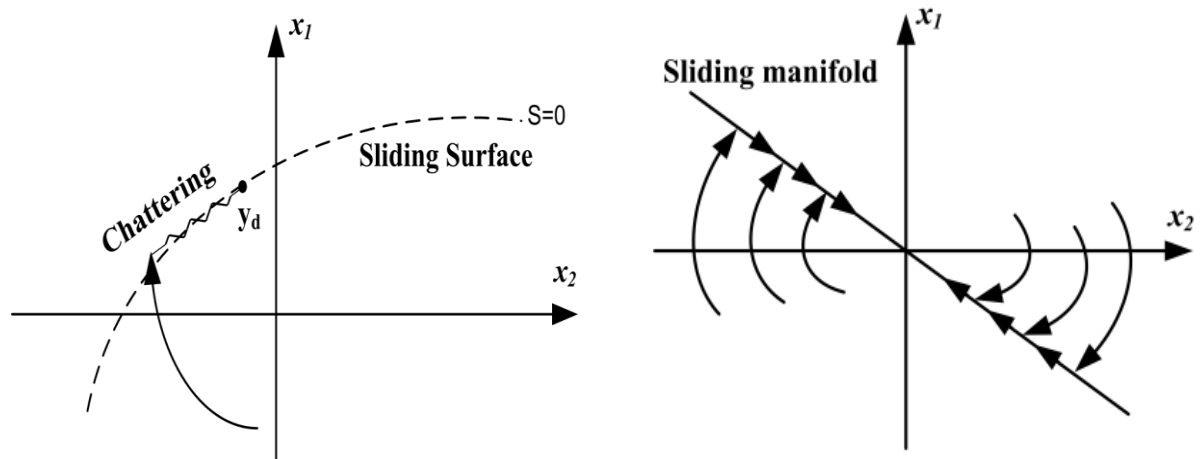
Trong luận văn này, bộ thông số giải thuật trượt [17] được chọn thông qua phương pháp Genetic Algorithm như sau:

$c1 = 61.8834$, $c2 = 76.5393$, $c3 = -82.9995$, $c4 = -30.4564$, $k = 45$. Thông số nuy η được tìm thông qua phương pháp Linear Regression được trình bày mục 3.3.

3.2 Chattering

Hiện tượng Chattering là hiện tượng dườg như luôn có trong tất cả các phương pháp điều khiển trượt [4'] [5'] [6']. Vì Sliding Mode Control có u_{sw} có tác dụng đưa các trạng thái đang hướng ở vị trí nào thì sẽ cung cấp một điện áp với chiều điều khiển ngược lại với điện áp vô cùng lớn qua đó làm các trạng thái dao động xung quanh vị trí cân bằng. Điều đó vô tình cũng gây nên hiện tượng Chattering. Đây là hiện tượng có hại trong quá trình điều khiển, vì nó gây nên tổn thất nhiệt cao trong các động cơ, làm tăng nhanh sự hao mòn trong các bộ phận cơ khí di chuyển,...

Do đó, sinh viên sinh đề xuất một phương pháp để giảm thiểu hiện tượng Chattering bằng các sử dụng giải thuật Linear Regression.



Hình 12 [14] Hiện tượng Chattering

3.3 Linear Regression

Như đã đề cập ở trên, hiện tượng chattering phụ thuộc vào giá trị của nuy. Giá trị này giúp mặt trượt trượt về giá trị zero. Nếu giá trị này nhỏ, hiện tượng chattering sẽ nhỏ và ngược lại. Dựa vào mối liên hệ đó giữa nuy và chattering, sinh viên đề xuất giải thuật Linear Regression cùng với kinh nghiệm thực tế để xây dựng phương trình tìm nuy theo giá trị mặt trượt. Khi trượt càng lớn nuy càng lớn và ngược lại, điều này sẽ giúp hệ trượt về giá trị zero nhanh mà cũng đảm bảo giảm thiểu hiện tượng Chattering này.

Linear Regression là giải thuật cơ bản trong Machine Learning, thuộc nhánh học có giám sát. Phương pháp này thường được sử dụng trong các vấn đề như dự đoán giá trị output với một input mới dựa vào các thông tin input đã có đưa vào và output của những thông tin đó. Mối liên hệ giữa input và output được diễn tả bởi phương trình tuyến tính sau:

$y \approx \hat{y} = w_0 x_0 + w_1 x_1 + w_2 x_2 + \dots + w_N x_N = X^T w$	(89)
---	------

Trong đó: w_0 là bias và x_0 là vector đơn vị (có cùng chiều dài với X). Input $X = [x_1, x_2, x_3, \dots, x_N]$ được gọi là vector đặc trưng của hệ thống với mỗi vector thành phần $x_i (i \in N)$ cho giá trị của từng đặc trưng. Vector output $y = [y_1, y_2, y_3, \dots, y_N]$ cho

ra kết quả theo các vector đặc trưng. Và cuối cùng, $w = [w_1, w_2, w_3, \dots, w_N]$ gọi là các weight vector và đây là các giá trị cần tìm và là chìa khoá trong giải thuật này.

Hàm mất mát: $L(w) = \frac{1}{2N} \sum_{i=1}^N (y - \hat{y})^2 = \frac{1}{2N} \sum_{i=1}^N (y - X^T w)^2$ đưa ra sự sai số giữa giá trị y thực tế và giá trị \hat{y} dự đoán. Mục đích của giải thuật Linear Regression là tìm w sao cho hàm mất mát có giá trị càng bé càng tốt $w = \underset{w}{\operatorname{argmin}} L(w)$, khi đó y và \hat{y} sẽ gần bằng nhau. Vì hàm mất mát có dạng lồi và có đạo hàm tại mọi w nên giải pháp tối ưu cho nó là đạo hàm bằng không [15].

$\frac{\nabla L(w)}{\nabla w} = \frac{1}{N} (-XX^T w + X y) = 0$	(90)
$\rightarrow w = (XX^T)^{-1} X y$	(91)

Nếu thành phần vector trọng số w được tìm tốt, khi đó giá trị dự đoán sẽ rất gần với giá trị thật, trong đề tài luận văn này, giá trị đó là nuy.

Trong giới hạn đề tài luận văn này, lần lượt các thông số input/ output mẫu X và y cho phương trình (90), (91) như sau:

$y = [9, 8, 7, 6, 5, 4, 3, 2, 1, 0]$
 $x = [30, 15, 8, 7, 6, 5, 4, 3, 2, 0]$

3.4. Giải thuật di truyền

Giới thiệu

GA được Holland đưa ra năm 1975 là giải thuật tìm kiếm lời giải tối ưu trên nguyên tắc phỏng theo quá trình tiến hóa và quy luật di truyền của sinh vật trong tự nhiên. Bản chất toán học của GA là giải thuật tìm kiếm theo xác suất.

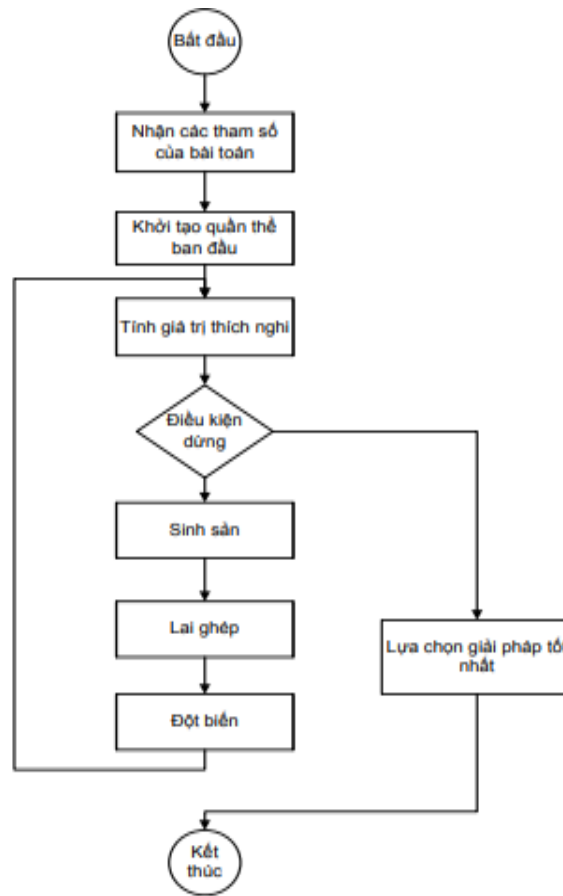
GA cũng như các giải thuật tiến hóa nói chung dựa trên quan điểm cho rằng, quá trình tiến hóa của tự nhiên là quá trình là quá trình hoàn hảo nhất, hợp lý nhất và tự nó đã mang tính tối ưu. Quan niệm này có thể xem như là một tiên đề đúng không chứng minh được nhưng phù hợp với thực tế khách quan. Quá trình tiến

hóa thể hiện tính tối ưu ở chỗ, thế hệ sau bao giờ cũng tốt hơn (phát triển hơn, hoàn thiện hơn) thế hệ trước. Tiến hóa tự nhiên được duy trì nhờ hai quá trình cơ bản: sinh sản và chọn lọc tự nhiên, các thế hệ mới luôn được sản sinh ra để bổ sung, thay thế thế hệ cũ. Cá thể nào phát triển hơn, thích ứng hơn với môi trường sẽ có nhiều khả năng tồn tại và phát triển, cá thể nào không thích ứng được với môi trường sẽ bị đào thải. Sự thay đổi của môi trường là động lực của quá trình tiến hóa. Ngược lại, tiến hóa cũng tác động trở lại góp phần làm thay đổi môi trường.

Các cá thể mới sinh ra trong quá trình tiến hóa nhờ sự lai ghép ở thế hệ cha-mẹ. Một cá thể mới có thể mang những tính trạng của cha-mẹ (di truyền), cũng có thể mang những tính trạng hoàn toàn mới (đột biến). Di truyền và đột biến là hai cơ chế có vai trò quan trọng như nhau trong tiến trình tiến hóa, dù rằng hiện tượng đột biến diễn ra với xác suất nhỏ hơn nhiều so với di truyền. GA nói riêng và các giải thuật tiến hóa nói chung tuy có điểm khác biệt nhưng đều mô phỏng ba quá trình cơ bản của tiến hóa tự nhiên: chọn lọc tự nhiên, lai ghép (sinh sản) và đột biến.

GA là phương pháp tối ưu hóa ngẫu nhiên tổng quát dùng để giải các bài toán tìm kiếm. GA có một số khác biệt so với giải thuật tối ưu hóa và tìm kiếm thông thường. Thứ nhất, GA làm việc với quần thể gồm nhiều NST (tập hợp nhiều lời giải), tìm kiếm nhiều điểm cực trị cùng một lúc. Bằng các phép toán di truyền, GA trao đổi thông tin giữa những đỉnh cực trị, do đó hạn chế khả năng rơi vào cực trị cục bộ. Nói cách khác, GA là phương pháp tìm kiếm lời giải tối ưu toàn cục. Thứ hai, GA làm việc với chuỗi ký hiệu (chuỗi NST) chứ không làm việc với các thông số. Thứ ba, GA chỉ cần đánh giá hàm mục tiêu để định hướng quá trình tìm kiếm mà không cần bất cứ thông tin gì khác (như đạo hàm...). Do các đặc điểm trên, GA có khả năng giải quyết rất hiệu quả các bài toán tìm cực trị hàm phi tuyến, không khả vi, không liên tục...

➤ Lưu đồ giải thuật



Hình 13 Lưu đồ giải thuật di truyền

Lưu đồ GA dùng để giải bài toán tối ưu trong điều khiển được trình bày ở [16]. Để áp dụng GA giải bài toán tối ưu trước hết phải **mã hóa** (encode) lời giải của bài toán thành chuỗi NST có thể là chuỗi số nhị phân, chuỗi số thập phân, chuỗi số tự nhiên hay chuỗi số thực. Mỗi NST đại diện cho một cá thể trong quần thể. Để đánh giá các cá thể phải định nghĩa một **hàm thích nghi** (fitness function), thường là hàm cần tìm cực trị hoặc một biến đổi tương đương của hàm cần tìm cực trị. Thế hệ đầu tiên gồm nhiều cá thể (lời giải) được khởi động ngẫu nhiên. Qua quá trình chọn lọc tự nhiên, những cá thể thích nghi nhất với môi trường sống (có độ thích nghi cao) mới tồn tại và có cơ hội sinh sản để tạo ra thế hệ con có xu hướng thích nghi với môi trường sống tốt hơn thế hệ bố mẹ. Cá thể nào có độ thích nghi càng cao thì càng có nhiều cơ hội để tồn tại và bắt cặp với một cá thể khác để sinh ra thế hệ con. Nhờ quá trình lai ghép mà các đặc tính tốt của thế hệ

trước được truyền lại cho thế hệ sau. Đôi khi do lỗi trong quá trình di truyền trong tự nhiên xảy ra hiện tượng đột biến với xác suất rất thấp. Hiện tượng đột biến có thể tạo ra những cá thể có độ thích nghi rất tốt, có độ thích nghi vượt trội hơn hẳn so với các cá thể còn lại trong quần thể. Chọn lọc tự nhiên sẽ loại bỏ những cá thể kém thích nghi, kết quả là qua quá trình tiến hóa được lặp lại từ thế hệ này sang thế hệ khác, thế hệ sau có xu hướng thích nghi với môi trường sống tốt hơn thế hệ trước, điều đó có nghĩa là lời giải của bài toán tiến dần đến lời giải tối ưu.

Chương trình viết trong Matlab, sinh viên tham khảo tài liệu [17]:

```
% Programmed by      : Huynh Thai Hoang, University of Technology at Ho Chi
Minh City.
% Last updated       : November 25, 2005

clc;%xoa bo man hinh
clear;%xoa bo toan bo du lieu truoc do
rand('state',sum(100*clock));
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
thong_so
Jmin = 1000000000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

max_generation=20000; %so the he toi da trong qua trinh chay(chay toi toi
da 200 the he thi dung lai)
max_stall_generation=50000;
epsilon=0.001; %J chuan(neu the he nao co J<=epsilon tuc la da tim duoc
thog so thoa man, GA khong can chay tiep nua)
pop_size=500; % so ca the trong quan the(so cha me)
npar = 5; %co 6 nhien sac the trong 1 ca the(6 nhien sac the nay lan luot
la Kp1,Ki1, Kd1, Kp2,Ki2,Kd2
range=[ -1  -1  -1  -1  0;...
        1   1   1   1  1].*100; %tam cua cac NST
dec=[2 2 2 2 2].*3/2; %vi tri dau cham thap phan
sig=[6 6 6 6 6]; %so chu so co nghia trong moi nhien sac the

cross_prob = 0.8; %he so lai ghep
mutate_prob = 0.2; %he so dot bien(he so dot bien + he so lai ghep =1)
elitism = 1; %luon giu lai gia tri tot nhat

rho=0.02; %trong so quyet dinh e1,e2 va u cai nao quan trong voi J hon

par=Init(pop_size,npar,range); %khoei tao 20 ca the cha me dau tien

Terminal=0; %KHOI DONG
generation = 0; %CAC GIA TRI
stall_generation=0;%DAU TIEN TRUOC KHI CHAY ga

for pop_index=1:pop_size
%Khoei tao
    c1 = par(pop_index,1);
%cac the he
```

```

        c2 = par(pop_index,2);
%cha me
        c3 = par(pop_index,3);
%dau tien
        c4 = par(pop_index,4);
%(cho chay GA truoc
        k = par(pop_index,5);
        %k = 150;
        sim('SMC_trackking.slx');
%(tuc 1 cho lai ghep thu truoc khi
        if length(e1)>9500 %lai ghep thiet)
            J=e1'*e1+e3'*e3;
            if J < Jmin
                Jmin = J;
                c1
                c2
                c3
                c4
                k
                J
            end
            %J=e4'*e4
            fitness(pop_index)=1/(J+eps);
%
        else
%
            J=10^100;
%
            fitness(pop_index)=1/(J+eps);
%
        end
%
end
%
[bestfit,bestchrom]=max(fitness);
%

c10=par(bestchrom,1); %cac
c20=par(bestchrom,2); %nhien sac the
c30=par(bestchrom,3); %cua cha me
c40=par(bestchrom,4); %toi uu nhat

J0=1/bestfit+0.001; %do elitism = nen doi
hoi phai co 1 cha me tot nhat de so sanh voi cac he he sau, tim toi uu
nhat)

while ~Terminal %Terminal se bang 1 neu chay du 200 the he hoac trong qua
trinh chay co 1 the he con cai co bestfit thoa man sai so <epsilon
    generation = generation+1;
%truoc moi lan chay cho hien generation tuong ung voi lan chay thu
1,2,3,...(tuong ung generation 1,2,3... hay the he thu 1,2,3... )
    disp(['generation #' num2str(generation) ' of maximum '
num2str(max_generation)]); %
    pop=Encode_Decimal_Unsigned(par,sig,dec);
%ma hoa thap phan (nhien sac the cua cac ca the)
    parent=Select_Linear_Ranking(pop,fitness,0.2,elitism,bestchrom);
%sap hang cha me tuyen tinh(cha me tot thi xac suat co con se cao hon, cha
me yeu ot thico it con hoac ko co con)
    child=Cross_Twopoint(parent,cross_prob,elitism,bestchrom);
%con cai se duoc sinh ra dua vao lai ghep da diem(trong truong hop nay la
hai diem)

```

```

    pop=Mutate_Uniform(child,mutate_prob,elitism,bestchrom);
%dot bien theo dang phan bo deu
    par=Decode_Decimal_Unsigned(pop,sig,dec);
%giai ma ket qua ve lai dang nhien sac the(la so duong)

    for pop_index=1:pop_size %lan
        luot test tung ca the trong quan the
            c1 = par(pop_index,1);
        %cac the he
            c2 = par(pop_index,2);
        %cha me
            c3 = par(pop_index,3);
        %dau tien
            c4 = par(pop_index,4);
            k = par(pop_index,5);
            %k = 150; %Kp2,Ki2, Kd2
        %
        sim('SMC_tracking.slx'); %tien hanh chay mo
        phong de kiem tra xem ca the nao toi uu
        % J=(e1'*e1)+(e2'*e2);%+rho*(u'*u);
        % fitness(pop_index)=1/(J+eps);
        if length(e1)>9500
        %neu mo phong chay duoc du 10000 mau(100s) thi

            J=e1'*e1+e3'*e3;
            if J < Jmin
                Jmin = J;
                c1
                c2
                c3
                c4
                k
                J
            end
            %+rho*(u'*u); %tih J dua vao mau thu 8000 den
        9500(giay thu 80
            %J=e4'*e4
            fitness(pop_index)=1/(J+eps);
        %den giay thu 95)
        else
        %neu mo phong chay chua den duoc giay thu 95 ma da dung
            J=10^100;
        %nghia la gia tri da dao dong dem mat on dinh
            fitness(pop_index)=1/(J+eps);
        %tuc Kp1, Ki1, Kd1, Kp2, Ki2, Kd2 ko tot, khog can tinh toan tiep
        end
        %luc nay cho J la 1 gia tri rat lon(-->bestfit qua nho nen ko duoc chon lam
        gia tri toi uu)
        end
        [bestfit(generation),bestchrom]=max(fitness);
        %ca the nao co Kp1, Ki1, Kd1, Kp2, Ki2, Kd2 lam cho J nho nhat, tuc best
        fit lon nhat

        %se duoc chon am ca the toi uu sau khi chay hoan thanh GA, tuc la ca the
        bestchrom

        if generation == max_generation %neu
        chay du 200 the he rui ma chua co ca the thoa J <=epsilon (o dong thu 10)
            Terminal = 1; %thi
        cho Terminal=1, stop chuong trinh GA, chon ca the tot nhat trong cac truong
        hop do lam bestchrom
    end
end

