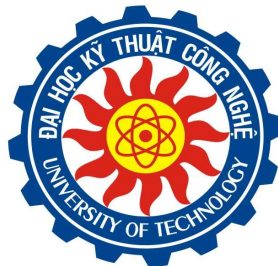


**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ TP. HCM**

---



**NGUYỄN QUỐC HOÀNG**

# **ĐIỀU KHIỂN HỆ PENDUBOT TRÊN NỀN LABVIEW**

**LUẬN VĂN THẠC SĨ**

**Chuyên ngành : KỸ THUẬT ĐIỆN**

**Mã số ngành: 60520202**

**TP. HỒ CHÍ MINH, tháng 04 năm 2013.**

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH TẠI  
TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHỆ TP. HCM**

Cán bộ hướng dẫn khoa học : TS. NGUYỄN THANH PHƯƠNG

*(Ghi rõ họ, tên, học hàm, học vị và chữ ký)*

Luận văn Thạc sĩ được bảo vệ tại Trường Đại học Kỹ thuật Công nghệ  
TP. HCM ngày 10 tháng 05 năm 2013.

Thành phần Hội đồng đánh giá Luận văn Thạc sĩ gồm:

*(Ghi rõ họ, tên, học hàm, học vị của Hội đồng chấm bảo vệ Luận văn Thạc sĩ)*

**1. TS. NGÔ CAO CƯỜNG**

**2. TS. NGUYỄN HÙNG**

**3. PGS.TS TRẦN THU HÀ**

**4. TS. ĐINH HOÀNG BÁCH**

**5. TS. HUỲNH CHÂU DUY**

Xác nhận của Chủ tịch Hội đồng đánh giá Luận sau khi Luận văn đã được  
sửa chữa (nếu có).

Chủ tịch Hội đồng đánh giá LV

TRƯỜNG ĐH KỸ THUẬT CÔNG NGHỆ TP. HCM  
**PHÒNG QLKH - ĐTSĐH**

**CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**Độc lập - Tự do - Hạnh phúc**

*TP. HCM, ngày..... tháng..... năm 20.....*

## **NHIỆM VỤ LUẬN VĂN THẠC SĨ**

Họ tên học viên: **NGUYỄN QUỐC HOÀNG**

Giới tính: NAM

Ngày, tháng, năm sinh: 04/11/1981

Nơi sinh: ĐỒNG NAI

Chuyên ngành: KỸ THUẬT ĐIỆN

MSHV: 1181031016

**I- TÊN ĐỀ TÀI: ĐIỀU KHIỂN HỆ PENDUBOT TRÊN NỀN LABVIEW**

**II- NHIỆM VỤ VÀ NỘI DUNG:**

- Nghiên cứu phương pháp điều khiển cân bằng hệ Pendubot
- Xây dựng mô hình Pendubot
- Thiết kế bộ điều khiển LQR và PID điều khiển hệ Pendubot
- Thực hiện mô phỏng và thực nghiệm đánh giá kết quả.

**III- NGÀY GIAO NHIỆM VỤ: 21/06/2012**

**IV- NGÀY HOÀN THÀNH NHIỆM VỤ: 09/04/2013**

**V- CÁN BỘ HƯỚNG DẪN: TS. NGUYỄN THANH PHƯƠNG**

**CÁN BỘ HƯỚNG DẪN**

**KHOA QUẢN LÝ CHUYÊN NGÀNH**

(Họ tên và chữ ký)

(Họ tên và chữ ký)

## **LỜI CAM ĐOAN**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi. Các số liệu, kết quả nêu trong Luận văn là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Tôi xin cam đoan rằng mọi sự giúp đỡ cho việc thực hiện Luận văn này đã được cảm ơn và các thông tin trích dẫn trong Luận văn đã được chỉ rõ nguồn gốc.

**Tp. Hồ Chí Minh, ngày 01 tháng 11 năm 2012.**

**Ký tên**

**Nguyễn Quốc Hoàng.**

## LỜI CẢM ƠN

Sau một thời gian học tập và nghiên cứu tại trường, nay học viên đã hoàn thành đề tài tốt nghiệp cao học của mình. Để có được thành quả này, học viên đã nhận được rất nhiều sự hỗ trợ và giúp đỡ tận tình từ thầy cô, gia đình, cơ quan và đồng nghiệp và bạn bè .

Học viên xin trân trọng bày tỏ lòng biết ơn sâu sắc, chân thành đến thầy TS. Nguyễn Thanh Phương, người đã tận tình trực tiếp hướng dẫn học viên thực hiện hoàn thành luận văn này.

Học viên cũng xin chân thành đến quý thầy cô trường Đại học Kỹ Thuật Công Nghệ Thành phố Hồ Chí Minh đã trang bị cho học viên một lượng kiến thức rất bổ ích, đặc biệt học viên cũng chân thành cảm ơn quý thầy cô Khoa Điện – Điện tử đã tạo mọi điều kiện thuận lợi nhất cho học viên trong quá trình học tập cũng như nghiên cứu và làm luận văn này.

Học viên xin gửi lời cảm ơn chân thành nhất đến đồng nghiệp, đại gia đình học viên, bạn bè đã giúp đỡ cho học viên rất nhiều trong quá trình hoàn thành luận văn này.

**Xin chân thành cảm ơn!**

**Tp. Hồ Chí Minh, ngày 01 tháng 11 năm 2012.**

(Ký tên và ghi rõ họ và tên)

**Nguyễn Quốc Hoàng.**

## TÓM TẮT LUẬN VĂN

Hệ thống Pendubot là hệ thống không ổn định, phi tuyến mức cao. Nó được sử dụng như một mô hình phổ biến cho các ứng dụng trong điều khiển tuyến tính và phi tuyến. Luận văn này trình bày khái quát về cấu trúc vật lý, mô hình động học của hệ thống Pendubot, với mong muốn thiết kế xây dựng bộ điều khiển Hệ Pendubot bằng phương pháp điều khiển toàn phương tuyến tính LQR (Linear Quadratic Regulation), và bộ điều khiển PID để so sánh thời gian ổn định bền vững nhất. Với bộ điều khiển PID thông qua phần mềm LabVIEW để điều khiển thực nhưng có thể thích nghi nếu có bất kỳ thông số nào thay đổi xảy ra trong hệ thống.

Phần cuối của luận văn trình bày việc thiết kế phần cứng mô hình thật hệ thống Pendubot, thiết lập bộ điều khiển PID và PID cải tiến để ổn định hệ thống mô hình thực trong môi trường LabVIEW với độ ổn định theo thời gian cao nhất, kết nối thông qua card thu nhập từ liệu USB 9090 do Hocdelam.group. Và mô phỏng mô hình thật 3D qua môi trường LabVIEW.

## **ABSTRACT**

Inverted Pendubot system is unstable, highly nonlinear. It is used as a common model for engineering applications in linear and nonlinear control. This project presents an overview of the physical structure, dynamic model of rotating inverted Pendubot system, with the desire to design the control system construction method Pendubot linear quadratic control LQR (Linear Quadratic Regulation), and PID controller to compare the time most solid stable surface control methods balance Pendubot by PID improving the stability balance Pendubot. With PID controller through LabVIEW software to control, but can be adapted if any parameters change occurs in the system.

The last part of the thesis presents the hardware design of model so system Pendubot, set the PID controller and PID improvements to system stability model in LabVIEW environment with highest stability over time, connected via USB card income from 9090 by Hocdelam.group. and 3D model simulate it through the LabVIEW environment.

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN.....	1
1.1 Giới thiệu .....	1
1.2 Giới thiệu về mô hình Pendubot.....	2
1.3 Mục tiêu và giới hạn của đề tài.....	4
1.4 Phương pháp nghiên cứu: .....	5
1.5 Tổng quan chung về lĩnh vực nghiên cứu .....	6
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT .....	10
2.1. Lý thuyết điều khiển tự động.....	10
2.1.1 Khái niệm.....	10
2.1.1.1 Điều khiển .....	10
2.1.1.2.Hệ thống điều khiển .....	10
2.1.1.3. Bộ điều khiển .....	11
2.1.1.4. Các nguyên tắc điều khiển .....	11
2.2 Lý thuyết thuật toán điều khiển PID (Proportional Integral Derivative) .....	13
2.2.1. Cơ bản về vòng điều khiển .....	13
2.2.2. Giới thiệu bộ điều khiển PID .....	15
2.2.2.1. Khâu P .....	16
2.2.2.2. Khâu I.....	17
2.2.2.3. Khâu D .....	18
2.2.2.4. Tổng hợp ba khâu – Bộ điều khiển PID.....	19
2.2.2.5. Rời rạc hóa bộ điều khiển PID:.....	19
2.2.3. Điều chỉnh vòng lặp.....	21



2.2.3.1 Độ ổn định.....	21
2.2.3.2 Tối ưu hóa hành vi .....	22
2.2.3.3 Tổng quan các phương pháp .....	22
2.2.3.4 Điều chỉnh thủ công .....	23
2.2.3.5 Phương pháp Ziegler–Nichols .....	24
2.4 Các bước giải phương trình tối ưu.....	26
<b>CHƯƠNG 3: PHƯƠNG TRÌNH TOÁN HỌC HỆ PENDUBOT.....</b>	<b>28</b>
3.1. Hệ thống Pendubot .....	28
3.2 Cơ sở chung .....	29
3.3 Phương trình toán hệ thống .....	30
3.3.1 Giới thiệu các thông số của hệ thống.....	30
3.3.2 Phương trình toán .....	32
3.4 Giới thiệu phương pháp nhận dạng thông số.....	36
3.4.1 Phương pháp năng lượng. ....	36
3.4.2 Phương pháp đo trực tiếp các thông số của hệ thống .....	39
3.5 Diễn tả không gian trạng thái hệ thống.....	40
3.6 Những điểm cân bằng của hệ thống .....	43
3.7 Tính chất có thể điều khiển được và không điều khiển được.....	45
<b>CHƯƠNG 4: THIẾT KẾ BỘ ĐIỀU KHIỂN .....</b>	<b>47</b>
4.1 Thiết kế điều khiển hệ thống Pendubot dùng giải thuật LQR .....	47
4.1.1 Tuyến tính hóa phương trình chuyển động.....	47
4.1.2 Điểm cân bằng vị trí MID:.....	49
4.1.3 Điểm cân bằng vị trí TOP .....	53

4.2 Kết quả mô phỏng vị trí MID dùng LQR .....	55
4.3 Kết quả mô phỏng vị trí TOP dùng LQR .....	59
4.4 Kết quả mô phỏng vị trí MID dùng PID.....	64
4.4 Kết quả mô phỏng vị trí TOP dùng PID.....	68
CHƯƠNG 5: THIẾT KẾ PHẦN CỨNG HỆ PENDUBOT .....	74
5.1 Giới thiệu phần cứng hệ thống Pendubot: .....	74
5.1.1 Phần cơ khí .....	75
5.1.2 Phần mạch điện.....	76
5.2 Phần thu nhập dữ liệu thông qua card USB 9090 .....	79
CHƯƠNG 6: ỨNG DỤNG LABVIEW ĐIỀU KHIỂN HỆ PENDUBOT .....	82
6.1 Thực nghiệm và khảo sát chương trình điều khiển hệ Pendubot .....	82
6.1.1 Lưu đồ giải thuật điều khiển hệ Pendubot trên nền Labview .....	82
6.1.2 Trang giao diện chính của chương trình.....	87
6.1.3 Chuẩn bị thực nghiệm và khảo sát chương trình nội dung thực hiện.....	87
6.2 Nhận dạng và điều khiển hệ Pendubot trên nền Labview .....	89
6.2.1 Khởi tạo hệ thống .....	89
6.2.2 Giá trị đặt, Trượt hóa giá trị đặt.....	93
6.2.3 Tính sai lệch hệ thống và nhận giá trị Kp:.....	94
6.2.4 Nhập giá trị Kp: .....	95
6.2.5 Tính khâu P:.....	95
6.3.6 Đóng băng “Chức năng tích phân “ .....	99
6.3.7 Tính khâu PI.....	100
6.4 Kết luận.....	105

CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	107
7.1 Kết quả đạt được.....	107
7.2 Hạn chế .....	107
7.3 Hướng phát triển của đề tài .....	107
7.4 Kết luận: .....	108
TÀI LIỆU THAM KHẢO.....	109

## **DANH MỤC CÁC BẢNG, BIỂU ĐỒ, ĐỒ THỊ, HÌNH ẢNH**

Hình 1.1: Hệ thống Pendubot.....	3
Hình 2. 1: Mô tả hệ thống điều khiển .....	10
Hình 2. 2: Sơ đồ điều khiển bù nhiễu.....	11
Hình 2. 3: Sơ đồ điều khiển cân bằng sai lệch. ....	12
Hình 2. 4: Sơ đồ điều khiển phối hợp. ....	12
Hình 2. 5: Sơ đồ điều khiển thích nghi .....	13
Hình 2. 6: Sơ đồ hệ thống điều khiển dùng PID .....	15
Hình 2. 7: Đáp ứng của khâu P .....	16
Hình 2. 8: Đáp ứng của khâu I và PI.....	17
Hình 2. 9 : Đáp ứng của khâu D và PD.....	18
Hình 2. 10:Đáp ứng của khâu P, PI và PID .....	19
Hình 2. 11: Sơ đồ khối PID.....	20
Bảng 2.1: Lựa chọn phương pháp điều chỉnh .....	23
Bảng 2.2: Tác động của việc tăng một thông số độc lập .....	24
Hình 3.1: Hệ thống Pendubot.....	28
Hình 3.2: Hệ trục tọa độ Pendubot.....	30
Hình 3.3 Cơ cấu chấp hành và cơ cấu chấp hành bên dưới của Pendubot .....	44
Hình 5.1 : Vị trí cân bằng MID .....	49
Hình 4.2 Vị trí cân bằng Top .....	53
Hình 4.3: Sơ đồ khối mô phỏng vị trí Mid.....	55
Hình 4.4: Sơ đồ subsystem1 vị trí Mid .....	55
Hình 4.5: Mô phỏng tín hiệu ra của $q_1$ .....	56