```

```

elseif generation>1
    if abs(bestfit(generation)-bestfit(generation-1))<epsilon
        stall_generation=stall_generation+1;
        if stall_generation == max_stall_generation, Terminal = 1;end
%con trong qua trình chạy mà xuất hiện cả thể thỏa  $J \leq \epsilon$  (đọc thu 10)
thì cho Terminal =1, dùng GA và lấy giá trị do làm bestchrom
    else
        stall_generation=0;
    end
end

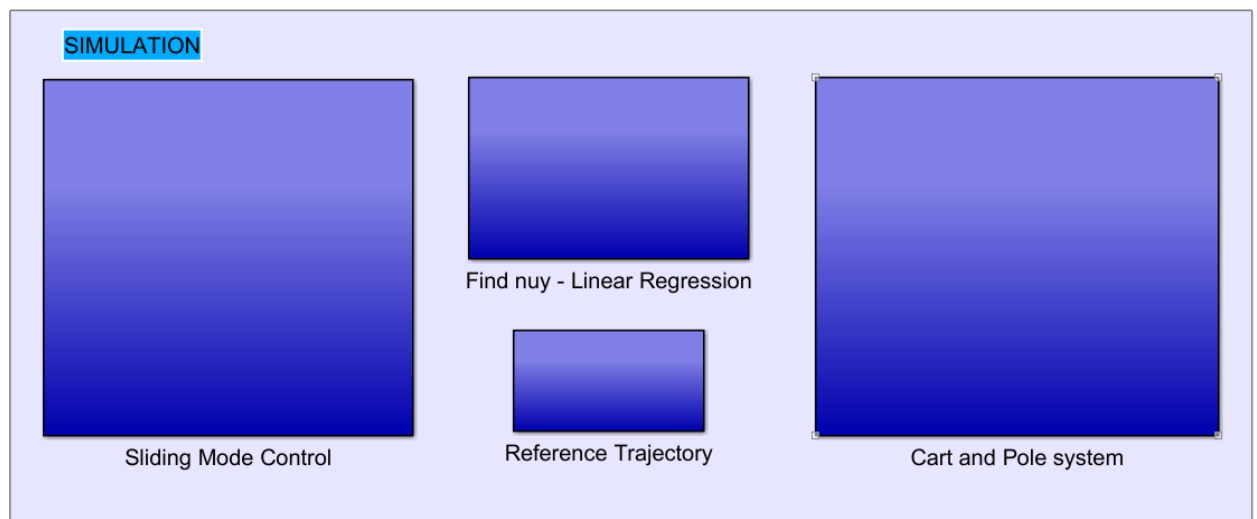
end%While

plot(1./bestfit)

```

3.5 Mô phỏng hệ thống trên Matlab/ Simulink

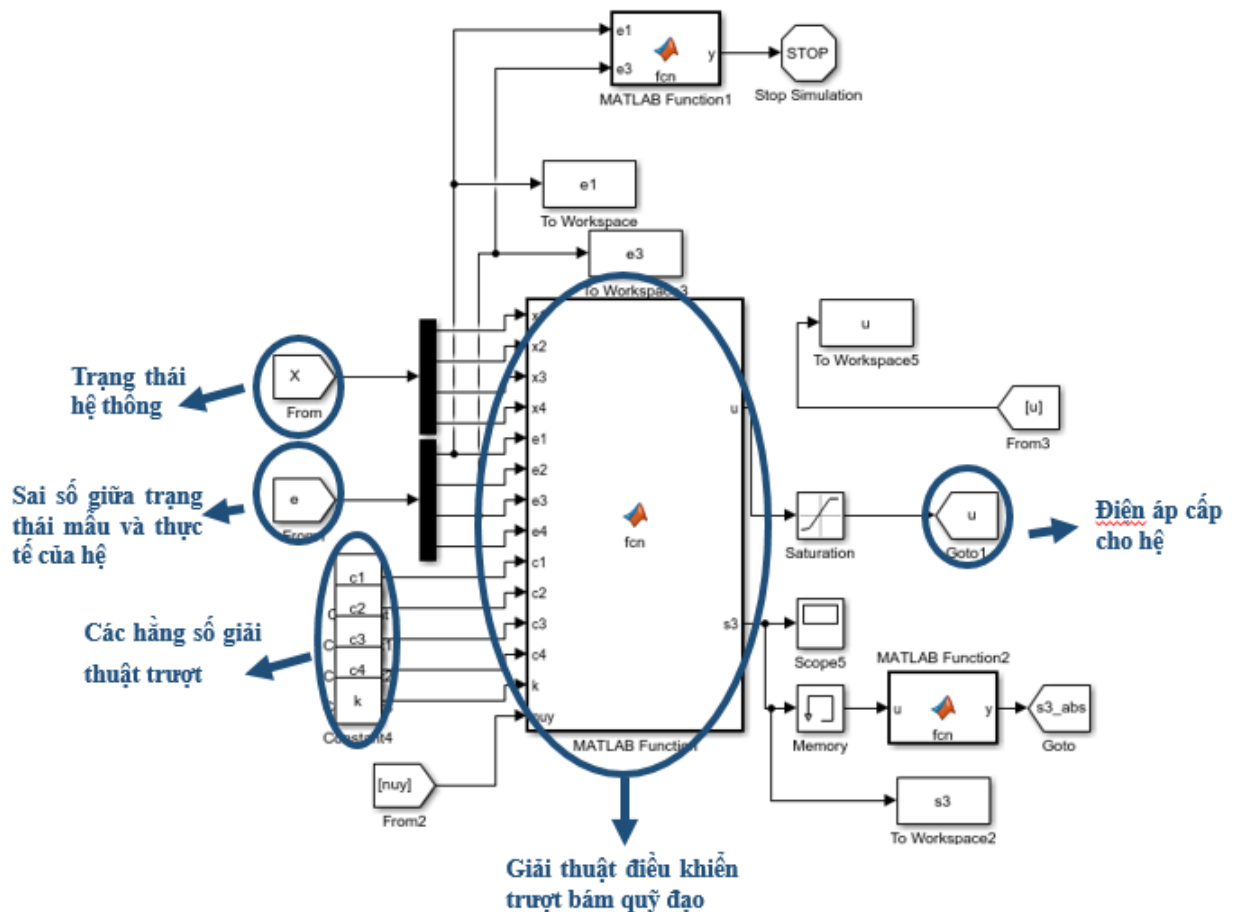
Dựa trên những tính toán cũng như phương pháp lý thuyết được nêu ở các phần trên. Sinh viên tiến hành thực hiện lập trình mô phỏng hệ thống thông qua Matlab/ Simulink để có được cái nhìn trực quan về vận hành hệ thống này cũng như tính khả thi khi thực hiện trên hệ thống thực.



Hình 14 Mô phỏng hệ thống Cart and Pole dùng ISMC - LR

Hình 14 là tổng thể chương trình mô phỏng hệ thống xe con lắc ngược bám quỹ đạo sử dụng giải thuật điều khiển trượt tổng hợp cùng với Linear Regression. Chương trình sẽ được chia ra thành 4 function (4 khối trong lập trình Matlab/ Simulink). Bao gồm: Sliding Mode Control, Find nuy – Linear Regression, Reference Trajectory, Cart and Pole System.

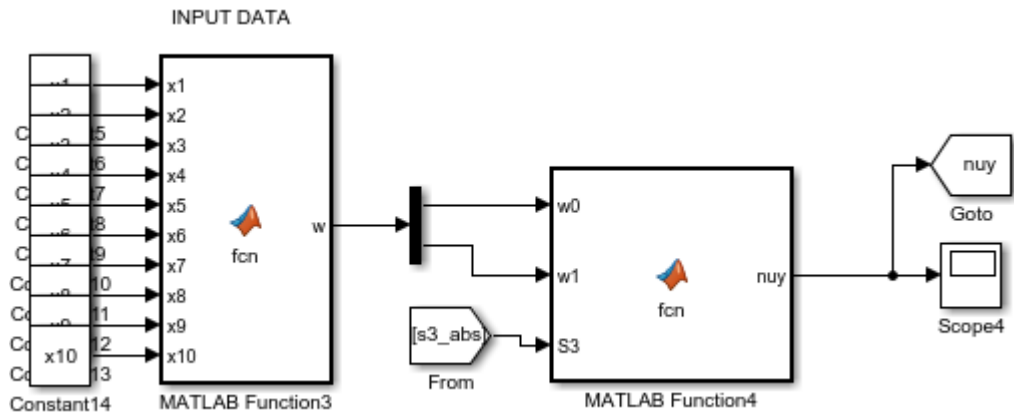
3.4.1 Simulation – Sliding Mode Control



Hình 15 Bên trong khối Sliding Mode Control

Trong khối này, sinh viên viết chương trình xử lý các thông tin input đầu vào bao gồm các trạng thái của hệ thống, reference trajectory của nó, các hằng số giải thuật trượt, k và nuy (tìm bằng Linear Regression). Đưa các thông tin đó vào khối chương trình và dùng giải thuật Sliding Mode Control để xử lý và tìm giá trị điện áp u để điều khiển hệ thống. Đồng thời return về giá trị mặt trượt để dùng làm input cho khối function tìm nuy .

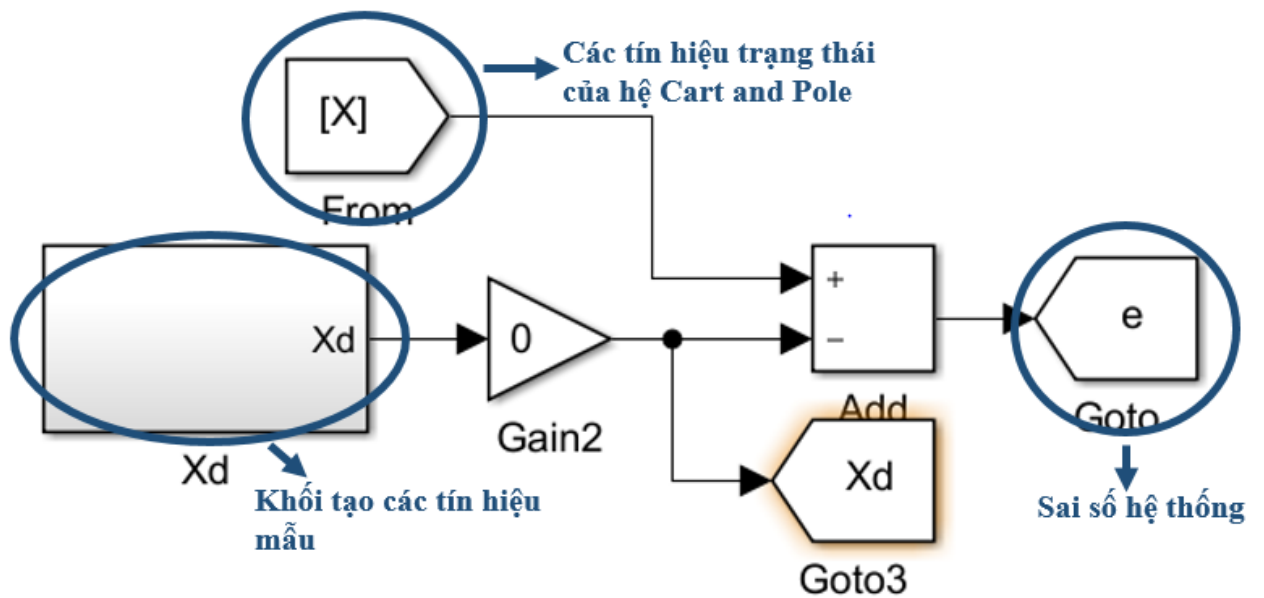
3.4.2 Simulation – Find nuy – Linear Regression



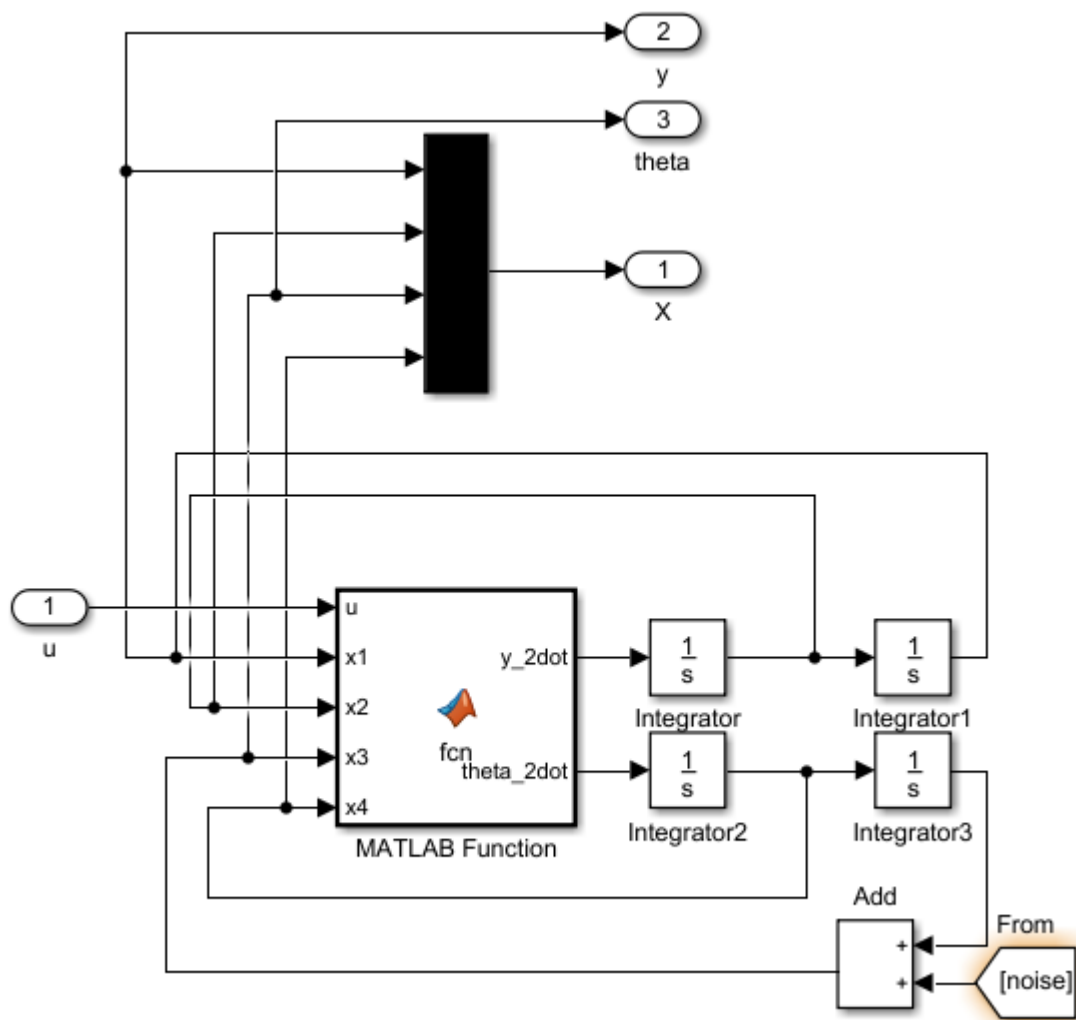
Hình 16 Bên trong khối Find nuy – Linear Regression

Trong khối này, sinh viên thực hiện chương trình tìm nuy dựa vào lý thuyết Linear Regression. Từ x_1 đến x_{10} là các input giá trị mẫu đầu vào để cho giải thuật. Từ đó tìm ra phương trình sao cho nuy dự đoán gần với nuy thực tế nhất.

3.4.3 Simulation – Reference Trajectory



Hình 17 Bên trong khối Reference Trajectory



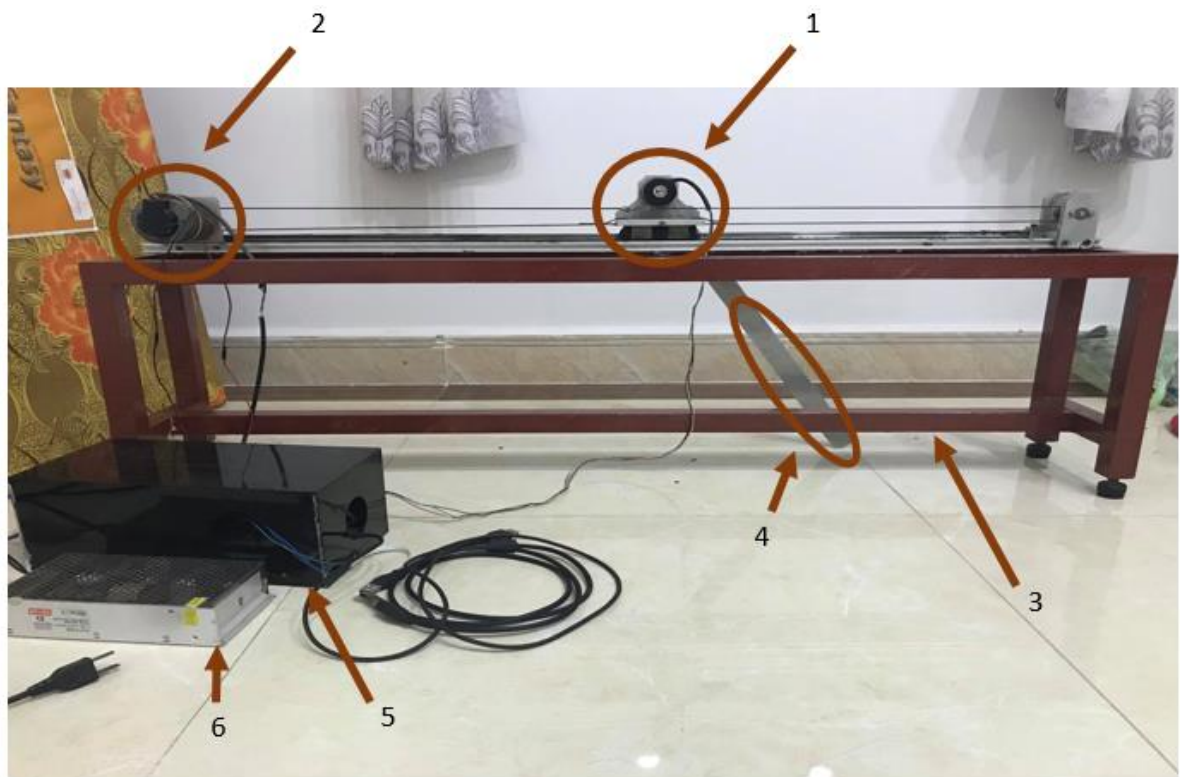
Hình 20 Bên trong khối Cart and Pole system – Cart and Pole system

Function này sẽ mô phỏng lại hệ thống thực của Cart and Pole ngoài thực tế, dựa vào các phương trình được xây dựng từ (1) đến (41) trong chương 2. Tín hiệu u sẽ đưa vào các chương trình và kết quả return lại là trạng thái của hệ, bao gồm: góc con lắc, vận tốc góc con lắc, vị trí xe và vận tốc của chiếc xe.

Chương 4: Thiết kế hệ thống Cart and Pole trên thực tế

4.1 Thiết kế cơ khí của mô hình thực nghiệm hệ con lắc ngược trên xe

Hình 21 là mô hình phần cứng ngoài thực tế được sinh viên sử dụng trong đề tài luận văn tốt nghiệp.



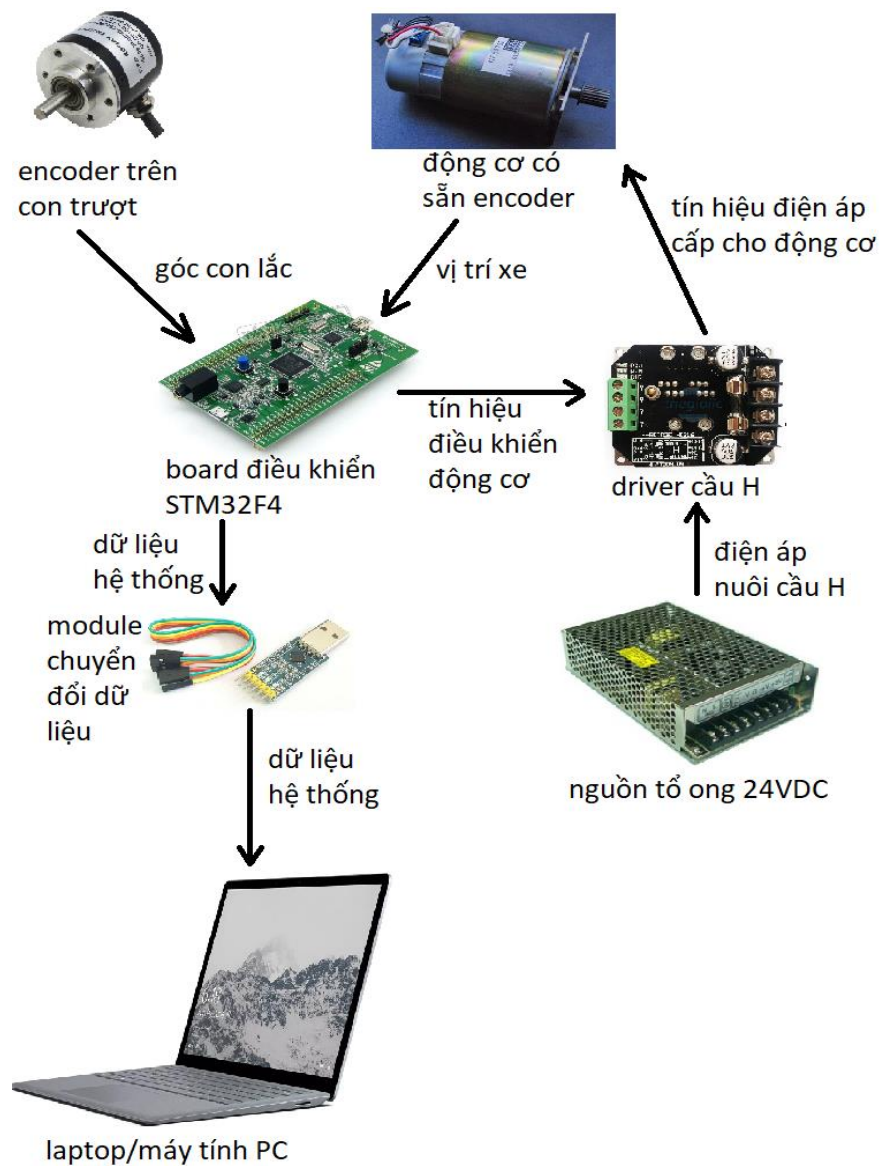
Hình 21. Hệ thống Cart and Pole thực tế

Cấu trúc của hệ thống thực tế Cart and Pole bao gồm các phần như sau:

- 1: Một con trượt gất trên base và được kéo qua lại thông qua motor đặt đầu base, trung gian qua dây curoa. Sự ma sát được sinh viên giảm thiểu bằng cách bôi mỡ bôi trơn đoạn trượt của xe. Trên thanh trượt có gắn một Encoder 600 xung để tính toán góc lệch của con lắc so với phương thẳng đứng hướng lên.
- 2: Motor điều khiển hệ thống, trên trục động cơ có gắn Encoder 600 xung để tính toán đo độ dịch chuyển của con trượt (Cart).