Hình 4.7: Tín hiệu ra của góc $q_1$ .....	57
Hình 4.11: Tín hiệu ra của $\dot{q}_2$ .....	59
Hình 4.13: Sơ đồ subsystem2 vị trí Top .....	60
Hình 4.14: Tín hiệu ra của $q_1$ .....	61
Hình 4.15: Tín hiệu ra của $q_2$ .....	61
Hình 4.16: Tín hiệu ra của $q_1$ .....	61
Hình 4.19: Tín hiệu ra của $\dot{q}_1$ .....	63
Hình 4.21: Sơ đồ khối mô phỏng vị trí Mid-PID .....	64
Hình 4.22: Sơ đồ subsystem1 vị trí Mid-PID .....	64
Hình 4.23: Mô phỏng tín hiệu ra của $q_1$ .....	65
Hình 4.25: Tín hiệu ra của góc $q_1$ .....	66
Hình 4.27 Tín hiệu ra moment $T_m$ .....	67
Hình 4.28: Tín hiệu ra $\dot{q}_1$ .....	67
Hình 4.29 Tín hiệu ra của $\dot{q}_2$ .....	68
Hình 4.31 Sơ đồ subsystem2 vị trí Top PID .....	69
Hình 4.33 Tín hiệu ra của $q_2$ .....	70
Hình 4.34 Tín hiệu ra của $q_1$ .....	70
Hình 4.35 Tín hiệu ra của $q_2$ .....	71
Hình 4.37 Tín hiệu ra của $\dot{q}_1$ .....	72
Hình 5.1 Sơ đồ khối hệ thống điều khiển hệ Pendubot.....	74
Hình 5.3. Phần cơ khí mô hình hệ Pendubot .....	75
Hình 5.4 Mạch động lực 3A-DCS HIPO .....	76

Mô tả: .....	77
Hình 5.6 Động cơ DC .....	77
Hình 5.7 Sơ đồ điện đấu nối hai mạch giao tiếp máy tính PC điều khiển Hệ Pendubot. ....	78
Hình 5.8 Card USB 9090 .....	79
Hình 5.9 Sơ đồ chân Card USB 9090 .....	80
Bảng 5.1 Bảng các thông số của Card USB HDL 9090 .....	81
Hình 6.1 Sơ đồ giải thuật tính khâu .....	83
Hình 6.3 Sơ đồ giải thuật điều khiển hệ Pendubot bằng thuật toán PI .....	85
Hình 6.4 Chương trình điều khiển Hệ Pendubot trên nền Labview.....	86
Hình 6.5 Giao diện điều khiển hệ Pendubot trên nền Labview .....	87
Hình 6.6 Biểu diễn Link 2 cân bằng thẳng đứng ( $\theta_2 = 0^\circ$ ) ở vị trí bất kỳ đối với Link 1 thay đổi vị trí bất kỳ ( $\theta_1 \neq 0^\circ$ ). ....	89
Hình 6.7 Giao diện Khởi tạo hệ thống .....	89
Hình 6.8 Thuật toán của chương trình labview thực hiện việc khởi tạo hệ thống....	90
Hình 6.9 Thuật toán tính giá trị H.....	91
Hình 6.10 Đọc giá trị Encoder E2.....	92
Hình 6.11 Giải thuật Hình Giải thuật khởi tạo ra giá trị encoder 1 $E1^*$ .....	92
Hình 6.12 Giải thuật khởi tạo ra giá trị encoder 2 $E2^*$ .....	93
Hình 6.13 Chương trình thuật toán labview thực hiện trượt hóa giá trị đặt.....	93
Hình 6.14 Thuật toán trượt hóa giá trị đặt $E2^*$ .....	94
Hình 6.15 Thuật toán tính sai lệch hệ thống e .....	95
Hình 6.16 Giao diện chính điều khiển khâu P .....	95

Hình 6.17 Chương trình thuật toán labview thực hiện tính Khâu P .....	96
Hình 6.18 Chương trình thuật toán labview thực hiện tính khâu I .....	97
Hình 6.19 Giải thuật thời gian lấy mẫu.....	98
Hình 6.20 Giao diện nhập dữ liệu nhập giá trị $K_i$ và tính Khâu I.....	98
Hình 6.21 Giải thuật trình bày phương pháp tính toán khâu I. ....	98
Hình 6.22 Khối Giải thuật trình bày phương pháp đóng băng tích phân.....	99
Hình 6.23 Hàm In range coerce (trong khoảng) .....	99
Hình 6.24 Khối Giải thuật trình bày tính khâu PI.....	100
Hình 6.25 Kết quả Đáp ứng giá trị đặt khi chưa bù mômen xoay. ....	100
Hình 6.26 Kết quả đáp ứng khi chưa có Bù Momen xoay, $K_i$ hiệu chỉnh lớn.....	100
Hình 6.27 biểu diễn góc lệch của Link 1 so với phương OY là $\beta$ . ....	103
Hình 6.28 Sơ đồ điều khiển hệ Pendubot bằng phương pháp bù mômen xoay ....	102
Hình 6.29 Giải thuật trình bày phương pháp bù mômen xoay phi tuyến .....	103
Hình 6.30 Đáp ứng có Bù Momen xoay, $K_i$ hiệu chỉnh nhỏ.....	104
Hình 6.31 đáp ứng góc $\theta_1 \cong 45^\circ$ , $\theta_2 \cong 0^\circ$ .....	104
Hình 6.32 Kết quả đáp ứng điện áp ngõ ra góc $\theta_1 \cong 45^\circ$ .....	104
Hình 6.33 Đáp ứng góc $\theta_1 \cong 90^\circ$ , $\theta_2 \cong 0^\circ$ .....	105
Hình 6.34 Kết quả đáp ứng điện áp ngõ ra góc $\theta_1 \cong 90^\circ$ .....	105

# CHƯƠNG 1: TỔNG QUAN

## 1.1 Giới thiệu

Trong những thập niên gần đây khái niệm về robot và chế tạo robot để phục vụ nhu cầu sản xuất trong môi trường công nghiệp. Gần đây hàng loạt robot đã được sản xuất với những mục đích và những nhu cầu khác nhau. Roboties là một môi trường rất rộng lớn như thiết kế máy, phân tích hoạt động và điều khiển, xử lý tín hiệu và hình ảnh, giải thuật tính toán. Hay nói một cách đơn giản, Roboties được hiểu là một ngành khoa học có nhiệm vụ nghiên cứu thiết kế chế tạo các robot và ứng dụng chúng trong các hoạt động khác nhau của xã hội loài người, như nghiên cứu khoa học-kỹ thuật, kinh tế, quốc phòng và dân sinh.