- 3: Base là một khung sắt 4 chân. Cấu tạo như khung để bàn thấp thường thấy ở quê. Khung nhẹ và tiện lợi cho việc di chuyển nhưng cũng đảm bảo được sự chắc chắn khi hệ thống hoạt động.
- 4: Trên trục encoder có gắn 1 thanh con lắc đồng chất bằng nhôm (kí hiệu 3). Thanh con lắc không quá nhẹ để có thể đủ tạo sự mất cân bằng, thể hiện được giải thuật điều khiển trên phần cứng. Tuy nhiên, thanh con lắc cũng không quá nặng để có thể bảo vệ trục encoder trên con trượt không bị gãy.
- 5: Hộp bảo vệ vi xử lý và cầu H.
- 6: Nguồn 24VDC cấp cho hệ thống.

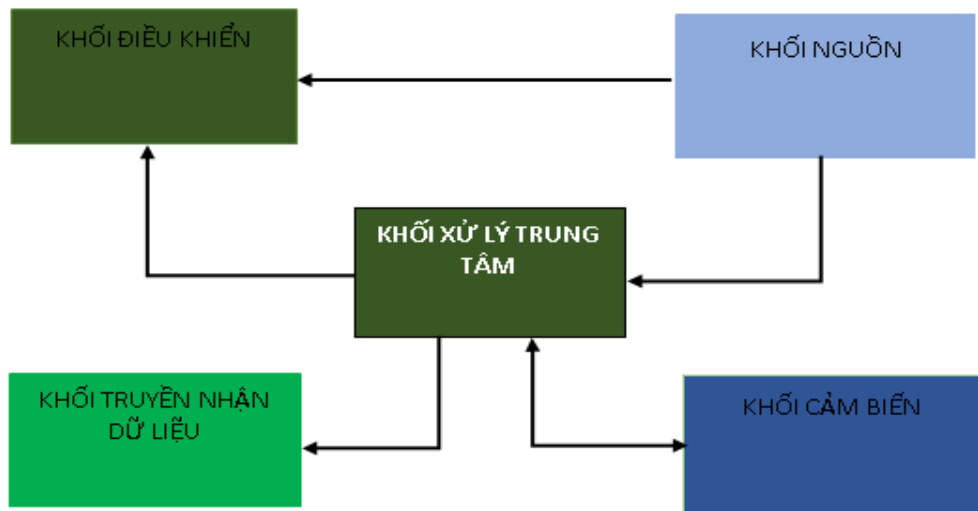
4.2 Sơ đồ kết nối hardware



Hình 22 Sơ đồ Hardware

Hệ thống hoạt động như sau: Laptop sẽ build chương trình vào STM32F4 thông qua thư viện Waijung Blockset. Sau đó STM32F4 làm việc xử lý tín hiệu Encoder đưa vào và gửi các tín hiệu PWM và DIR cho cầu H để tiến hành điều khiển động cơ. Toàn bộ tín hiệu trạng thái hệ thống cũng như điện áp hoặc reference trajectory được sinh viên đọc từ STM32F4 thông qua cổng giao thức UART.

Sơ đồ hệ con lắc ngược xe [18]:

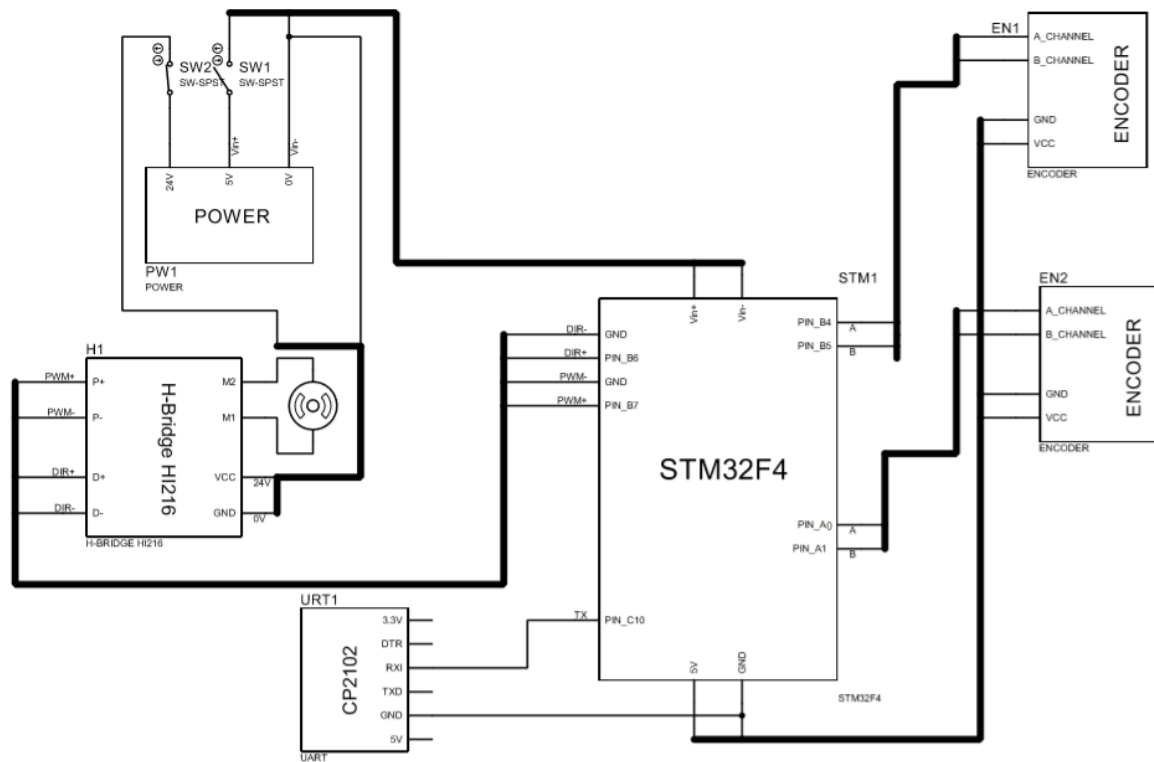


Hình 23. Sơ đồ truyền nhận dữ liệu

Nguyên lí hoạt động của hệ thống:

- Khối nguồn cung cấp điện cho khối xử lý trung tâm (vi điều khiển) và mạch công suất điều khiển động cơ.
- Khối điều khiển nhận tín hiệu điều khiển từ khối xử lý trung tâm và điện áp từ nguồn để điều khiển động cơ DC.
- Khối xử lý trung tâm nhận tín hiệu cảm biến và thực hiện xử lý sau đó đưa ra tín hiệu điều khiển cho khối điều khiển động cơ.
- Khối cảm biến sẽ đọc các tín hiệu cảm biến và đưa kết quả đọc được về khối xử lý trung tâm.
- Khối truyền nhận dữ liệu có tác vụ nhận dữ liệu từ khối điều khiển trung tâm để truyền dữ liệu sang máy tính, đọc các giá trị thu thập được trên Terminal.

Tóm tắt cách kết nối hệ thống:



Hình 24. Kết nối điện hệ thống

4.3 Chi tiết các thiết bị được sử dụng

4.3.1 STM32F4

STM32F4 là vi xử lý được dùng nhiều trong các hệ SIMO vì tốc độ cao và giá thành hợp lý (theo tài liệu [19]). STM32F4 giải quyết được nhiều nhược điểm của các loại vi xử lý khác như: Arduino (tốc độ thấp), TMS320F28335 (giá thành mắc),... Ngoài ra, STM32F4 có thể được nhúng chương trình từ Matlab/Simulink để ứng dụng nghiên cứu các giải thuật điều khiển học thông qua thư viện Waijung Blockset.



Hình 25. STM32F4

Thông số kỹ thuật:

- Hoạt động ở tốc độ cao, có thể lên đến 160 MHz.
- Có thể lập trình trên Simulink thông qua bộ thư viện Waijung Blockset.
- Vi điều khiển chính: STM32F407VGT6 microcontroller featuring 32-bit ARM Cortex-M4F core, 1 MB Flash, 192 KB RAM.
- Nguồn cấp từ cổng Mini USB qua các IC nguồn chuyển thành 3V3 để cấp cho MCU.
- Có sẵn các chân nguồn: 3V3 và 5V.

Bảng 3. Thông số kỹ thuật STM32F4

Vi điều khiển	STM32F4
Điện áp hoạt động	5v
Tần số hoạt động	Lên đến 168 MHz
Số ngõ vào ra	Lên đến 140 ngõ, có khả năng ngắt
Dòng tối đa trên mỗi chân I/O	25mA

Dòng ra tối đa(3v)	100mA
Dòng ra tối đa(5v)	100mA
Bộ nhớ Flash	1MB
Led	8 led

Datasheet: <https://www.alldatasheet.com/datasheet-pdf/pdf/504999/STMICROELECTRONICS/STM32F407VG.html>

4.3.2 H-Bridge HI-216

Trong đề tài luận văn TN, để có thể điều khiển tốc độ động cơ cũng như chiều quay của động cơ. Sinh viên sử dụng H – Bridge HI – 216 để nhận tín hiệu điều khiển từ vi điều khiển STM32F4 sau đó xử lý băm xung điều chỉnh điện áp và đảo chiều động cơ.



Hình 26 H Bridge HI-216

Mạch cầu H là mạch được cấu tạo bởi 4 transistor hay FET. Đôi khi mạch cầu H cũng được cấu tạo bởi 2 transistor hay FET. Tác dụng của các transistor hay FET là các van đóng mở dẫn dòng điện từ nguồn xuống tải với công suất lớn. Tín hiệu điều khiển các van là tín hiệu nhỏ (điện áp hay dòng điện), cho dẫn dòng và điện áp lớn để cung cấp cho tải.

Tín hiệu điều khiển (tín hiệu đầu ra của vi điều khiển) thường nhỏ hơn 5V (do các điều chế PWM) mà muốn điều khiển động cơ lại cần dòng và điện áp lớn. Các

van điều khiển hay các chân điều khiển chỉ cần tín hiệu nhỏ (điện áp hoặc dòng điện) là mở khóa (transistor) dẫn dòng cho tải. Thế nên chúng ta mới dùng mạch cầu H.

Mạch cầu H có thể đảo chiều dòng điện qua tải nên hay được sử dụng trong các mạch điều khiển động cơ DC và trong các mạch bấm áp. Đối với mạch điều khiển động cơ thì cầu H có thể đảo chiều một cách đơn giản, chỉ cần mở khóa các van đúng chiều mà mình muốn.

Board cầu H HI216 dùng IC kích FET chuyên dụng, cho FET luôn dẫn tốt nhất, tránh hiện tượng trùng dẫn, giúp công suất và hiệu năng của board luôn đạt ngưỡng dẫn tốt nhất và ít hư hỏng FET, Module này chuyên dùng cho điều khiển tốc độ động cơ công suất lớn, điều khiển vị trí, vận tốc DC Motor, có thể lựa chọn mức kích âm hoặc dương để dễ dàng điều khiển.

Nguyên lý hoạt động:

Cấp mass của board STM cho chân DIR- và chân PWM- thì lúc đó ta sẽ điều khiển tốc độ bằng PWM+, nghĩa là:

Độ rộng xung mức 1 của PWM càng lớn thì tốc độ của động cơ càng nhanh và đạt maximum khi PWM=max, và ngược lại độ rộng xung mức 1 càng nhỏ thì tốc độ của động cơ càng chậm và dừng hẳn khi PWM=min.

Chân PWM+ lúc này nối với chân PWM của VDK.

Chân DIR+ cấp mức 1 thì động cơ quay thuận, cấp mức 0 động cơ quay nghịch.

Thông số kỹ thuật:

Dòng liên tục 15A, dòng đỉnh 20A, công suất 600W, tại 25°C.

Điện áp công suất từ 12V-48V.

Điện áp kích từ 3.3V-5V.

Có cầu chì bảo vệ ngắn mạch.

Có Led báo nguồn (POW), tín hiệu xung đưa vào (PWM) và tín hiệu đảo chiều (DIR).

Board được thiết kế nhỏ gọn với kích thước 52x64x22mm.

Tần số hoạt động lên tới 100KHz, sử dụng Opto HCPL-0631 cho tần số hoạt động cao.

Độ rộng xung 100%.

Ứng dụng:

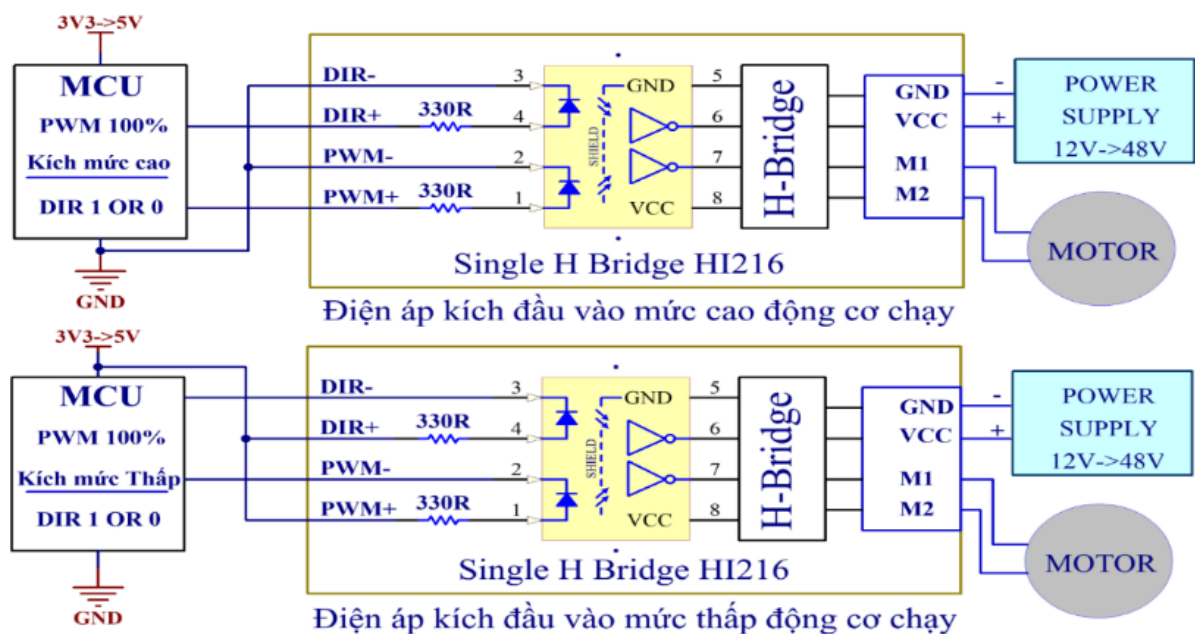
Điều khiển động cơ DC.

Điều khiển vị trí, vận tốc các SERVO.

Trong máy in, xe tự hành, xe điện, ... có sử dụng động cơ DC.

Dùng trong ROBOT công nghiệp.

Sơ đồ kết nối:



Hình 27 Sơ đồ mạch cầu H HI-216

Datasheet:

[http://nhatbon.vn/upload/files/Manual%20H%20Brigge%20HI216\(1\).pdf](http://nhatbon.vn/upload/files/Manual%20H%20Brigge%20HI216(1).pdf)

4.3.3 Động cơ DC Minertia

Để có thể tạo ra moment tải tương đối điều khiển con trượt qua lại (điều khiển vị trí Cart) của hệ con lắc ngược trên xe, động cơ minertia motor YASKAWA ugfmed - 03sri21 được sử dụng (như **Error! Reference source not found.**)



Hình 28 YASKAWA Minertia Motor

Thông số kỹ thuật:

Động cơ: DC Minertia Motor

Mã kỹ thuật : UGFMED – 03SRI21

Nguồn: DC-24V

Tốc độ: 3000 rpm

Đường kính động cơ: 60mm

Chiều dài: 98mm (tính luôn encoder)

Đường kính trục: 10mm

Output power: 50W

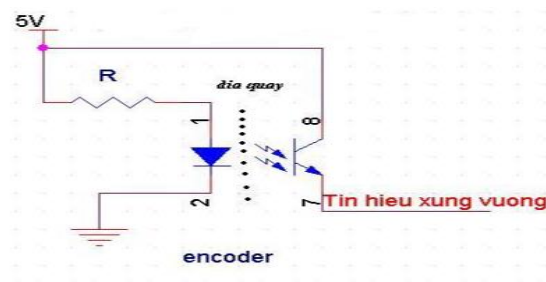
Khối lượng: 900g

Encoder: 400 xung/ vòng , 5V

Các thông số bên trong động cơ được sinh viên trình bày phương pháp ở phần XX.

Cấu trúc:

Bên trong cảm biến Encoder là một Opto như Hình 29 dưới đây. Encoder có 4 dây, bao gồm: 2 dây kênh (A, B), dây + , dây -.



Hình 29 Hoạt động Encoder bên trong động cơ

Tham khảo: <https://www.china.cn/weixingdiandongji/2581425297.html>

4.3.4 Encoder



Hình 30. Encoder tương đối 600 xung

Bảng 4 Thông số Encoder

Model	Vcc	GND	Kênh A	Kênh B	Resolution
Encoder	5V-Red	0V-Black	Green	White	600ppr

Nguyên lý hoạt động:

Nguyên lý cơ bản của Encoder tương đối, đó là một đĩa tròn xoay, quay quanh trục. Trên đĩa có các lỗ (rãnh). Người ta dùng một đèn Led chiếu lên mặt đĩa. Khi đĩa quay, chỗ không có lỗ (rãnh), đèn Led không chiếu xuyên qua được, chỗ có lỗ (rãnh), đèn Led chiếu xuyên qua. Khi đó phía mặt bên kia của đĩa người ta đặt một con mắt thu. Với tín hiệu có hoặc không có ánh sáng, chiếu qua người ta ghi nhận được Led có chiếu qua lỗ hay không.

Khi trục quay, giả sử trên đĩa có một lỗ duy nhất, cứ mỗi lần con mắt thu nhận được tín hiệu Led thì có nghĩa là đĩa đã quay được một vòng.

Tham khảo: <https://vi.aliexpress.com/item/32857035640.html>

4.3.5 UART CP2102

Ở đây, ta sẽ sử dụng UART CP2102 để kết nối đưa dữ liệu để hiển thị trên màn hình máy tính. CP2102 là chip USB to UART của Silabs. CP2102 có kích thước nhỏ gọn và yêu cầu rất ít thành phần bên ngoài để hoạt động được ngay.



Hình 31 UART CP-2102

Thông số kỹ thuật:

CP2102 không sử dụng thạch anh ngoài như các chip PL2303.

Điện áp ngõ ra 3.3V dòng tải tối đa 100mA.

Trên mạch có 6 cổng: 3.3V, RST, 5V, Tx, Rx, GND.

RST dùng để ngắt kết nối giữa Module USB với máy tính.

RST có thể được chỉnh sửa thành chân Reset dùng để nạp chương trình cho Arduino Pro Mini mà không phải nhấn nút Reset.

Tốc độ tối đa: 115200 bps.