Cùng với sự phát triển của xã hội, khoa học kỹ thuật, các phương pháp điều khiển hiện đại, điều khiển thông minh, ra đời. Các phương pháp này ngày càng được nghiên cứu, phát triển, ứng dụng rộng rãi. Chúng tỏ ra có nhiều ưu thế vượt trội so với các phương pháp điều khiển kinh điển trước đây nhất là trong lĩnh vực điều khiển đối tượng phi tuyến. Điều khiển mờ (Fuzzy control) là một trong những phương pháp điều khiển thông minh, có hiệu quả để điều khiển đối tượng phi tuyến. Kỹ thuật điều khiển này với ưu điểm điều khiển đưa vào những thông tin không rõ ràng, dùng kinh nghiệm của người thiết kế để điều khiển nhưng đối tượng phi tuyến không xác định được mô hình toán. điều khiển trượt là hệ thống điều khiển mà thông số và cấu trúc của bộ điều khiển thay đổi trong quá trình vận hành Pendubot nhằm giữ vững chất lượng của hệ thống có sự hiện diện của những thông số bất định hoặc biến đổi không biết trước trong thông số hệ thống. Thực tế, đối tượng luôn tồn tại nhưng thông số bất định như: nhiễu, ảnh hưởng của điều kiện làm việc... Pendubot là một đối tượng đã có rất nhiều nghiên cứu, khảo sát. Đây là những đối tượng thường được các nhà nghiên cứu lựa chọn để kiểm chứng những thuật toán điều khiển của mình từ những thuật toán điều Khiển cổ điển đến những thuật toán điều khiển thông minh.



Các phương pháp điều khiển mới cho phép thiết kế bộ điều khiển Pendubot khiến trực tiếp không qua nhận dạng trước mô hình như : Điều khiển dựa trên cơ sở luật mờ Điều khiển dùng phương pháp giải thuật di truyền hoặc học tăng cường để tìm luật điều khiển tối ưu.

Từ những hiểu biết sơ bộ về chức năng và kết cấu của robot, chúng ta hiểu robotics là một liên ngành, gồm cơ khí, điện tử, kỹ thuật điều khiển và công nghệ thông tin. Theo thuật ngữ hiện nay, robot là sản phẩm của ngành cơ điện tử (Machatronic).Khía cạnh nhân văn và khía cạnh khoa học - kỹ thuật của việc sản sinh ra robot thống nhất ở một điểm: Thực hiện hoài bão của con người, là tạo ra các thiết bị thay thế mình, trong những hoạt động không thích hợp với mình như:

Trong môi trường khắc nghiệt hoặc nguy hiểm: Như ngoài không gian vũ trụ, trên chiến trường, nơi có phóng xạ,...

Công việc lặp đi lặp lại nhàm chán, nặng nhọc: vận chuyển nguyên vật liệu, lau cọ nhà cửa,...

Những việc đòi hỏi có độ chính xác cao như thông tắc mạch máu hoặc các ống dẫn trong cơ thể, lắp ráp các cấu tử trong vi mạch,...

Lĩnh vực ứng dụng robot là rất rộng và ngày càng được mở rộng thêm. Ngày nay, khái niệm về robot đã mở rộng hơn khái niệm nguyên thủy rất nhiều. Sự phỏng tác về kết cấu, chức năng, dáng vẻ của con người là cần thiết nhưng không còn ngự trị trong robot nữa. Kết cấu của nhiều “con” robot khác xa với kết cấu các bộ phận cơ thể con người và cũng có thể thực hiện những việc vượt xa khả năng con người.

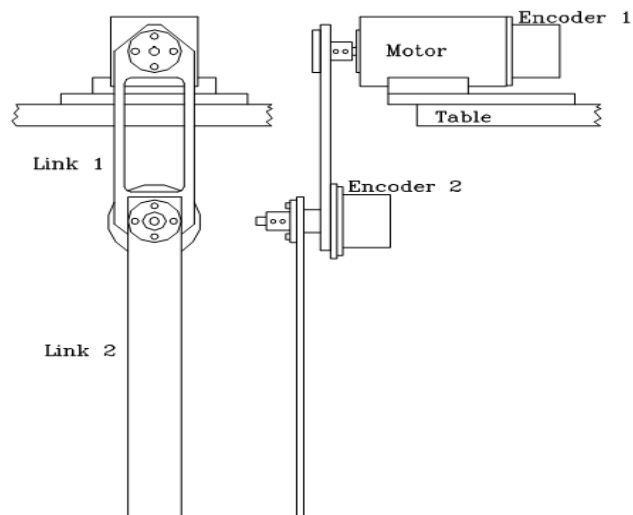
## **1.2 Giới thiệu về mô hình Pendubot**

Ngày nay những yêu cầu về những sản phẩm mới và tự động hóa trong các xí nghiệp nhà máy là rất cần thiết. Trên thế giới cũng như trong nước đã và đang nghiên cứu đưa các robot vào trong các nhà máy sản xuất.

Trong luận văn này, tôi tìm hiểu về hệ thống Pendubot (hay gọi là con lắc ngược hai bậc tự do)

Mô hình Pendubot được phát triển từ mô hình con lắc ngược, là hệ thống máy có cơ cấu chấp hành bên dưới (underactuated), gồm hai khâu máy hay gọi là hai link, link1 và link2. Hai link này được điều khiển từ vị trí cân bằng ổn định bên dưới sau đó được chuyển sang vị trí cân bằng mới cân bằng không ổn định. Ở mô hình này có nhiều vị trí cân bằng khác nhau.

Tại trường đại học Illinois, việc nghiên cứu rất rộng và phát triển đã có khái niệm về thiết kế robot hai link khởi động bên dưới trên mặt phẳng được gọi là Acrobot. Để mở rộng khái niệm trên ta có hệ thống gồm hai link gọi là Pendubot (Hình 1.1)



Hình 1.1: Hệ thống Pendubot

Hệ thống Pendubot gồm hai link, link1 gắn ở trục motor DC và ở đuôi motor gắn một encoder. Link2 nối kết đến link1 tại khớp chung và tại đây gắn 1 encoder. Những encoder tại những điểm nối cung cấp phép đo vị trí góc mỗi liên kết. Hệ thống PENDUBOT được tạo ra bằng từ ghép (Pendulum và robot) như hình bên dưới (Hình 1.1 ). Hệ thống Pendubot này hoạt động tại vai (shoulder) của link 1 và không hoạt động tại khớp (elbow) link 2. Link 2 này có thể dao động tự do xung quanh link1 và nó hoạt động nhờ vào moment quán tính chuyển động của link1 được gắn trực tiếp với trục động cơ servo .