Datasheet:

[..https://pdf1.alldatasheet.com/datasheetpdf/view/201067/SILABS/CP2102.html](https://pdf1.alldatasheet.com/datasheetpdf/view/201067/SILABS/CP2102.html)

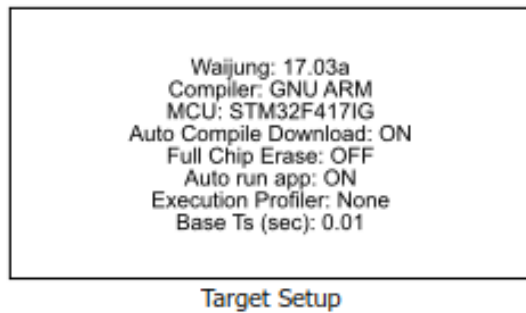
4.4 Bộ thư viện waijung

4.4.1 Bộ thư viện Waijung Blockset [20]

Waijung Blockset (<http://waijung.aimagin.com>) là một thư viện gồm các khối được xây dựng trên nền tảng C của Matlab do một công ty THÁI phát triển mục đích là giảng dạy trên phần cứng họ thiết kế (dựa trên các board của ST), có chức năng cấu hình cho tất cả ngoại vi của vi điều khiển STM32F405xx-407xx-415xx-417xx (từ 64 pin đến 176 pin) với các loại cảm biến, module LAN, WIFI, CAN, USB, SPI, CAN, UART, I2C, SP2, GPS, GMS, RTC,...LED7, LCD, GLCD. Các khối trong Waijung Blockset đều có khả năng xuất code cho vi điều khiển STM32F4 khi sử dụng tính năng Real-time Workshop, phù hợp các phiên bản MATLAB 2009a, 2009b trở đi. Giao diện của từng khối rất dễ sử dụng và được mô tả bằng tiếng anh rất chi tiết (nút help trên từng khối), Cho phép sử dụng các khối của Matlab với Waijung.

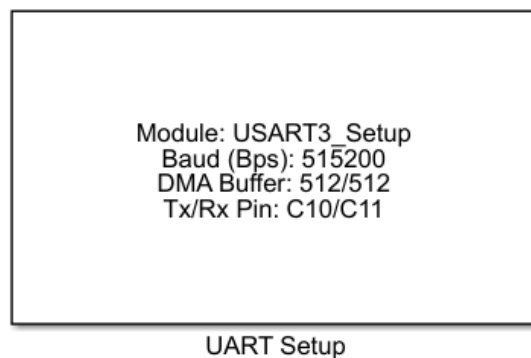
Một số khối Matlab/Simulink của thư viện được sử dụng trong đề tài luận văn này:

- Khối Target Setup: Dùng để khai báo vi xử lý STM mình sử dụng, trình biên dịch, thời gian lấy mẫu, ...



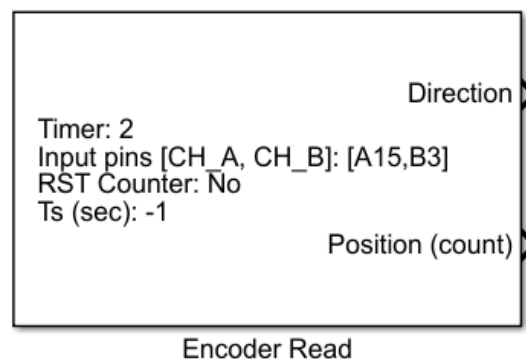
Hình 32. Khởi Target Setup

Khởi UART Setup: Khởi setup chức năng giao tiếp UART, khai báo chân sử dụng cho chức năng này của board xử lí, tốc độ đọc, ...



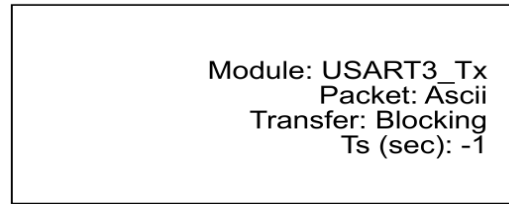
Hình 33 Khởi UART Setup

Khởi Encoder Read: hỗ trợ đọc tín hiệu, lựa chọn chân làm 2 kênh đọc tín hiệu được gửi từ Encoder về.



Hình 34 Khởi Encoder Read

Khởi UART Tx: Trong đề tài luận văn này, khối được sử dụng để truyền dữ liệu lên máy tính để sinh viên quan sát thông qua phần mềm được viết bằng visual studio.



UART Tx

Hình 35. Khối Uart Tx

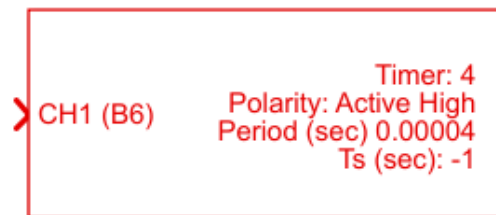
Khối UART Rx: Nhận dữ liệu từ máy tính để thực hiện các tác vụ khi được gọi (Từ phần mềm trong visual studio, ta truyền lệnh – ví dụ là lựa chọn mode quỹ đạo cho hệ thống).



UART Rx

Hình 36. Khối UART Rx

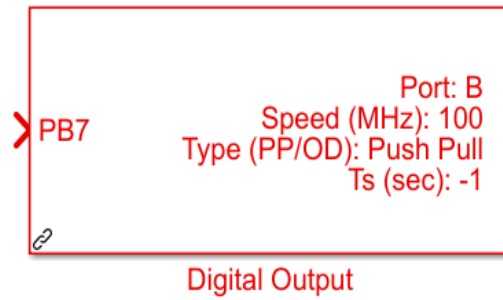
Khối Basic PWM: khai báo Timer và chân sử dụng chế độ PWM, khai báo chu kỳ.



Basic PWM

Hình 37. Khối Basic PWM

Khối Digital Output: xuất tín hiệu Digital điều khiển cầu H IR2184 đảo chiều động cơ.



Hình 38. Khối Digital Output

4.5 WinForm trên Visual Studio

Chương trình:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using ZedGraph;
using System.IO.Ports;
using System.Threading;

namespace App_Test
{
    public partial class Form1 : Form
    {
        //input data receive from serialport
        string sttheta;
        string stposition;
        string stsetpoint;
        string Data;

        public Form1()
        {
            InitializeComponent();
        }

        private void btnConn_Click(object sender, EventArgs e)
        {
            try
            {
                if (cbPort.Text != "")
                {
                    if (cbBaud.Text != "")
                    {
                        serialPort1.PortName = cbPort.Text;
                        serialPort1.BaudRate = Convert.ToInt32(cbBaud.Text);
                        serialPort1.Parity = Parity.None;
                        serialPort1.StopBits = StopBits.One;
                        serialPort1.DataBits = 8;
                        serialPort1.Handshake = Handshake.None;
                        serialPort1.RtsEnable = true;

                        if (serialPort1.IsOpen) return;
                    }
                }
            }
        }
    }
}
```

```

        serialPort1.Open();
        btnConn.Enabled = false;
        btnDisConn.Enabled = true;
        //

        cbBaud.Enabled = false;
        cbPort.Enabled = false;

        btnExit.Enabled = false;

    }
    else
        return;
    }
    else
        return;
}
catch
{
    return;
}
}

//Connect botton//
private void btnDisConn_Click(object sender, EventArgs e)
{
    try
    {
        if (serialPort1.IsOpen == false) return;
        serialPort1.Close();
        btnConn.Enabled = true;
        btnDisConn.Enabled = false;
        //

        cbBaud.Enabled = true;
        cbPort.Enabled = true;

        btnExit.Enabled = true;
    }
    catch
    {
        return;
    }
}

//Serial data receive//
private void serialPort1_DataReceived(object sender,
SerialDataReceivedEventArgs e)
{
    try
    {
        //split data receive from serialport
        string[] arrList = serialPort1.ReadLine().Split(' ');
        sttheta = arrList[0]; // Psi value.
        stposition = arrList[1]; // Theta value.
        stsetpoint = arrList[2]; // Phi value.
        Data = serialPort1.ReadLine();
    }
}

```

```

    }
    catch
    {
        return;
    }
}

int intlen = 0;
private void timer1_Tick(object sender, EventArgs e)
{
    txtKQ.Text = Data;
    if (btnConn.Enabled == false)
    {
        Draw(sttheta, stposition, stsetpoint);
        //display value on graph
        // text color and value
    }
    //auto detect COM port//
    string[] ports = SerialPort.GetPortNames();
    if (intlen != ports.Length)
    {
        intlen = ports.Length;
        cbPort.Items.Clear();
        for (int j = 0; j < intlen; j++)
        {
            cbPort.Items.Add(ports[j]);
        }
    }
}
//
int TickStart1;
int TickStart2;

//
private void Form1_Load(object sender, EventArgs e)
{
    cbBaud.Enabled = true;
    cbPort.Enabled = true;
    //Disable button control

    //Load value//
    cbBaud.Items.Add(9600);
    cbBaud.Items.Add(19200);
    cbBaud.Items.Add(38400);
    cbBaud.Items.Add(57600);
    cbBaud.Items.Add(74880);
    cbBaud.Items.Add(115200);
    cbBaud.Items.Add(230400);
    cbBaud.Items.Add(250000);
    //
    btnDisConn.Enabled = false;

    //Show contents of graph
    //Name graph
    //Value x,y axis..
    GraphPane myPane1 = zedGraphControl1.GraphPane;
    myPane1.Title.Text = "Theta Graph";
    myPane1.XAxis.Title.Text = "Time (s)";
    myPane1.YAxis.Title.Text = "Angle (rad)";
}

```

```

        RollingPointPairList list1 = new RollingPointPairList(60000);
        LineItem Curve1 = myPane1.AddCurve("Theta Value", list1, Color.Blue,
SymbolType.None);

        myPane1.XAxis.Scale.Min = 0;
        myPane1.XAxis.Scale.Max = 10;
        myPane1.YAxis.Scale.Min = -0.5;
        myPane1.YAxis.Scale.Max = 0.5;

        zedGraphControl1.AxisChange();
        TickStart1 = Environment.TickCount;

        /*Display the graph 2 of contents*/
        //Psidot graph//
        GraphPane myPane2 = zedGraphControl2.GraphPane;
        myPane2.Title.Text = "Position of Cart Graph";
        myPane2.XAxis.Title.Text = "Time (s)";
        myPane2.YAxis.Title.Text = " Position (m)";

        RollingPointPairList list2 = new RollingPointPairList(60000);
        LineItem Curve2 = myPane2.AddCurve("Position Value", list2, Color.Blue,
SymbolType.None);
        //
        RollingPointPairList list2_ = new RollingPointPairList(60000);
        LineItem Curve2_ = myPane2.AddCurve("Setpoint", list2_, Color.Red,
SymbolType.None);

        myPane2.XAxis.Scale.Min = 0;
        myPane2.XAxis.Scale.Max = 10;
        myPane2.YAxis.Scale.Min = -0.7;
        myPane2.YAxis.Scale.Max = 0.7;

        zedGraphControl2.AxisChange();
        TickStart2 = Environment.TickCount;

    }
    //Draw line in the zedgraph using string data    //
    //receive from serialport convert to double value//
    private void Draw(string intheta, string inposition, string insetpoint)
    {
        double _theta;
        double _position;
        double _setpoint;

        double.TryParse(intheta, out _theta);//Conver tring to double//
        double.TryParse(inposition, out _position);//Conver tring to double//
        double.TryParse(insetpoint, out _setpoint);//Conver tring to double//

        //Error color text

        if (zedGraphControl1.GraphPane.CurveList.Count <= 0)
            return;
        if (zedGraphControl2.GraphPane.CurveList.Count <= 0)
            return;

        LineItem curve1 = zedGraphControl1.GraphPane.CurveList[0] as LineItem;

```

```

        LineItem curve2 = zedGraphControl2.GraphPane.CurveList[0] as LineItem;

        LineItem curve2_ = zedGraphControl2.GraphPane.CurveList[1] as LineItem;//
SetPoint Curve Theta

        if (curve1 == null)
            return;
        if (curve2 == null)
            return;

        if (curve2_ == null)
            return;

        //
        IPointListEdit list1 = curve1.Points as IPointListEdit;
        IPointListEdit list2 = curve2.Points as IPointListEdit;

        IPointListEdit list2_ = curve2_.Points as IPointListEdit;

        //
        if (list1 == null)
            return;
        if (list2 == null)
            return;

        if (list2_ == null)
            return;

        //
        double time1 = (Environment.TickCount - TickStart1) / 1000.0;
        double time2 = (Environment.TickCount - TickStart2) / 1000.0;

        //
        list1.Add(time1, _theta);
        list2.Add(time2, _position);
        //
        list2_.Add(time2, (_setpoint));

        //
        Scale xScale1 = zedGraphControl1.GraphPane.XAxis.Scale;
        Scale xScale2 = zedGraphControl2.GraphPane.XAxis.Scale;

        //
        Scale yScale1 = zedGraphControl1.GraphPane.YAxis.Scale;
        Scale yScale2 = zedGraphControl2.GraphPane.YAxis.Scale;

        //
        if (time1 > xScale1.Max - xScale1.MajorStep)
        {
            xScale1.Max = time1 + xScale1.MajorStep;
            xScale1.Min = xScale1.Max - 10;//Auto scale x axis in limit time
        }
        if (time2 > xScale2.Max - xScale2.MajorStep)
        {
            xScale2.Max = time2 + xScale2.MajorStep;
            xScale2.Min = xScale2.Max - 10;
        }

        //
        zedGraphControl1.AxisChange();
        zedGraphControl2.AxisChange();

```

```

        //
        zedGraphControl1.Invalidate();
        zedGraphControl2.Invalidate();
    }

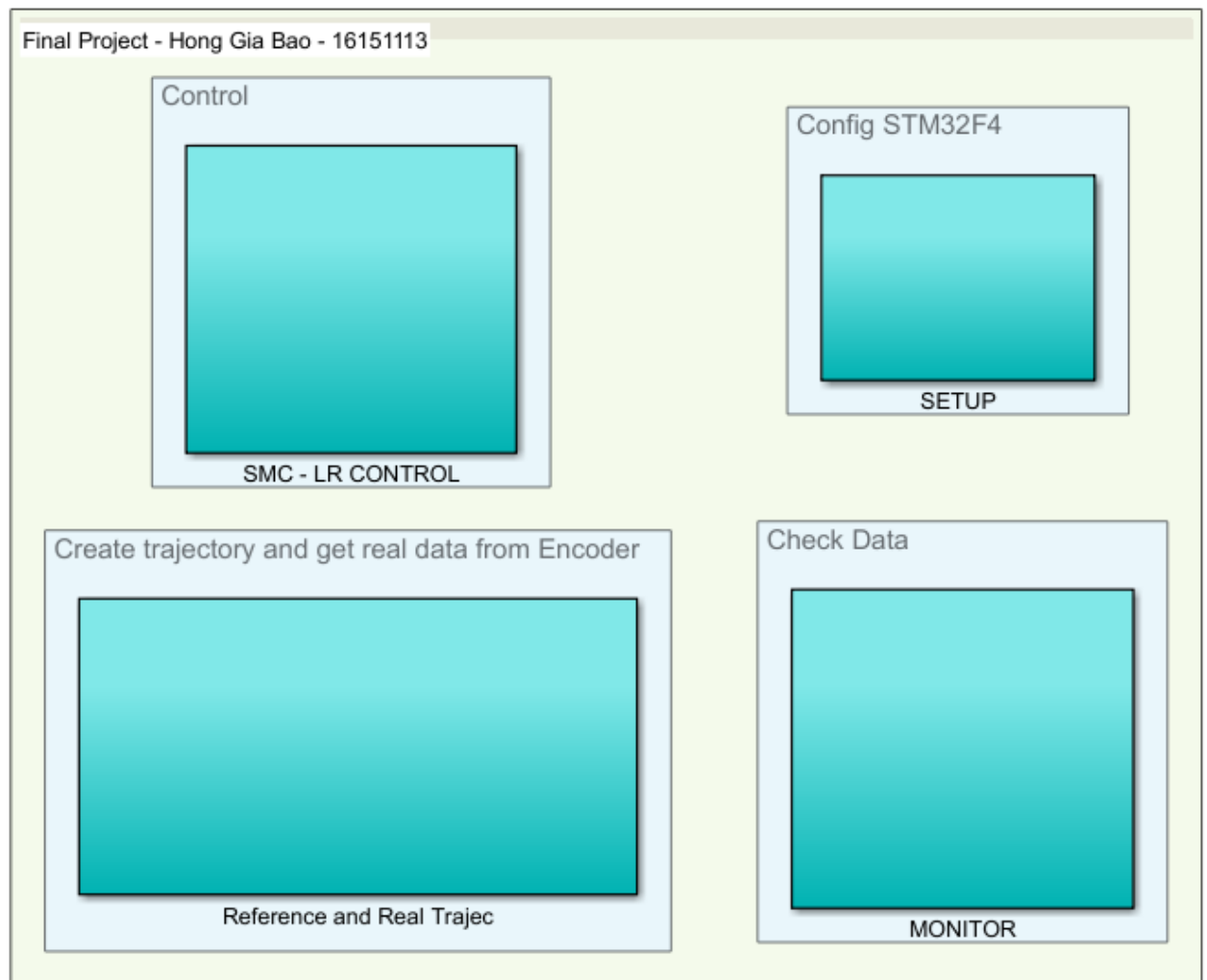
    private void btnExit_Click(object sender, EventArgs e)
    {
        serialPort1.Close();
        Close();
    }

    private void btnSend_Click(object sender, EventArgs e)
    {
        string data_;
        if (serialPort1.IsOpen==true)
        {
            data_ = txtCase.Text;
            serialPort1.WriteLine(data_ + "\r");
        }
        else
        {
            MessageBox.Show("Bạn chưa mở cổng COM");
        }
    }
}
}
}

```

4.5 Chương trình điều khiển Cart and Pole trên thực tế

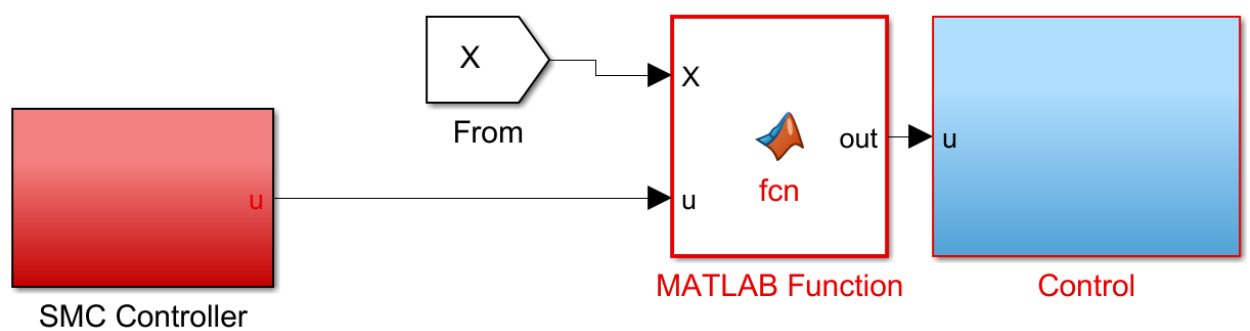
4.5.1 Tổng quát chương trình trên Matlab/Simulink



Hình 39. Chương trình điều khiển thực tế

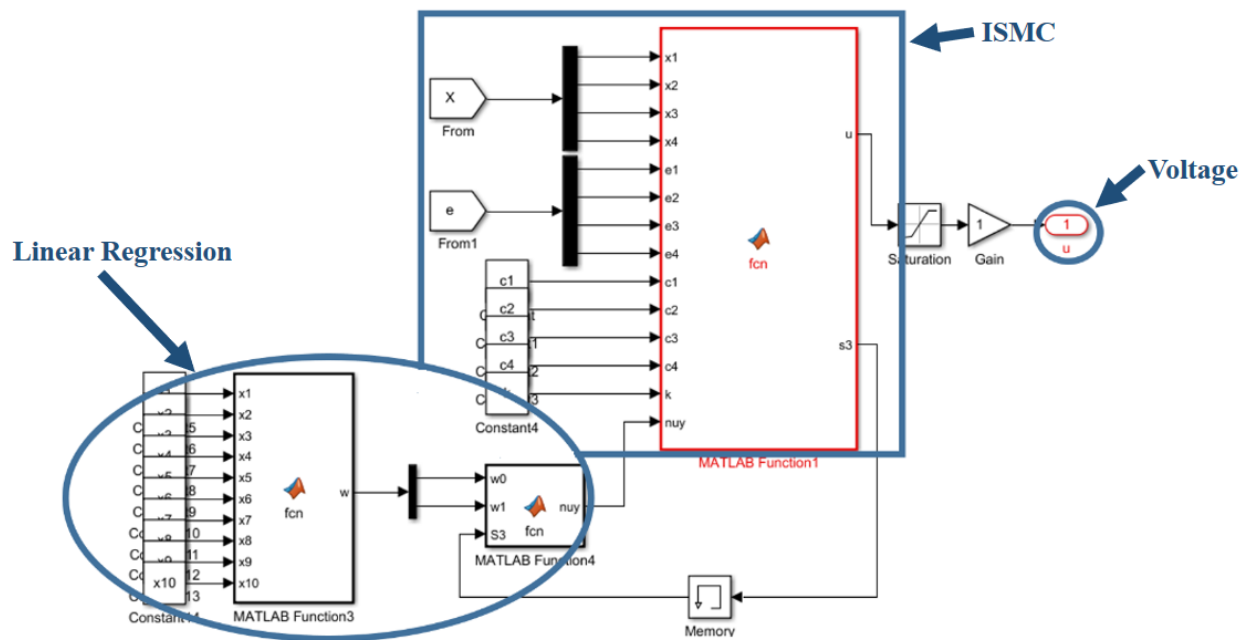
4.5.2 Khối SMC – LR CONTROL

Tổng quát



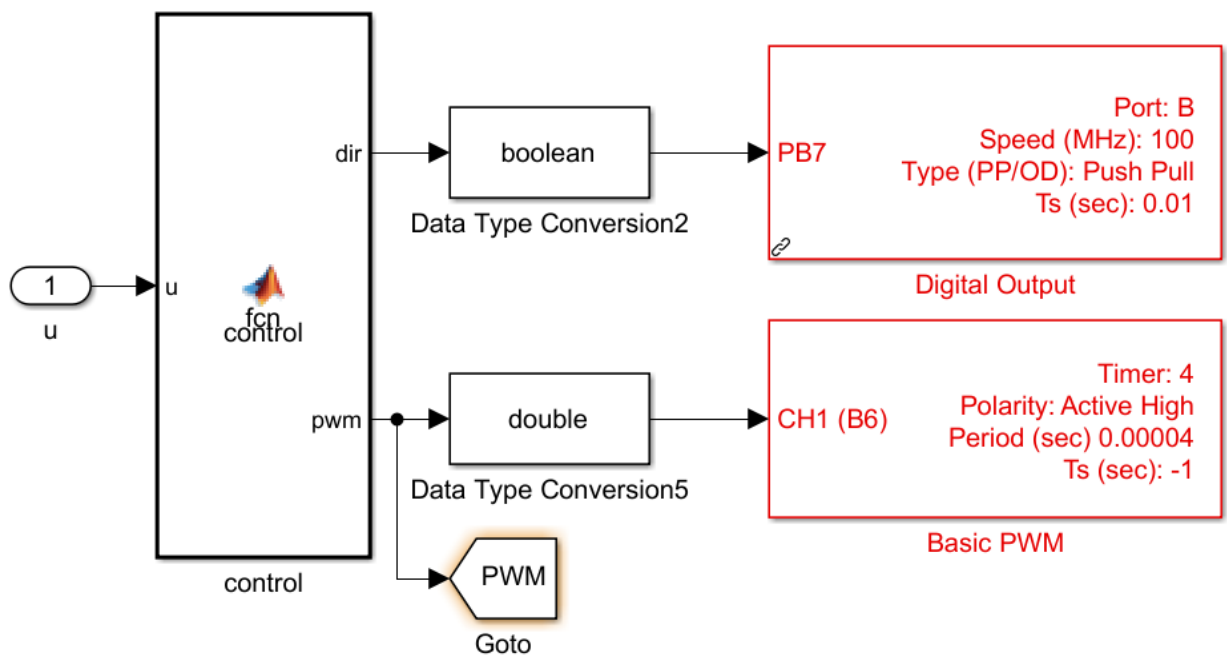
Hình 40. Khối SMC – LR CONTROL

Bên trong khối SMC Controller



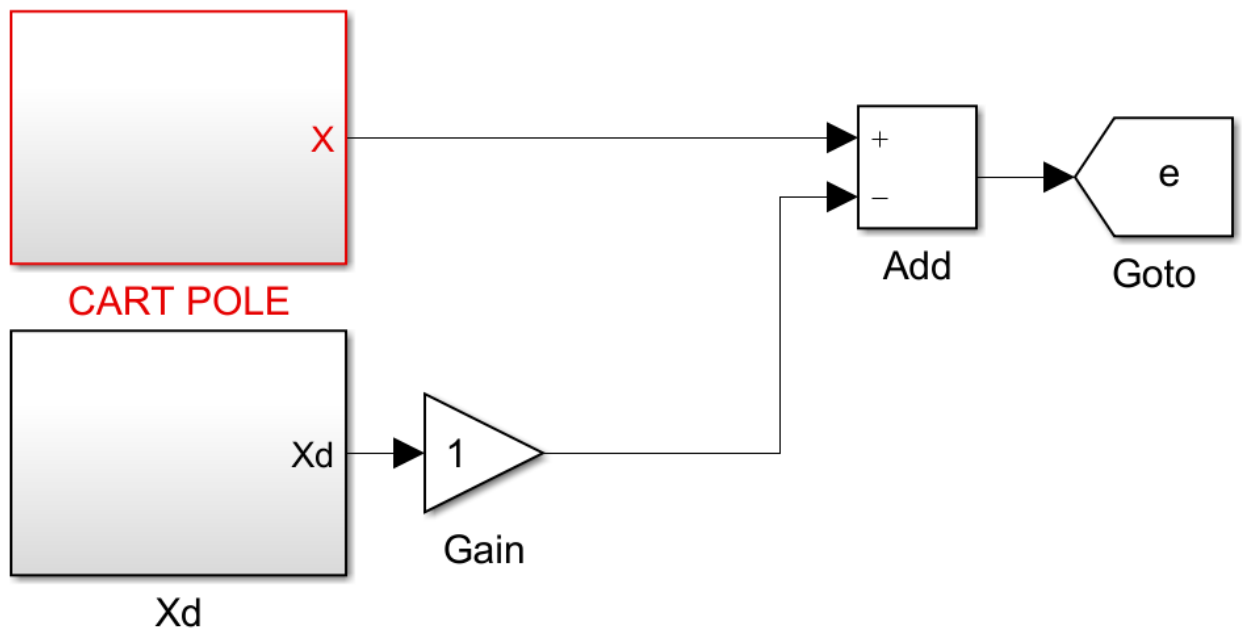
Hình 41. Khối SMC Controller

Bên trong khối Control

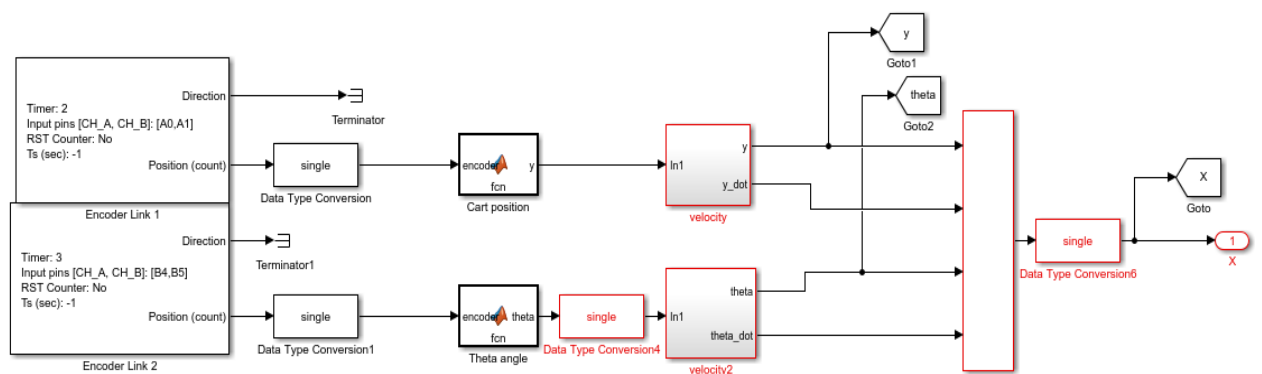


Hình 42. Khối Control

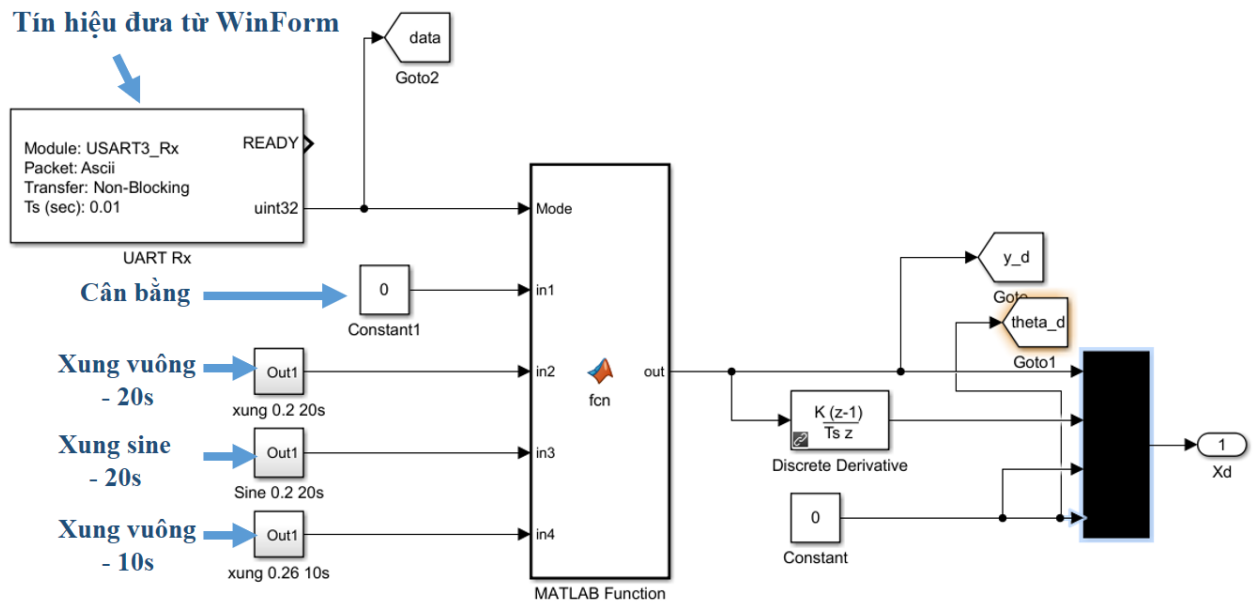
4.5.3 Khối Reference and Real Trajec



Hình 43. Khối Reference and Real Trajec
Bên trong khối CART POLE

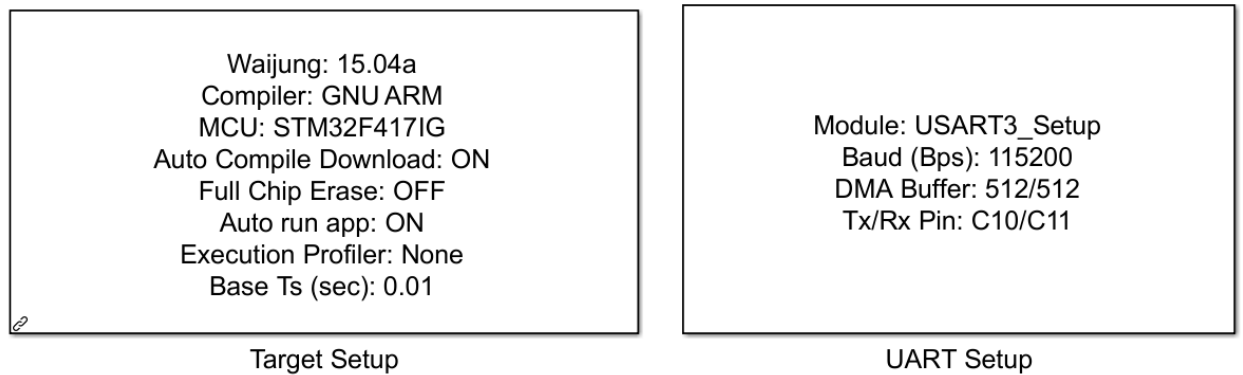


Hình 44. Khối CART POLE
Bên trong khối Xd



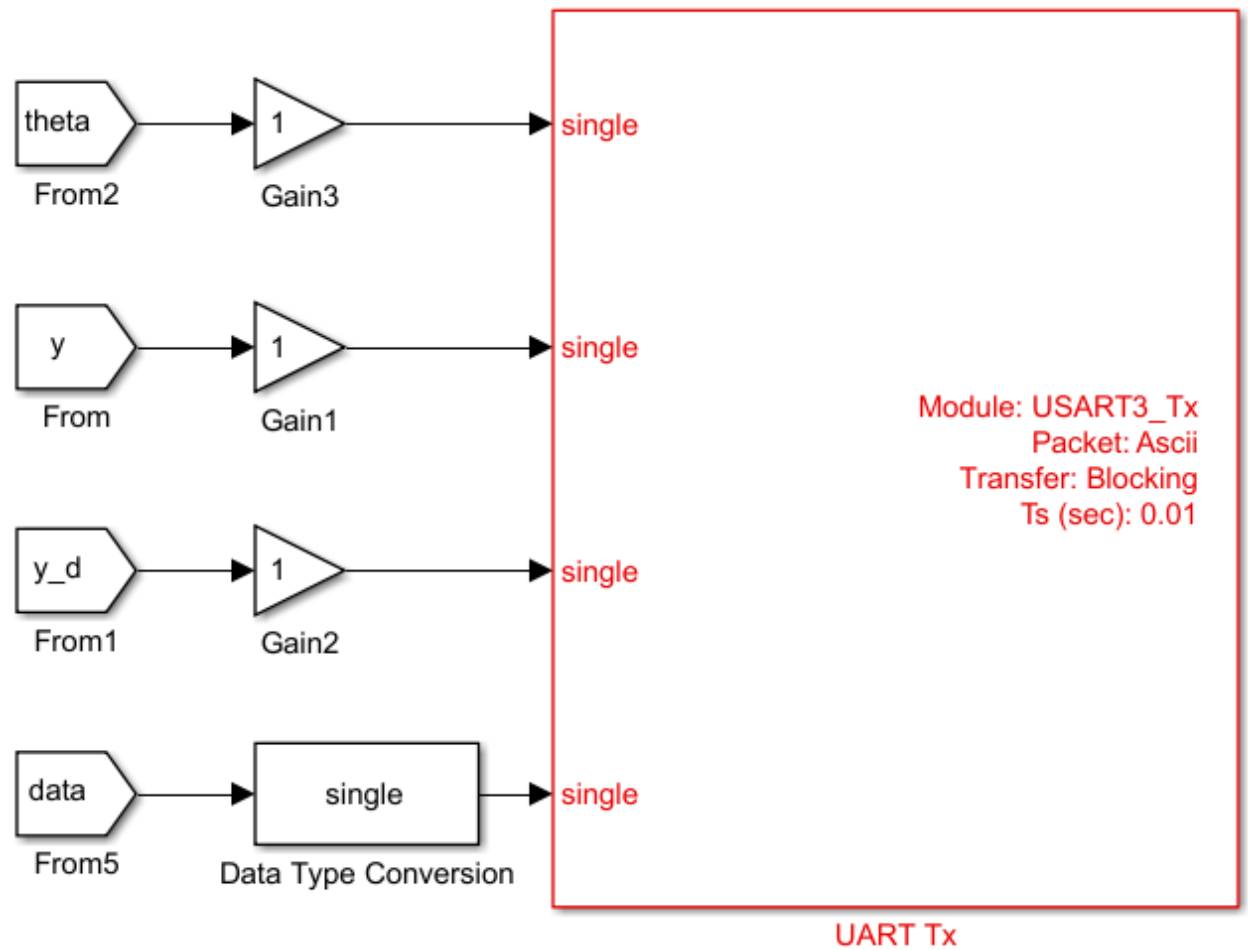
Hình 45. Khối Xd

4.5.4 Khối SETUP



Hình 46. Khối SETUP

4.5.5 Khối MONITOR

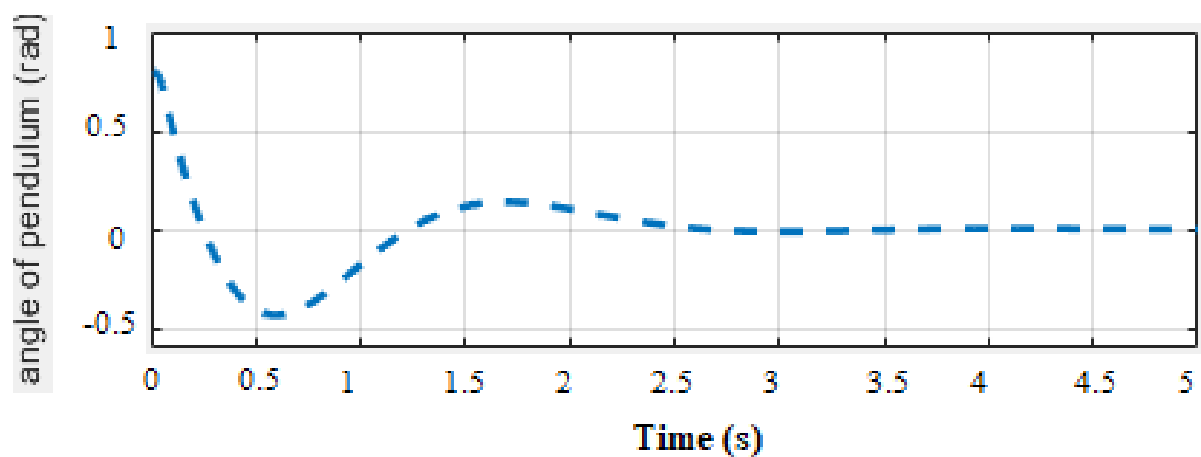


Hình 47. Khối MONITOR

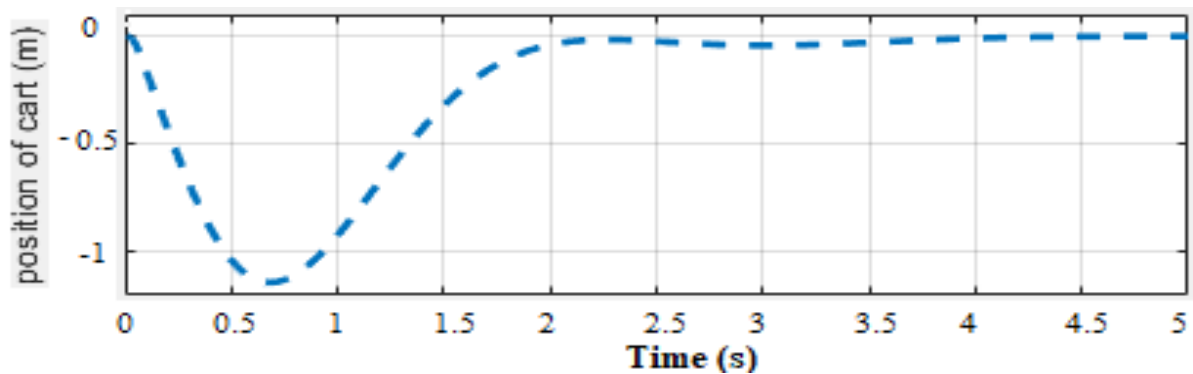
5.1 Kết quả mô phỏng

5.1.1 Cân bằng

Kết quả mô phỏng ở Hình 48- Hình 49 cho thấy bộ điều khiển ISMC điều khiển tốt hệ thống ổn định tại điểm làm việc cân bằng. Cả xe và con lắc được điều khiển về đúng vị trí không. Xe đứng yên và con lắc ngược nằm ở vị trí thẳng đứng, hướng lên.



Hình 48. Góc con lắc dưới bộ điều khiển ISMC – Cân bằng (rad)

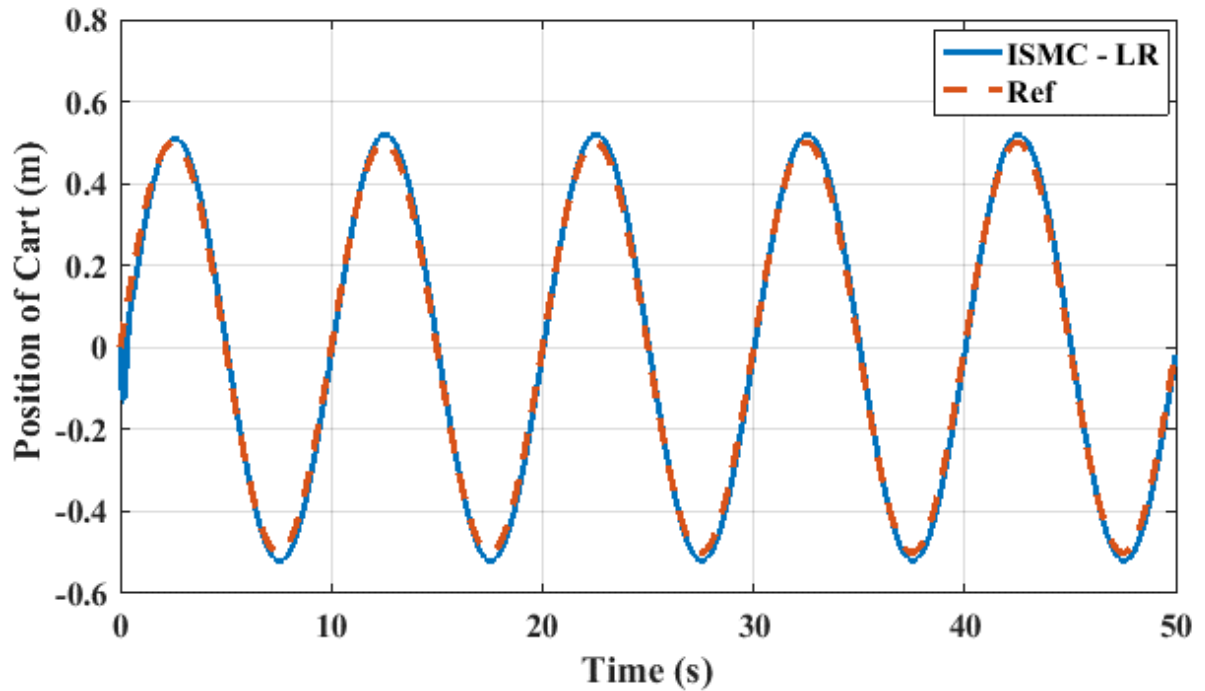


Hình 49. Vị trí xe dưới bộ điều khiển ISMC – Cân bằng (m)

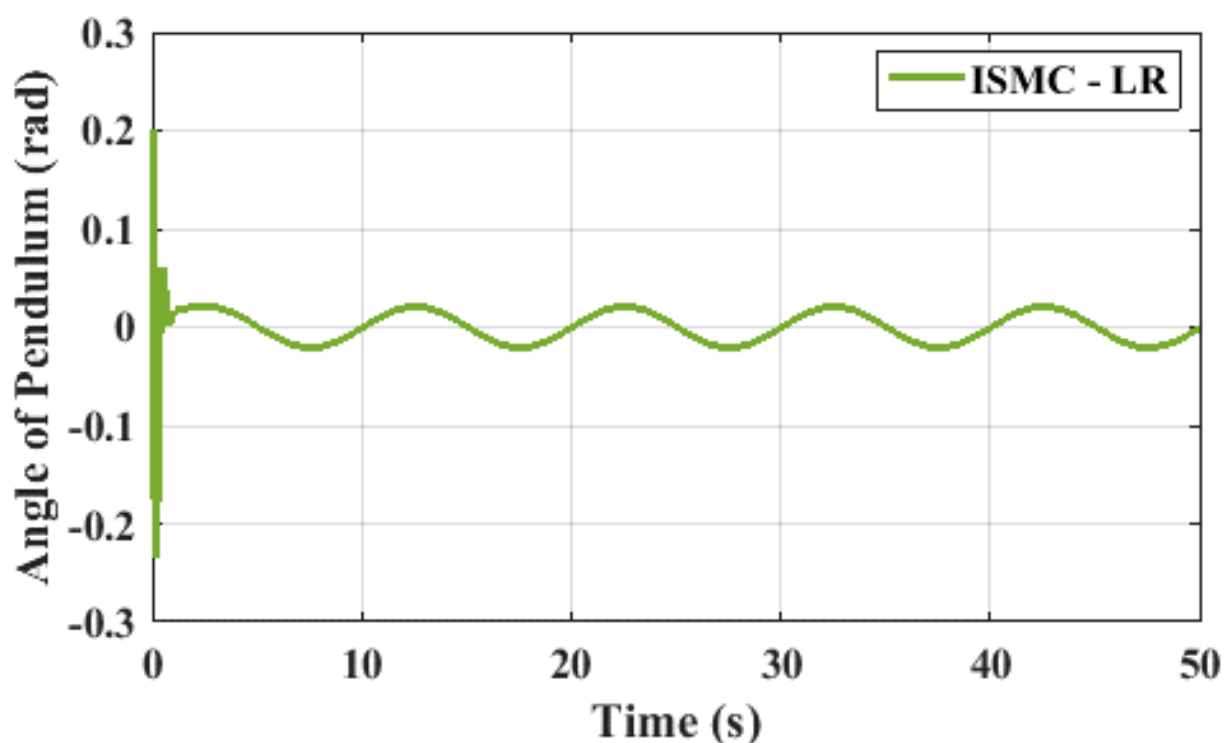
5.1.2 Quỹ đạo sóng sine

Trường hợp thứ 2, sinh viên sẽ cung cấp cho hệ một reference mẫu quỹ đạo – Xe sẽ di chuyển theo giá trị sóng sine trong khi đó luôn giữ pendulum ở vị trí thẳng

đứng hướng lên. Kết quả được đưa ra ở Hình 50 - Hình 51. Giải thuật điều khiển trượt điều khiển thành công hệ thống để khi xe di chuyển theo quỹ đạo hình sine thì con lắc ngược vẫn thẳng đứng và không bị ngã.



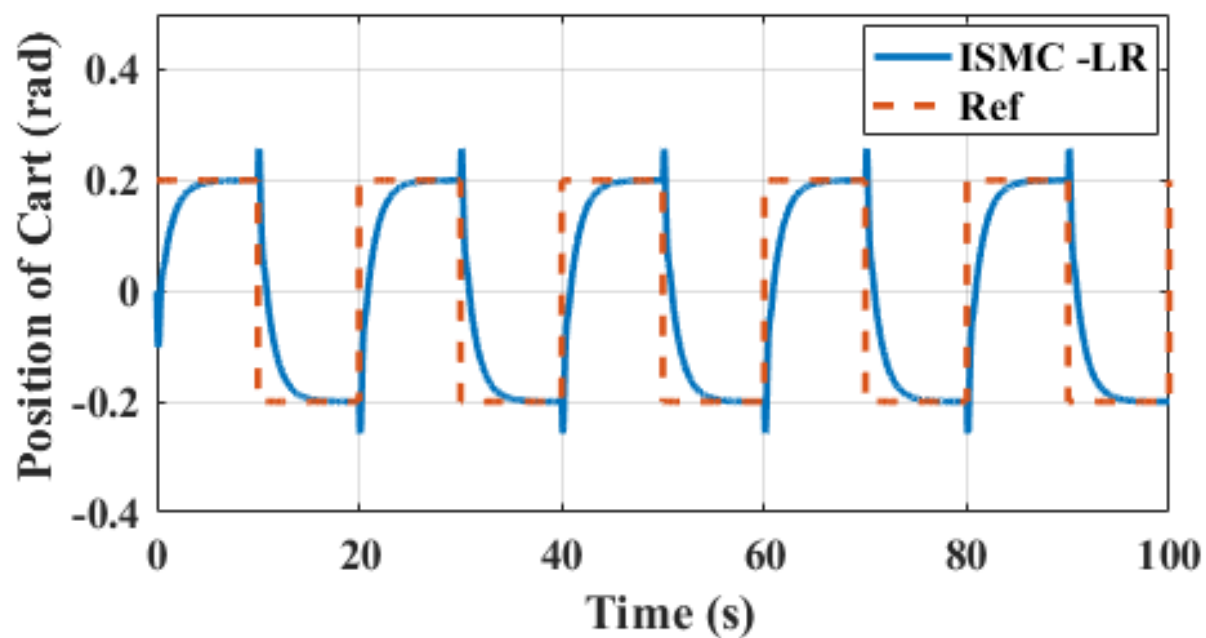
Hình 50. Vị trí xe dưới bộ điều khiển ISMC và LQR - Quỹ đạo sóng sine (m)



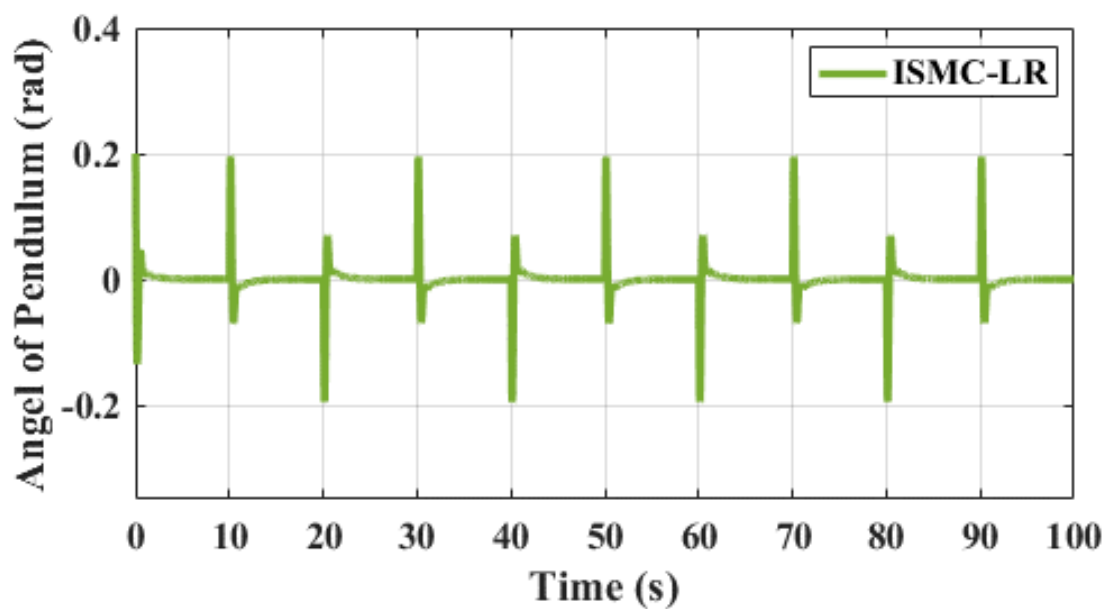
Hình 51. Vị trí con lắc dưới bộ điều khiển SMC và LQR - Quỹ đạo sóng sine (rad)

5.1.3 Quỹ đạo xung vuông

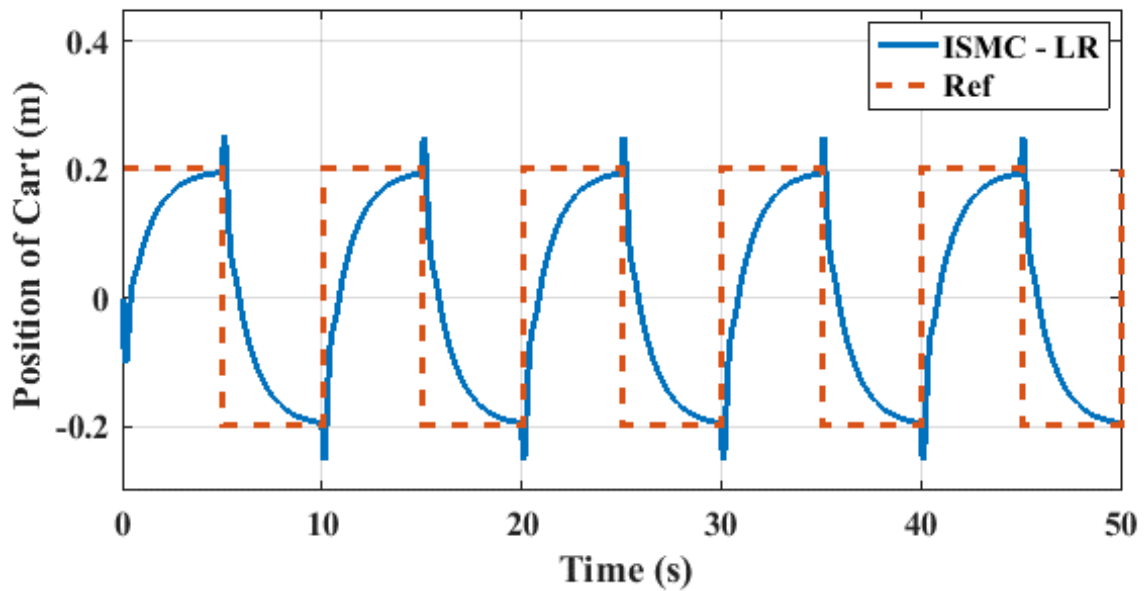
Trường hợp thứ 3, 4: Sinh viên sử dụng tín hiệu mẫu là xung vuông có chu kì lần lượt là 20s và 10s để kiểm chứng khả năng hoạt động của giải thuật. Kết quả từ Hình 52 đến Hình 55 đã cho thấy hệ thống vẫn ổn định. Xe bám theo quỹ đạo xung vuông trong khi con lắc đơn vẫn được giữ vững ở vị trí thẳng đứng. Ở hình Hình 52, thời gian cho vị trí xe xác lập là khoảng 4s, cùng với đó là độ vọt lố rất thấp. Đồng thời, góc con lắc từ vị trí ban đầu đã về lại góc thẳng đứng hướng lên chỉ trong khoảng dưới 2s. Ở trường hợp 4, Hình 53, với tốc độ chuyển vị trí nhanh hơn – chu kì 10s, ta thấy hệ thống vẫn hoạt động khá tốt, xe bám quỹ đạo xác lập trong 4s, đồng thời con lắc vẫn giữ được vị trí thẳng đứng trong khoảng thời gian dưới 2s.



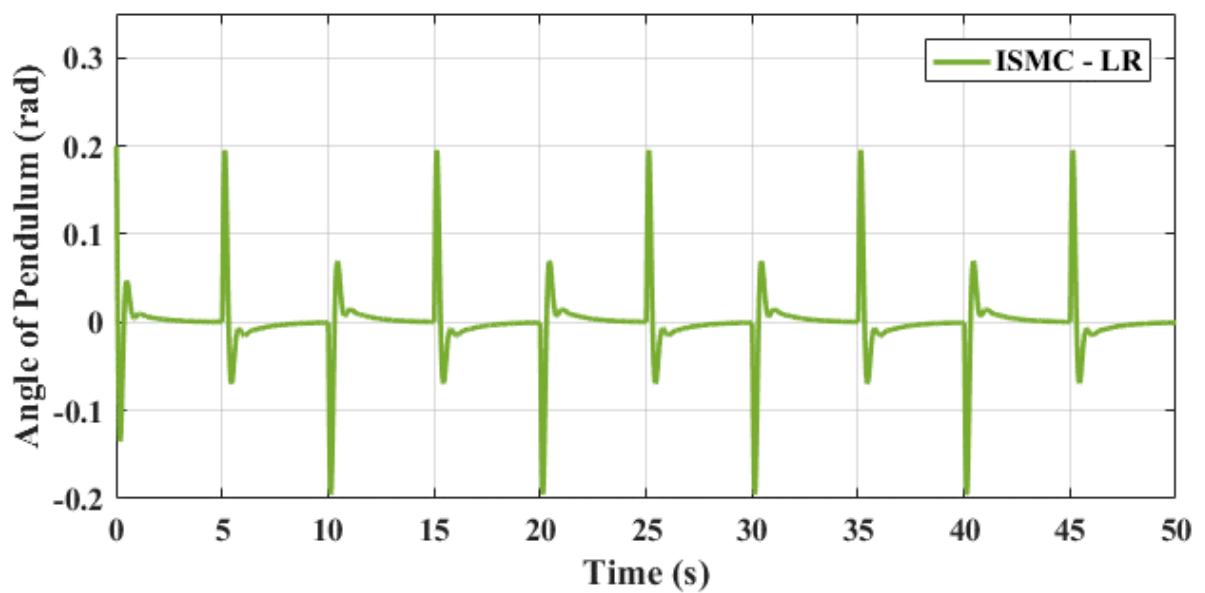
Hình 52. Vị trí xe dưới bộ điều khiển ISMC và LQR - Quỹ đạo xung vuông (m)



Hình 53. Vị trí con lắc dưới bộ điều khiển ISMC và LQR - Quỹ đạo xung vuông (rad)



Hình 54 Vị trí xe dưới bộ điều khiển ISMC và LQR - Quỹ đạo xung vuông 10s (m)

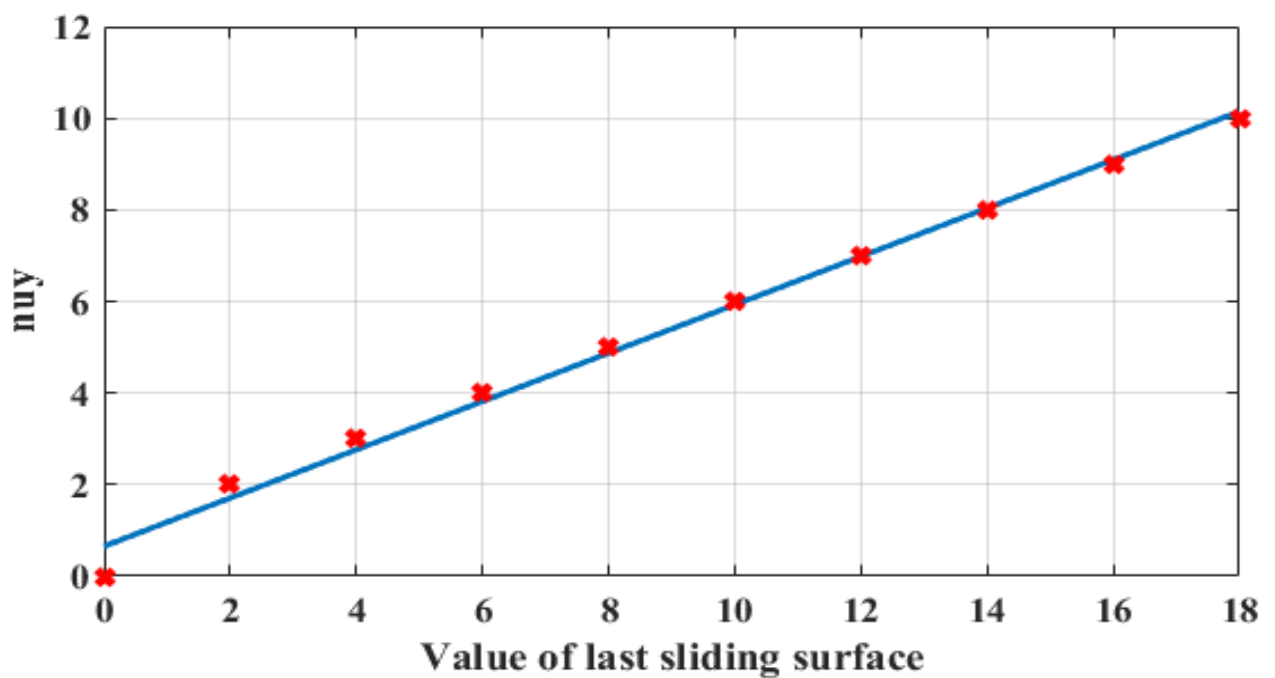


Hình 55 Vị trí con lắc dưới bộ điều khiển ISMC và LQR - Quỹ đạo xung vuông 10s (rad)

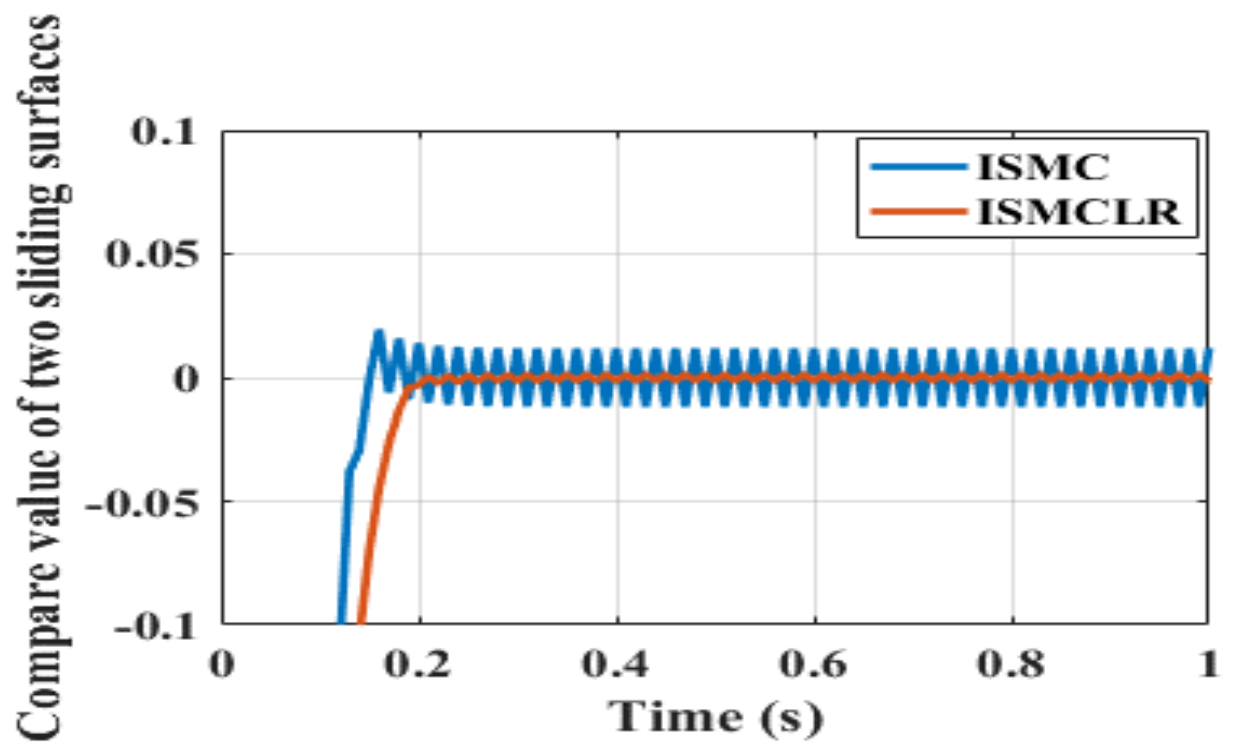
5.1.4 Sự cải tiến khi thêm LR tìm thông số nuy so với thông thường