Để điều khiển cân bằng hệ thống này là rất khó khăn. Mô hình phải được thiết kế chính xác, giải thuật điều khiển phải thật tốt mới có thể điều khiển cân bằng được hệ thống.

Có nhiều phương pháp điều khiển hệ thống này, như là điều khiển tối ưu, điều khiển thích nghi, hay là điều khiển thích nghi bền vững, điều khiển fuzzy ... Vì vậy học viên chọn đề tài nghiên cứu là “ **Điều khiển hệ Pendubot trên nền LABVIEW**” với mong muốn thiết kế xây dựng bộ điều khiển Hệ Pendubot bằng phương pháp điều khiển toàn phương tuyến tính LQR (Linear Quadratic Regulation), và bộ điều khiển PID để so sánh thời gian ổn định bền vững nhất.

### 1.3 Mục tiêu và giới hạn của đề tài

Trong luận văn này mục tiêu là nghiên cứu lập trình bằng ngôn ngữ Matlab phương pháp điều khiển hệ thống Pendubot toàn phương tuyến tính LQR và PID để mô phỏng và điều khiển hệ thống Pendubot. hay cụ thể hơn là điều khiển hai link của Pendubot cân bằng ở vị trí MID và TOP. Để thực hiện nhiệm vụ này là rất khó khăn vì đây là hệ phi tuyến hai bậc tự do và không ổn định, vì vậy công việc phải được sắp xếp hợp lý và cụ thể như sau:

- Nghiên cứu xây dựng mô hình toán hệ thống Pendubot.
- Nghiên cứu về nguyên lý cân bằng của hệ thống
- Thi công và tìm hiểu cách sử dụng module mạch giao tiếp 9090 và thu thập dữ liệu với máy tính PC..
- Nghiên cứu phương pháp điều khiển toàn phương tuyến tính LQR (Linear Quadratic Regulation), lập trình bằng ngôn ngữ Matlab để mô phỏng và điều khiển hệ thống Pendubot.
- Nghiên cứu phương pháp điều khiển PID, lập trình bằng ngôn ngữ Matlab để mô phỏng và điều khiển hệ thống Pendubot.
- Lập trình ngôn ngữ LabVIEW để thử tín hiệu Output, Input, PWM, Encoder.

- Điều khiển hệ thống từ vị trí cân bằng thẳng đứng sang vị trí cân bằng mới (link 1 ở vị trí cân bằng các góc bất kỳ và link 2 ở vị trí cân bằng thẳng lên) (như Hình 1.3)

#### **1.4 Phương pháp nghiên cứu:**

Dựa vào đặc tính động lực học của hệ thống dựa trên những cấu tạo về phần cứng và những tính chất vật lý, từ đây tác giả xây dựng phương pháp điều khiển hệ thống, vì vậy trong phần này sẽ giới thiệu tổng quan về phương pháp thực hiện như thế nào để hoàn thành và điều khiển được hệ thống Pendubot.

Do đây là hệ thống có độ phi tuyến cao để điều khiển được hệ thống này chúng ta phải biết được mô hình toán của hệ thống. Từ mô hình toán ta tuyến tính hóa hệ thống quanh điểm cân bằng của nó, những điểm cân bằng sẽ được giới thiệu phần sau và mô hình tuyến tính hóa ở những điểm cân bằng sẽ được giới thiệu. Sau đó xây dựng bộ điều khiển LQR và xây dựng bộ điều khiển PID để cân bằng cho hệ thống ở vị trí MID là nhiệm vụ trong luận văn này. Bộ điều khiển LQR, PID được viết trên phần mềm matlab 2010

Trong luận văn này tôi chọn hướng lập trình trên ngôn ngữ Matlab để mô phỏng hệ thống và lập trình thực nghiệm điều khiển hệ Pendubot trên ngôn ngữ LabVIEW, đây là một ngôn ngữ rất mạnh dùng cho lập trình có thể điều khiển trực tiếp mô hình thực.

Do thời gian có hạn và thiết bị cấu hình lên bộ Pendubot thấp, do đó tác giả thực nghiệm điều khiển hệ Pendubot trên bằng ngôn ngữ LabVIEW trên thuật toán PID đây là lợi thế rất lớn về thời gian và kinh tế.

Các phương pháp nghiên cứu được sử dụng trong luận văn bao gồm:

- Khảo sát, phân tích tổng hợp: Khảo sát một số mô hình, mô phỏng thực tế đang có từ các luận văn trước và các bài báo trên mạng internet, từ đó mô phỏng lại bằng phần mềm Matlab/simulink để so sánh với kết quả đã có nhằm rút ra những kinh nghiệm trong việc mô phỏng

- Mô phỏng trên máy tính: Phần nghiên cứu được kiểm chứng bằng việc mô phỏng và đánh giá trên phần mềm Matlab/simulink.

- Thiết kế mô hình thực nghiệm: Phần thiết kế mô hình thực nghiệm để kiểm chứng lại và đánh giá lại các kết quả mô phỏng dựa trên phần mềm LabVIEW.

### 1.5 Tổng quan chung về lĩnh vực nghiên cứu

- Tình hình nghiên cứu trên thế giới.

Hệ thống Pendubot là hệ thống phức tạp, không ổn định có tính phi tuyến cao việc điều khiển giữ cho con lắc cân bằng hay thiết kế bộ điều khiển Swing-up là những vấn đề rất được quan tâm trong lĩnh vực điều khiển tự động. Bên cạnh đó, nếu hệ thống được chế tạo với độ chính xác và tin cậy cao thì đây là mô hình lý tưởng cho các nhà nghiên cứu thực hiện các thí nghiệm về thu thập dữ liệu từ hệ thống để có thể sử dụng các thuật toán nhận dạng mô hình hệ thống mà không cần biết mô hình toán học của nó. Bên cạnh đó, việc sử dụng bộ điều khiển như thế nào để có thể giữ cho Pendubot đứng yên ở vị trí cân bằng là vấn đề rất quan trọng.

Hiện nay có rất nhiều nhà nghiên cứu đã sử dụng các thuật toán điều khiển khác nhau để điều khiển hệ thống Pendubot như thuật toán điều khiển PID, điều khiển trượt, thuật toán điều khiển tối ưu LQR và điều khiển logic mờ Fuzzy đã thu được một số thành công đáng kể được thực hiện bởi

K. J. Åström and K. Furuta, “Swing up a pendulum by energy control,” *Automatica*, vol. 36, pp. 287–295, 2000.

I. Fantoni, R. Lozano, and M. W. Spong, “Energy based control of the Pendubot,” *IEEE Trans. Automat. Contr.*, vol. 45, pp. 725–729, Apr. 2000.