Sau khi tham khảo qua nhiều tài liệu nói về cả ưu điểm và nhược điểm của giải thuật điều khiển trượt. Sinh viên nhận thấy ưu điểm rất tốt của giải thuật là vì giải thuật này là một giải thuật phi tuyến, nên có thể điều khiển hệ thống bám theo quỹ đạo mong muốn cũng như có thể phát triển lên điều khiển thích nghi ,... vì chúng dùng phương trình toán phi tuyến của mô hình để hoạt động. Tuy vậy, nhược điểm của giải thuật này có thể gây nên nhiều tổn thất cho motor cũng như các cơ cấu khác của hệ thống – chính là chattering, như đã nói ở phần 3.2.

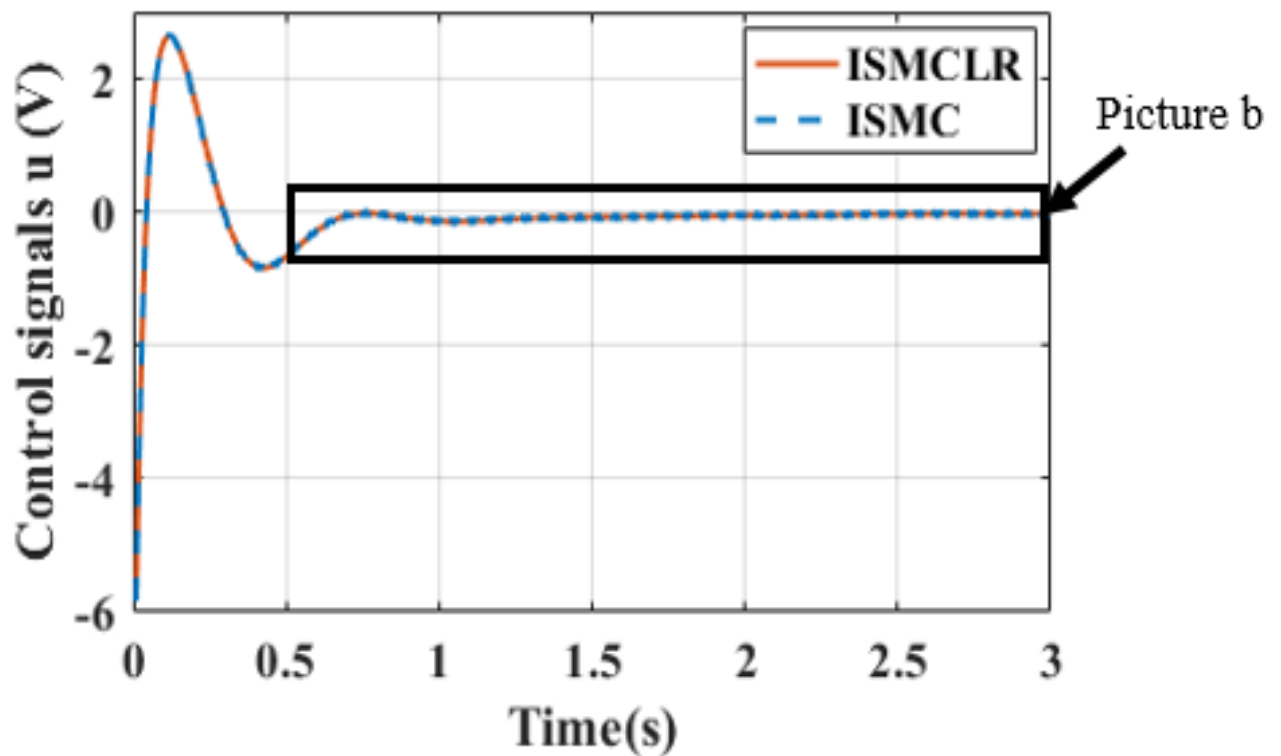
Sau khi áp dụng thêm phương pháp Linear Regression (như sinh viên đã đề cập ở phần 3.3). Dưới đây là các kết quả được sinh viên tổng hợp.



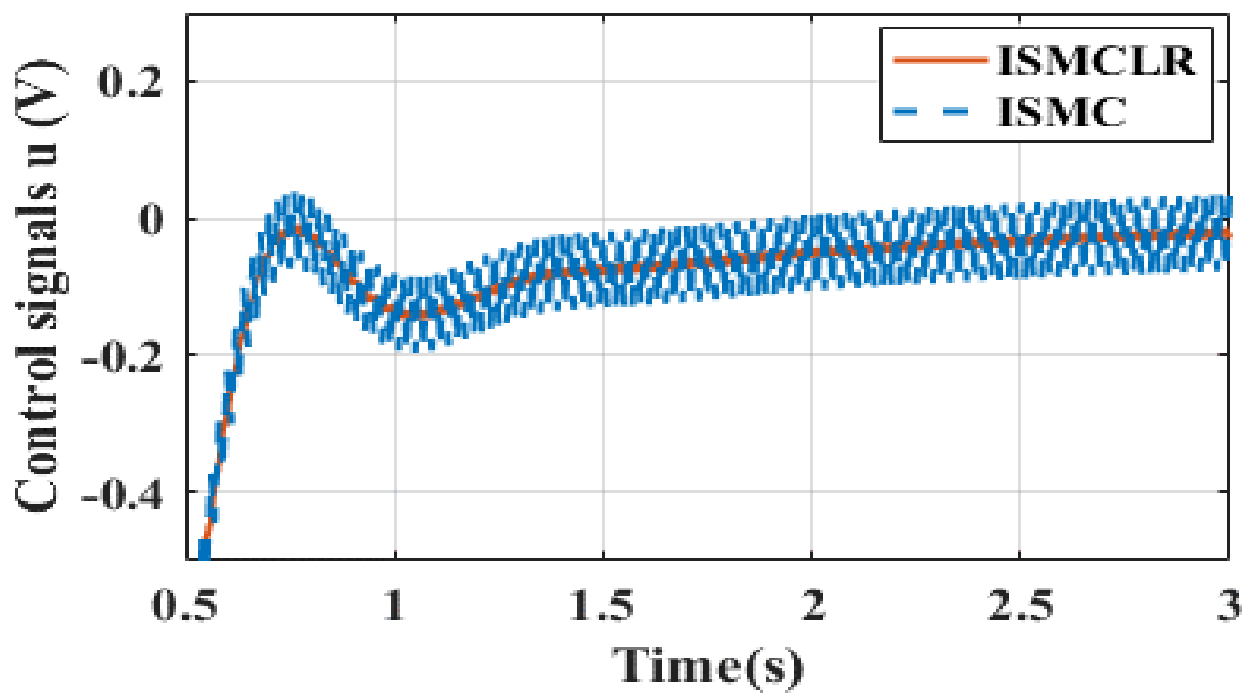
Hình 56. Mối quan hệ giữa mặt trượt tổng và giá trị nuy



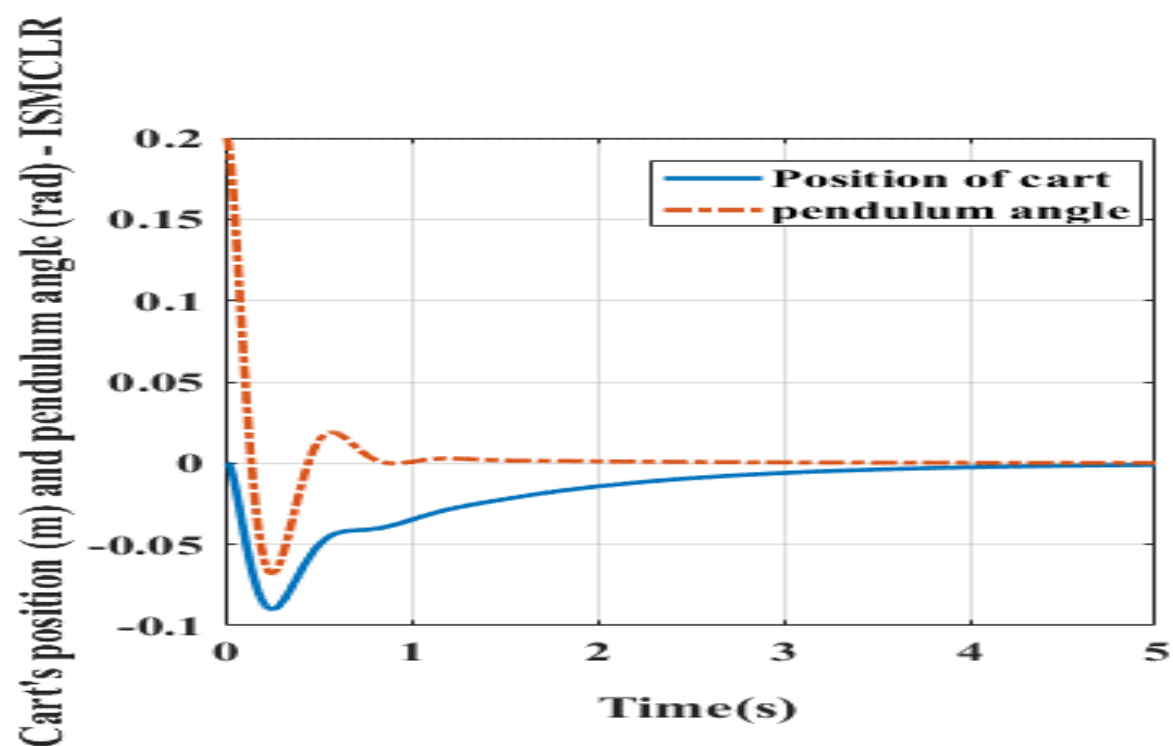
Hình 57. Sự sai khác về chattering giữa 2 phương pháp



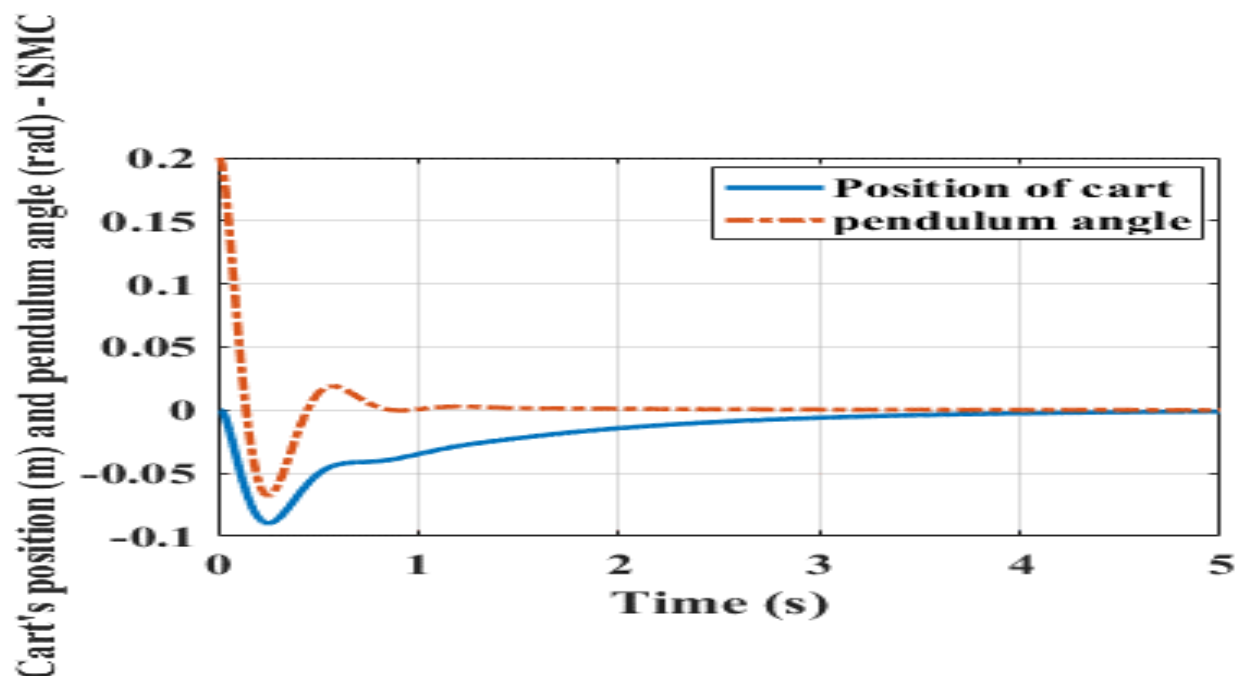
Hình 58. Tín hiệu điện áp cấp vào hệ thống ở cả 2 phương pháp ISMC và ISMC-LR



Hình 59. Picture b – phóng to Hình 58



Hình 60. Vị trí xe và con lắc ở giải thuật ISMC-LR



Hình 61. Vị trí xe và con lắc ở giải thuật ISMC

Tổng quan:

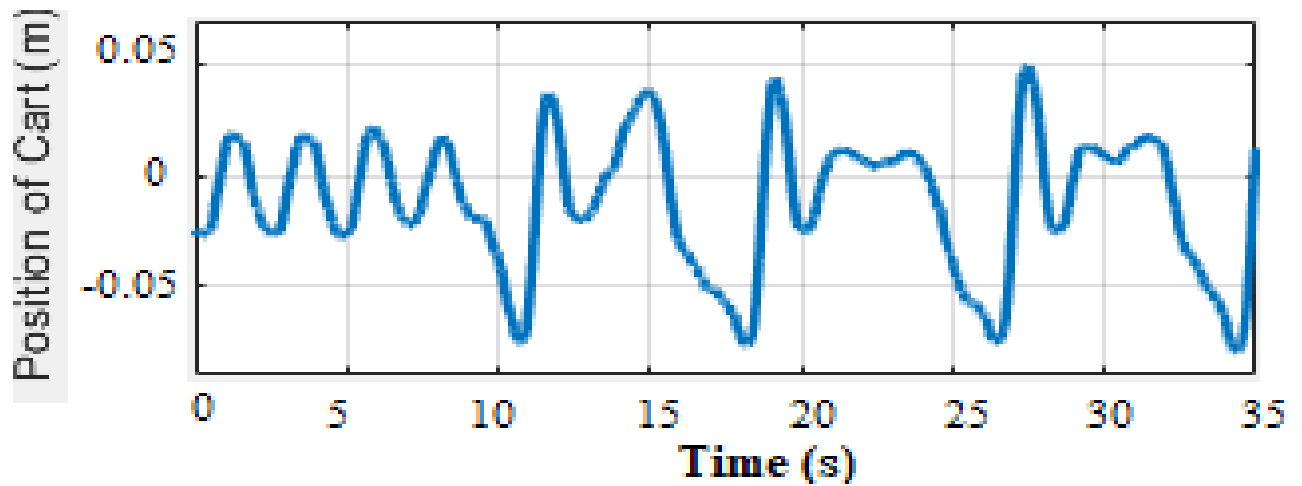
Thông qua các kết quả trên, sinh viên nhận thấy vẫn giữ được kết quả rất tốt của ISMC trong khi chattering trong điện áp cấp vào động cơ đã giảm hơn 4/5. Giảm đi được tổn hao trong động cơ mà vẫn giữ được khả năng điều khiển như ban đầu. Đây là mô phỏng hệ thống Cart and Pole cho mục đích thí nghiệm, nên sự hiệu quả của việc giảm chattering chưa thấy rõ, nhưng với những trường hợp cần điều khiển những hệ thống to lớn và cần một lượng điện áp lớn. Việc giảm chattering là rất cần thiết.

5.2 Kết quả thực nghiệm [21]

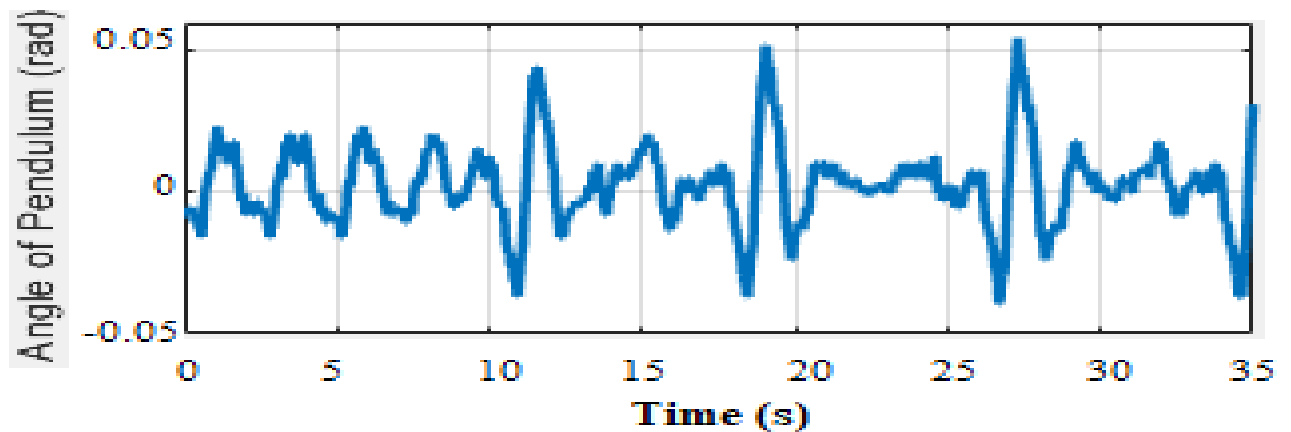
5.2.1 Cân bằng

Ở trường hợp cân bằng, các thông số bộ điều khiển trượt được sử dụng lại. Kết quả cho thấy hệ thống chạy ổn định dưới sự điều khiển của giải thuật. Ở thực tế, các điều kiện hoàn hảo trong mô phỏng sẽ không còn, trường hợp này có thêm các tác động như nhiễu ngoại lai, ma sát, nhận diện thông số động cơ còn sai số,... Tuy nhiên, việc hệ thống vẫn chạy rất tốt chứng tỏ SMC có độ bền vững rất cao dù hệ thống có tính bất ổn. Hai Hình 62 - Hình 63 cho thấy trong trường

hợp cân bằng, xe di chuyển dao động quanh 0 và qua lại khoảng 6cm và con lắc được giữ dao động quanh vị trí thẳng đứng hướng lên. Sở dĩ giá trị không bằng 0 tuyệt đối là vì sai số trong nhận diện động cơ, trong ma sát (phương trình hệ thống sinh viên xem xét giá trị này gần bằng 0),...



Hình 62. Vị trí xe (m) – Cân bằng



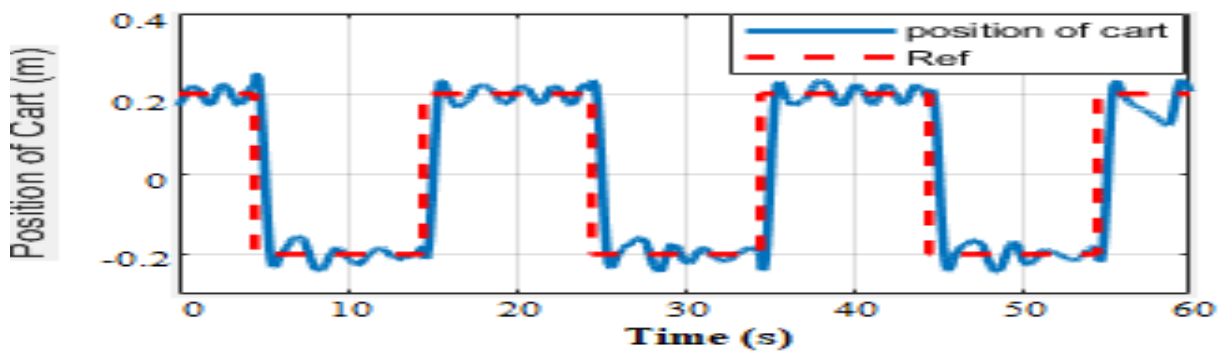
Hình 63. Góc con lắc (rad) – Cân bằng

5.2.2 Quỹ đạo xung vuông

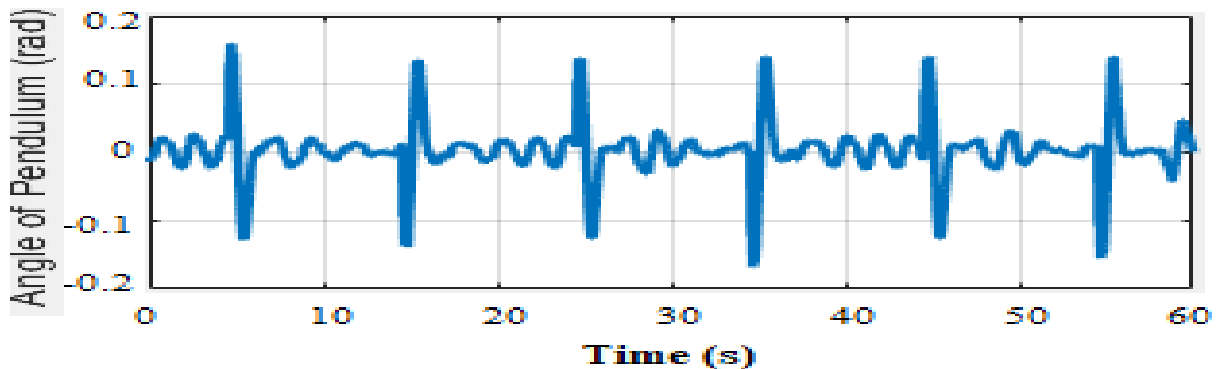
Bám quỹ đạo là một phương thức điều khiển dựa trên tính điều khiển ổn định của giải thuật. Khi một bộ điều khiển không có khả năng điều khiển ổn định hệ thống tại điểm cân bằng thì việc cho hệ bám quỹ đạo là không thể. Và nếu phương pháp chúng ta dùng là các phương pháp điều khiển tuyến tính (giả sử phương trình hệ thống là một dạng tuyến tính) thì việc điều khiển bám quỹ đạo là không khả thi vì

nó không đúng với phương trình thực tế của hệ thống. Vì thế, để có thể điều khiển bám quỹ đạo, giải thuật điều khiển trượt được sinh viên sử dụng vì đây là một giải thuật phi tuyến. Tín hiệu mẫu được đặt trước trong phần này là xung vuông có chu kỳ 20s (Hình 64 và Hình 65) và 10s (Hình 66 và Hình 67).

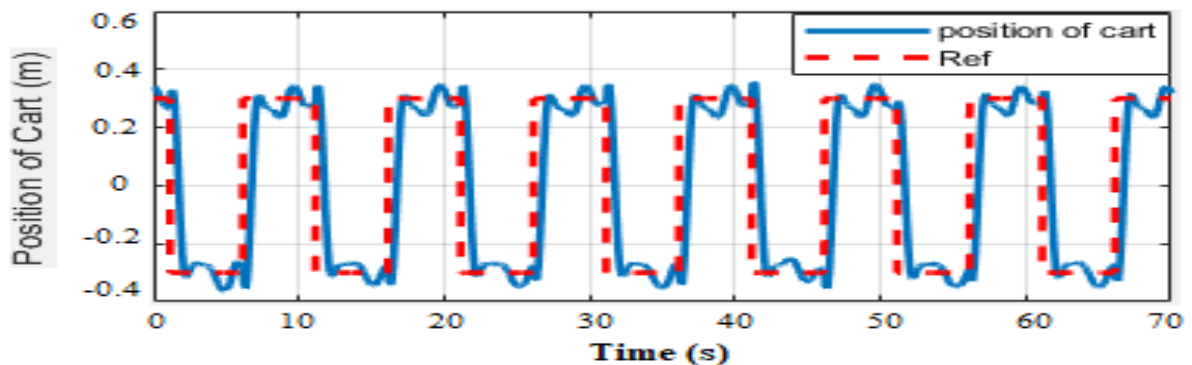
Trong hai trường hợp này, bộ điều khiển trượt vẫn chứng minh khả năng của mình, bám rất tốt quỹ đạo và con lắc vẫn được giữ ở vị trí cân bằng (đoạn vọt lên khi chuyển vị trí nhưng vẫn đảm bảo quay về ngay và không bị rớt).



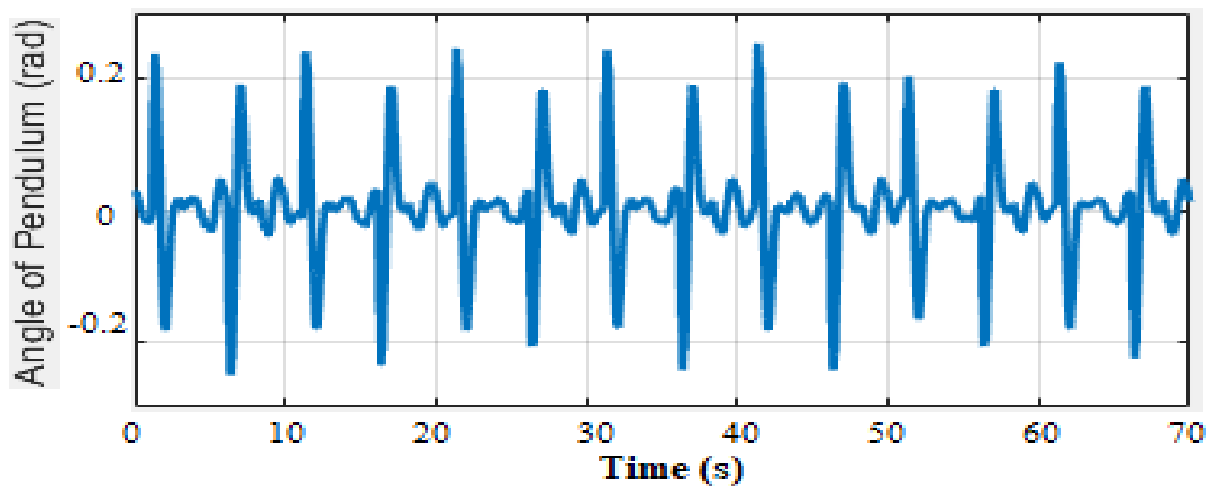
Hình 64. Vị trí xe so với Reference (m) - Quỹ đạo xung vuông chu kỳ 20s



Hình 65. Góc con lắc trong quỹ đạo xung vuông (rad) – chu kỳ 20s



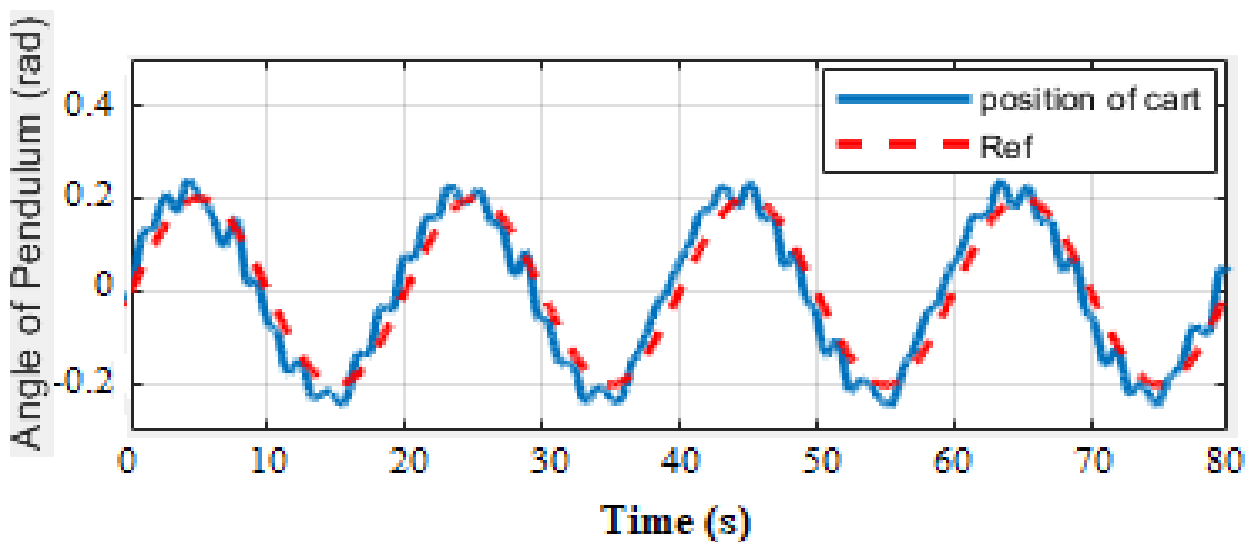
Hình 66. Vị trí xe so với Reference (m) - Quỹ đạo xung vuông chu kỳ 10s



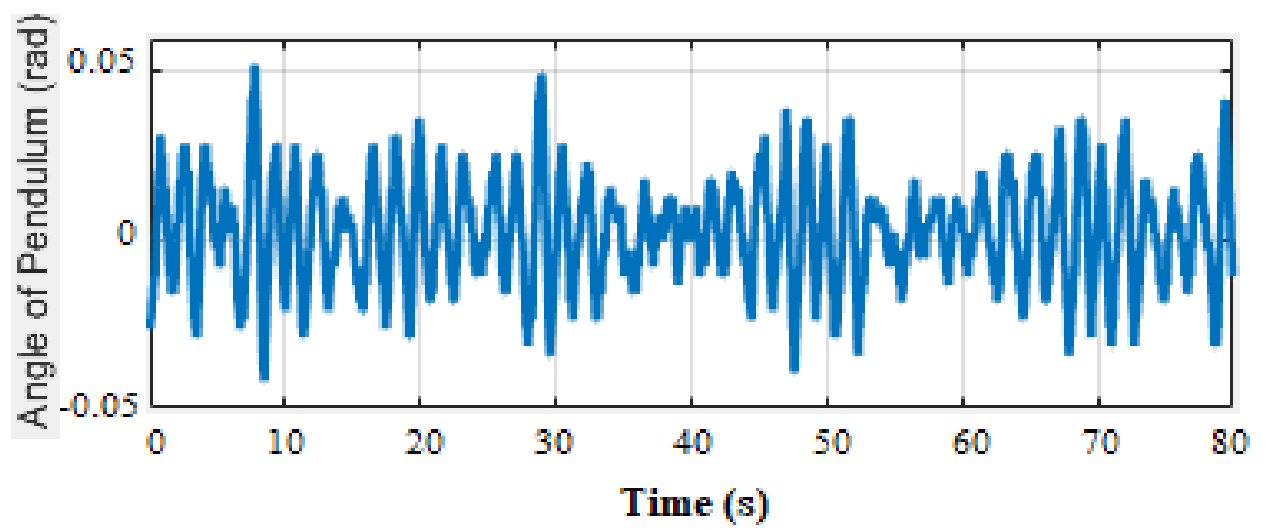
Hình 67. Góc con lắc trong quỹ đạo xung vuông (rad) – chu kỳ 10s

5.2.3 Quỹ đạo sóng sine

Giống như hai trường hợp được đề cập trên, trong quỹ đạo sóng sine, giải thuật trượt kết hợp linear regression vẫn làm việc rất ổn định. Kết quả là trong Hình 68 và Hình 69, quỹ đạo sóng sine được bám rất mượt mà cũng như con lắc lúc nào cũng được đảm bảo giữ ở vị trí TOP.



Hình 68. Vị trí xe so với Reference (m) - Quỹ đạo sóng sine chu kỳ 20s



Hình 69. . Góc con lắc trong quỹ đạo sóng sine (rad) – chu kỳ 10s

Chương 6: Kết luận và hướng phát triển

6.1 Kết quả đạt được

Sau khi hoàn thành đề tài luận văn tốt nghiệp, sinh viên đã hoàn thành được các kết quả như sau:

- Xây dựng được phương trình toán học của hệ Cart and Pole với điện áp là tín hiệu điều khiển.
- Xây dựng bộ điều khiển trượt bám quỹ đạo hình sine và xung vuông cũng như cân bằng tại chỗ cho hệ cả trên mô phỏng lẫn thực nghiệm.
- Dựa vào Linear Regression, đưa ra phương pháp giảm chattering ở điều khiển trượt.
- Nêu ra phương pháp và cách thức nhận diện thông số động cơ.
- Dùng giải thuật di truyền [8] để ứng dụng tìm ra thông số giải thuật trượt để tối ưu giải thuật.
- Xây dựng thành công mô hình thí nghiệm trên thực tế của hệ con lắc ngược trên xe.
- Điều khiển hệ con lắc ngược trên xe thành công trên cả mô phỏng lẫn thực nghiệm.
- Xây dựng Winform giao tiếp real-time với STM32F4 để quan sát quỹ đạo đường đi của hệ thống.

6.2 Hướng phát triển

Dù đã đạt được những kết quả khả quan như trên, song, vẫn còn nhiều vấn đề hạn chế cần khắc phục trong đề tài nghiên cứu này:

Các vấn đề như sau:

- Vì những hạn chế về kinh tế, nên mô hình được sinh viên xây dựng còn hạn chế về nhiều mặt, như thông số động cơ, vì sử dụng động cơ được mua cũ lại, nên những thông số bên trong của động cơ không được cung cấp, dù đã

được nhận diện bằng phương pháp nhận diện động cơ tuy vậy với nhiễu xung quanh, các thông số trên chỉ là đúng tương đối chứ không chính xác như thông số sản xuất. Dẫn đến việc điều khiển vẫn còn dao động nhẹ.

- Do thiết kế khá dài và con lắc được thiết kế khá nặng và to, bộ điều khiển không thích hợp cho điều khiển swing-up.
- Với giới hạn về kinh tế, Encoder sinh viên sử dụng trong đề tài này có độ phân giải không cao, dẫn đến việc cân bằng kinh tế tại chỗ còn dao động, không hoàn toàn đứng im tại vị trí không.
- Ma sát trong mô hình chưa được tính đến trong phương trình toán học hệ thống, đây cũng là một phần lý do quá trình điều khiển chưa đạt được đến mức tốt nhất.

Hướng phát triển:

Phương trình toán học bao gồm cả ma sát, nhiễu, giúp cho việc điều khiển thích nghi và tối ưu hơn cho hệ thống.

Gắn thêm link 2 trên con lắc ngược (hiện nay trên nhiều lab trên thế giới đã bắt đầu cho thêm 1 link vào hệ thống, để làm phức tạp mô hình lên). Từ đó, các nghiên cứu cho hệ thống SIMO bậc cao, có cơ cấu phức tạp được tiến hành. Về mô hình của sinh viên, để tiến hành được mô hình này cần cải tiến thêm về độ phân giải Encoder, thay đổi bọt đạn trong xe để giảm bớt ma sát (vì hệ thống đã xây dựng và chạy thử nhiều lần nên đã mòn và hệ có nhiều ma sát).

Thử một giải thuật tiên tiến hơn dựa trên giải thuật trượt nền tảng, vd: Trượt thích nghi, Trượt có kết hợp Neural Network để có thể tự nhận diện thông số mô hình (Thay đổi một thông số trong mô hình ví dụ cân nặng con lắc, hệ vẫn giữ được tính chất điều khiển ổn định, thích nghi ngay với cân nặng mới).

Thử nghiệm nhiều quỹ đạo phức tạp khác nhau.

Thay đổi để, làm gọn mô hình hơn để thực hiện điều khiển Swing up cho mô hình.

Tài liệu tham khảo