Edgar Sanchez Luis A. Nuno, Ya-Chan Hsu, and Guanrong Chen, “Fuzzy PD Scheme for Underactuated Robot Swing-up Control”, *IEEE International Fuzzy Systems Conference Proceedings*, 1998.

Một số nghiên cứu đã được thực hiện như nghiên cứu của Mohan Akole và Barjeev Tyagi thiết kế bộ điều khiển Logic mờ (Fuzzy) để điều khiển ổn định hệ thống con lắc ngược. Kết quả đáp ứng của hệ thống khi sử dụng bộ điều khiển

Fuzzy hiệu quả hơn bộ điều khiển PID [4]. Tuy nhiên bộ điều khiển mờ hoạt động chủ yếu phụ thuộc vào kinh nghiệm và phương pháp rút ra kết luận theo tư duy con người, sau đó được cài đặt vào máy tính dựa trên cơ sở logic mờ. Nghiên cứu thứ hai là của Mojtaba Ahmadih Khanesar sử dụng bộ điều khiển trượt để điều khiển con lắc ngược quay. Kết quả của nghiên cứu này là tác giả đã xác định được hai mặt trượt theo định nghĩa đúng đắn của hàm Lyapunov, kết hợp hai mặt trượt này để điều khiển con lắc ổn định. Tuy nhiên trong phương pháp điều khiển trượt việc tìm kiếm bề mặt trượt của hệ thống là không dễ dàng [5] đã được chứng minh.

Tyaki, M.A.a.B., *Design of Fuzzy logic controller for nonlinear model of inverted pendulum - cart system*. December 17-19,2008.

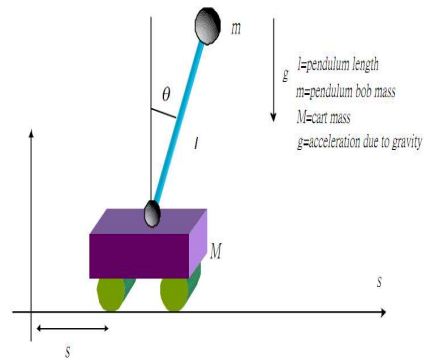
Khanesar, M.A., *Sliding Mode Control of Rotary Inverted Pendulum*. Tehran, Iran, July 27-29,2007.

Hầu hết các bộ điều khiển ở trên đều dựa trên hoạt động xung quanh một điểm. Các bộ điều khiển này sẽ thất bại nếu có bất kỳ hoạt động không chắc chắn hoặc thay đổi mà mô hình không biết. Những lợi thế của bộ điều khiển LQR và PID là nếu những thay đổi không chắc chắn trong quá trình hoạt động xảy ra LQR sẽ có thể thích ứng với các thông số của nó và duy trì kiểm soát các quá trình đó mà các bộ điều khiển khác không làm được.

Nhận dạng hệ thống là thủ tục để trình bày mô hình của hệ thống động lực căn cứ trên tín hiệu ngõ vào và ngõ ra từ hệ thống. Dữ liệu ngõ vào và ra trình bày một vài hoạt động động lực học. Các thông số của mô hình được điều chỉnh khi ngõ ra từ mô hình là giống với ngõ ra của hệ thống thực. Để trình bày một mô hình chính xác của hệ thống Pendubot, phương pháp khác (tuyến tính hay phi tuyến) của sự nhận dạng sẽ được kiểm tra. Một trong những vấn đề quan trọng là thu thập dữ liệu từ hệ thống. Dữ liệu ngõ ra từ hệ thống không ổn định không trình bày đủ thông tin và hoạt động của hệ thống. Việc điều chỉnh các thông tin phản hồi được trình bày một hệ thống ổn định trước sự nhận dạng có thể lấy ở các vị trí [6].

Johansson, R., *System modelling and identification*. Prentice Hall

*Li, J. H., Wu, M. F., and Lin, W. C., “Linear Quadratic Regulation of the Dsp-Based Pendubot,” Proceedings of the 2007 Intelligent Systems Conference on Engi-neering Applications, Tainan, Taiwan (2007).*



Hình 1.2a Con lắc ngược trên xe



Hình 1.2.b Con lắc ngược quay 1 bậc



Hình 1.2.c Các mô hình Pendubot

- Tình hình nghiên cứu trong nước

Trong những năm gần đây, hướng nghiên cứu điều khiển hệ thống Pendubot như thuật toán điều khiển PID, điều khiển trượt, thuật toán điều khiển tối ưu LQR và điều khiển logic mờ Fuzzy đã có nhiều công trình công bố những kết quả nghiên cứu về lý thuyết cũng như thực nghiệm, một số công trình đáng chú ý do nhóm tác giả trong nước Tuy nhiên các nghiên cứu trên chỉ dừng hầu hết đưa ra các giải thuật hoặc mô phỏng trên Matlab/Simulink, rất ít nghiên cứu đề cập đến kết quả thực nghiệm

Các bộ điều khiển PID hay LQR có trên thị trường của các hãng lớn thường được đóng gói thành sản phẩm hoàn chỉnh, khi lắp đặt vào hệ thống không cần phải làm gì thêm ngoài việc thiết lập các thông số hoạt động.



## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Lý thuyết điều khiển tự động

#### 2.1.1 Khái niệm

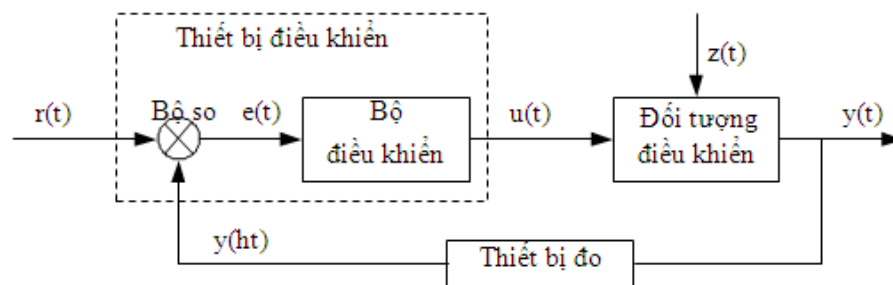
##### 2.1.1.1 Điều khiển

Là quá trình thu tập thông tin, xử lý thông tin và tác động lên hệ thống để đáp ứng hệ thống thỏa yêu cầu đặt ra.

Điều khiển tự động là quá trình điều khiển không có sự can thiệp của con người.

##### 2.1.1.2. Hệ thống điều khiển

Gồm ba thành phần cơ bản: Bộ điều khiển, Đối tượng điều khiển, Thiết bị đo. Như



Hình 2. 1: Mô tả hệ thống điều khiển

**Trong đó :**

$r(t)$ : tín hiệu vào.

$e(t)$  : tín hiệu sai lệch.

$u(t)$  : tín hiệu điều khiển.

$z(t)$ : tín hiệu nhiễu.

$y(t)$  : tín hiệu ra.

$y_{ht}(t)$  : tín hiệu hồi tiếp

### 2.1.1.3. Bộ điều khiển

Dùng tín hiệu sai lệch  $e$  để tạo tín hiệu điều khiển  $u$ . Thuật toán để xác định  $u(t)$  gọi là thuật toán điều khiển

Bộ điều khiển liên tục có thể thực hiện bằng cơ cấu cơ khí, thiết bị khí nén, mạch điện RLC, mạch khuếch đại thuật toán.

Bộ điều khiển số thực chất là các chương trình phần mềm chạy trên vi xử lý hay máy tính.

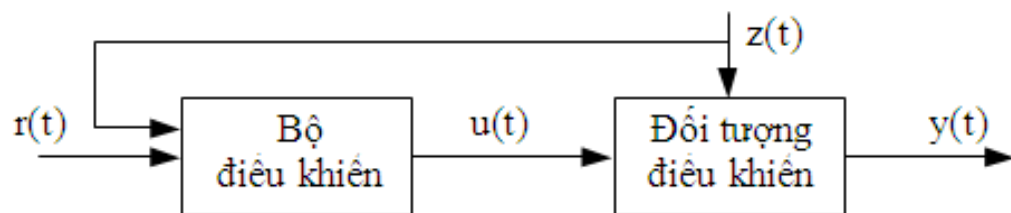
### 2.1.1.4. Các nguyên tắc điều khiển

Có ba nguyên tắc điều khiển giữ ổn định

#### - Nguyên tắc giữ ổn định

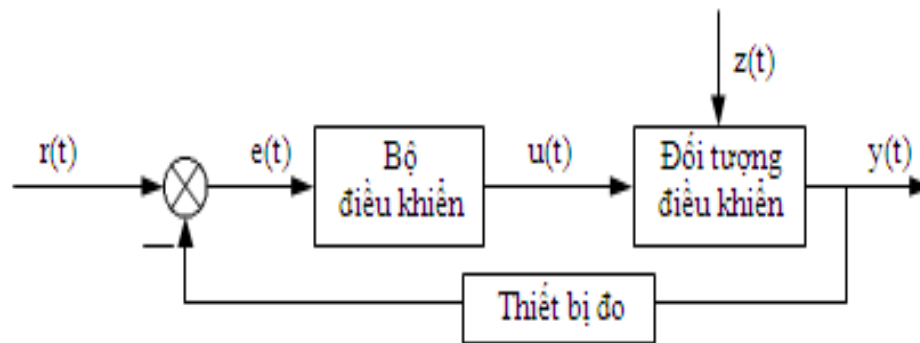
Là Giữ tín hiệu ra bằng tín hiệu vào:

+ **Điều khiển bù nhiễu** : Hệ thống lường trước nhiễu để tính toán, bù trừ sai số trước khi nhiễu ảnh hưởng đến đối tượng. Vì không thể lường hết các nhiễu nên điều khiển bù nhiễu chất lượng không cao.



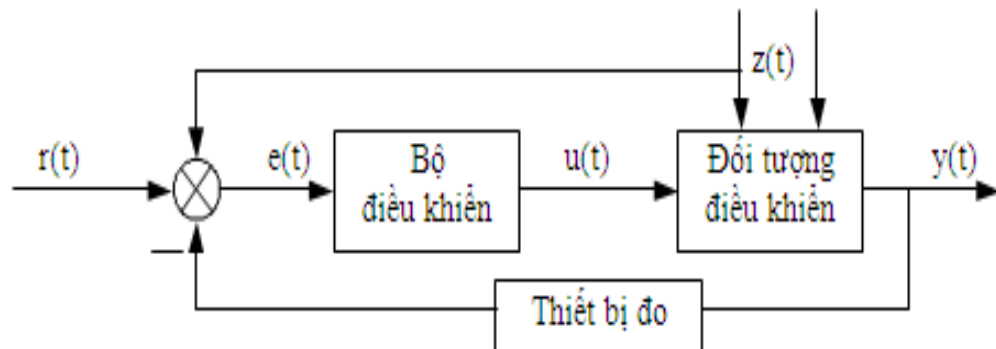
Hình 2. 2: Sơ đồ điều khiển bù nhiễu.

+ **Điều khiển san bằng sai lệch**: Tín hiệu ra  $y(t)$  được đo và phản hồi về so sánh với tín hiệu vào  $r(t)$ . Bộ điều khiển sử dụng độ sai lệch vào - ra để tính toán tín hiệu điều khiển  $u(t)$ . Nguyên tắc này có thể triệt tiêu các nhiễu không biết trước và không đo được. Tuy nhiên, nó điều khiển sau khi nhiễu đã thực sự ảnh hưởng đến đối tượng.



Hình 2. 3: Sơ đồ điều khiển cân bằng sai lệch.

**+ Điều khiển phối hợp:**



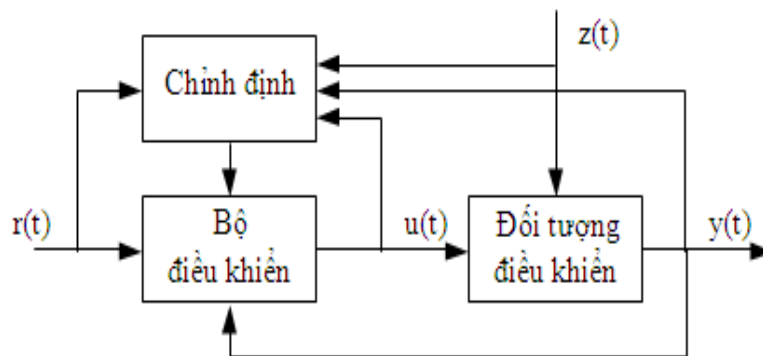
Hình 2. 4: Sơ đồ điều khiển phối hợp.

**- Nguyên tắc điều khiển theo chương trình**

Tín hiệu ra thay đổi theo một hàm thời gian định trước.

**- Nguyên tắc điều khiển thích nghi**

Khi cần điều khiển các đối tượng phức tạp, có thông số dễ bị thay đổi, hoặc nhiều đối tượng đồng thời mà phải đảm bảo cho một tín hiệu có giá trị cực trị, hay một chỉ tiêu tối ưu nào đó.



Hình 2. 5: Sơ đồ điều khiển thích nghi

## 2.2 Lý thuyết thuật toán điều khiển PID (Proportional Integral Derivative)

### 2.2.1. Cơ bản về vòng điều khiển

Một ví dụ quen thuộc của vòng điều khiển là hành động điều chỉnh vòi nước nóng và lạnh để duy trì nhiệt độ nước mong muốn ở đầu vòi nước. Thường ta phải trộn hai dòng nước, nóng và lạnh lại với nhau. Và chạm vào nước để cảm nhận hoặc ước lượng nhiệt độ của nó. Dựa trên phản hồi này, ta đi điều chỉnh van nóng và van lạnh cho đến khi nhiệt độ ổn định ở giá trị mong muốn.