- [1] Quanser Linear Servo Control Systems Brochure – online (pdf).
- [2] Mingjun Zhang và Tzyh - Jong Tarn, Hybrid control of the Pendubot, IEEE/ASME Transactions on Mechatronics, 7(1), 79-86, 2002.
- [3] Mojtaba Ahmadi Khamesar, Mohammad Teshnehlab, Mahdi Aliyari Shoorehdeli, Sliding mode control of Rotary Inverted Pendulum, 2007 Mediterranean Conference on Control & Automation.
- [4] Jia – Jun Wang, *Simulation studies of inverted pendulum based on PID controllers*, Simulation Modelling Practice and Theory, 2011.
- [5] Tran Vinh Toan, Tran Thu Ha và Tran Vi Do, Hybrid Control for Swingup and Balancing Pendubot System: An Experimental Result, 2017 International Conference on System Science and Engineering (ICSSE).
- [6] Rong Xu và Umit Ozguner, *Sliding Mode Control of a class of underactuated systems*, Automatica, 2008.

- [7] Jing – Jing Xiong, Guobao Zhang, Sliding mode control for a quadrotor UAV with parameter uncertainties, 2016 2nd International Conference on Control, Automation and Robotics (ICCAR).
- [8] Seung – Hi Lee và Chung Choo Chung, Predictive Control with Sliding Mode for Autonomous Driving Vehicle Lateral Maneuvering, 2017 American Control Conference.
- [9] Chaima Bensaci, Youcef Zennir, Dennis Pomorski, Nonlinear Control of a differential wheeled mobile robot in real time-Turtlebot 2, Third International Conference on Technological Advances in Electrical Engineering (ICTAEE), 2018.
- [10] Jie-Ren Hong (2003), *Balance control of a cart-pole inverted pendulum*, Luận văn Thạc Sĩ Trường Đại học National Cheng Kung, Đài Loan.
- [11] *Điều khiển Pendubot dùng kỹ thuật trượt*, <https://text.xemtailieu.com/tai-lieu/dieu-khien-pendubot-dung-ky-thuat-dieu-khien-truot-857713.html>
- [12] Yinxing Hao, Jianqiang Yi, Dongbin Zhao và Dianwei Qian, *Incremental Sliding Mode Control for Large-Scale Underactuated System*, Proceedings of the 2007 IEEE International Conference on Networking, Sensing and Control, London, UK, 15-17 April 2007.
- [13] Yinxing Hao, , Jianqiang Yi, Dongbin Zhao và Wei Wang, Proposal of incremental sliding mode control, Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06), 2006.
- [14] Awais Shah, Deqing Huang, Tianpeng Huang và Umar Farid, *Optimization of Buildings Energy Consumption by Designing Sliding Mode Control for Multizone VAV Air Condition Systems*, Energies Article.
- [15] Vũ Hữu Tiệp, *Bài 3: Linear Regression*, Sách Machine Learning cơ bản.

[16] ThS. Trần Kim Hương, ThS. Nguyễn Thị Ngọc Chi, Giải thuật di truyền và các ứng dụng, Hội nghị nghiên cứu khoa học Khoa SP Toán – Tin Đại học Đồng Tháp.

[17] PGS.TS Huỳnh Thái Hoàng, Chương 02: Chương trình mẫu GA chỉnh định tối ưu thông số bộ điều khiển PID dùng phương pháp mã hoá thập phân, Sách Điều khiển thông minh – Nhà xuất bản Đại học Quốc Gia Thành phố Hồ Chí Minh.

[18] Hồng Gia Bảo, Phạm Hồng Quân, Đồ án 1 – Vi Xử Lý.

[19] Vũ Đình Đạt, Luận văn Tốt nghiệp – Xây dựng bộ điều khiển trượt tối ưu cho hệ Pendubot bám quỹ đạo, Đại học Sư Phạm Kỹ Thuật TPHCM.

[20] Link download thư viện: <http://wajung.aimagin.com/>, Link hướng dẫn: https://www.youtube.com/watch?v=aEh1S_XN87w

[21] Hồng Gia Bảo, Mircea Nitulescu, Ionel Cristian Vladu, Nguyễn Minh Tâm, Lê Thị Thanh Hoàng, Nguyễn Phong Lưu, Trương Thanh Liêm, Nguyễn Văn Đông Hải, Huỳnh Xuân Dũng, *Trajectory Tracking Sliding Mode Control for Cart and Pole System*, Intelligent Computing in Engineering – select Proceedings of RICE 2019 (Research in Intelligent Computing in Engineering 2019).

[1'] https://en.wikipedia.org/wiki/Lyapunov_stability.

[2'] Nguyễn Đức Minh, Điều khiển trượt thích nghi hệ thống phi tuyến, Luận Án Tiến Sĩ Kỹ Thuật - Đại học Bách Khoa Thành Phố Hồ Chí Minh.

[3'] TS. Dương Hoài Nghĩa, trang 172 – 174 – Sách Điều khiển hệ thống đa biến, Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh.