Giá trị cảm biến nhiệt độ nước là giá trị tương tự(analog), dùng để đo lường giá trị xử lý hoặc biến quá trình (PV). Nhiệt độ mong muốn được gọi là điểm đặt (SP). Đầu vào chu trình (vị trí van nước) được gọi là biến điều khiển (MV). Hiệu số giữa nhiệt độ đo và điểm đặt được gọi là sai số (e), dùng để lượng hóa được khi nào thì nước quá nóng hay khi nào thì nước quá lạnh bằng giá trị.

Sau khi đo lường nhiệt độ (PV), và sau đó tính toán sai số, bộ điều khiển sẽ quyết định thời điểm thay đổi vị trí van (MV) và thay đổi bao nhiêu. Khi bộ điều khiển mở van lần đầu, nó sẽ mở van nóng tí xíu nếu cần nước ấm, hoặc sẽ mở hết cỡ nếu cần nước rất nóng. Đây là một ví dụ của điều khiển tỉ lệ đơn giản. Trong trường hợp nước nóng không được cung cấp nhanh chóng, bộ điều khiển có thể tìm cách tăng tốc độ của chu trình lên bằng cách tăng độ mở của van nóng theo thời gian. Đây là một ví dụ của điều khiển tích phân. Nếu chỉ sử dụng hai phương pháp

điều khiển tỉ lệ và tích phân, trong vài hệ thống, nhiệt độ nước có thể dao động giữa nóng và lạnh, bởi vì bộ điều khiển điều chỉnh van quá nhanh và vọt lố hoặc bù lố so với điểm đặt.

Để đạt được sự hội tụ tăng dần đến nhiệt độ mong muốn (SP), bộ điều khiển cần phải yêu cầu làm tắt dần dao động dự đoán trong tương lai. Điều này có thể thực hiện bởi phương pháp điều khiển vi phân.

Giá trị thay đổi có thể quá lớn khi sai số tương ứng là nhỏ đối với bộ điều khiển có độ lợi lớn và sẽ dẫn đến vọt lố. Nếu bộ điều khiển lặp lại nhiều lần việc thay đổi này sẽ dẫn đến thường xuyên xảy ra vọt lố, đầu ra sẽ dao động xung quanh điểm đặt, tăng hoặc giảm theo hình sin cố định. Nếu dao động tăng theo thời gian thì hệ thống sẽ không ổn định, còn nếu dao động giảm theo thời gian thì hệ thống đó ổn định. Nếu dao động duy trì tại một biên độ cố định thì hệ thống là ổn định biên độ. Con người không dễ xảy ra dao động như vậy bởi vì chúng ta là những "bộ" điều khiển thích nghi, biết rút kinh nghiệm; tuy nhiên, bộ điều khiển PID đơn giản không có khả năng học tập và phải được thiết đặt phù hợp. Việc chọn độ lợi hợp lý để điều khiển hiệu quả được gọi là điều chỉnh bộ điều khiển.

Nếu một bộ điều khiển bắt đầu từ một trạng thái ổn định tại điểm sai số bằng 0 ( $PV=SP$ ), thì những thay đổi sau đó bởi bộ điều khiển sẽ phụ thuộc vào những thay đổi trong tín hiệu đầu vào đo được hoặc không đo được khác tác động vào quá trình điều khiển, và ảnh hưởng tới đầu ra PV. Các biến tác động vào quá trình khác với MV được gọi là nhiễu. Các bộ điều khiển thông thường được sử dụng để loại trừ nhiễu và/hoặc bổ sung những thay đổi điểm đặt. Những thay đổi trong nhiệt độ nước cung cấp là do nhiễu trong quá trình điều khiển nhiệt độ ở vòi nước.

Về lý thuyết, một bộ điều khiển có thể được sử dụng để điều khiển bất kỳ một quá trình nào mà có một đầu ra đo được (PV), một giá trị lý tưởng biết trước cho đầu ra (SP) và một đầu vào chu trình (MV) sẽ tác động vào PV thích hợp. Các bộ điều khiển được sử dụng trong công nghiệp để điều chỉnh nhiệt độ, áp suất, tốc độ dòng chảy, tổng hợp hóa chất, tốc độ và các đại lượng khác có thể đo lường

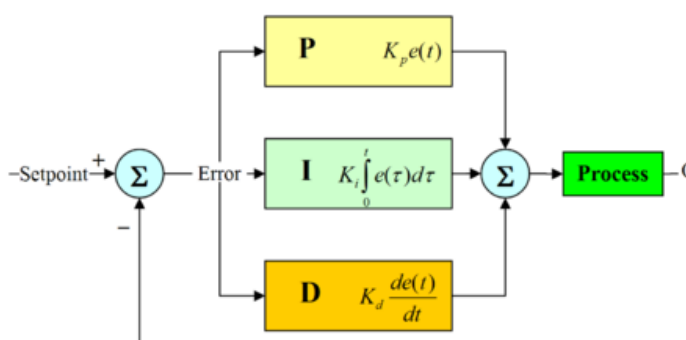
được. Xe hơi điều khiển hành trình là một ví dụ cho việc áp dụng điều khiển tự động trong thực tế.

Các bộ điều khiển PID thường được lựa chọn cho nhiều ứng dụng khác nhau, vì lý thuyết tin cậy, được kiểm chứng qua thời gian, đơn giản và dễ cài đặt cũng như bảo trì của chúng.

### 2.2.2. Giới thiệu bộ điều khiển PID

Bộ điều khiển PID là một bộ điều khiển vòng kín được sử dụng rộng rãi trong công nghiệp. Sử dụng bộ điều khiển PID để điều chỉnh sai lệch giữa giá trị đo được của hệ thống (process variable) với giá trị đặt (setpoint) bằng cách tính toán và điều chỉnh giá trị điều khiển ở ngõ ra.

Sơ đồ một hệ thống điều khiển dùng PID:



Hình 2. 6: Sơ đồ hệ thống điều khiển dùng PID

Một bộ điều khiển PID gồm 3 thành phần: P (proportional) – tạo tín hiệu điều khiển tỉ lệ với sai lệch (error – e), I (integral) – tạo tín hiệu điều khiển tỉ lệ với tích phân theo thời gian của sai lệch, và D (derivative) – tạo tín hiệu điều khiển tỉ lệ với vi phân theo thời gian của sai lệch.

### 2.2.2.1. Khâu P

Khâu P tạo ra tín hiệu điều khiển tỉ lệ với giá trị của sai lệch. Việc này được thực hiện bằng cách nhân sai lệch  $e$  với hằng số  $K_P$  – gọi là hằng số tỉ lệ.

Khâu P được tính dựa trên công thức:

$$P_{\text{out}} = K_p e(t)$$

Với:  $P_{\text{out}}$ : giá trị ngõ ra

$K_P$ : hằng số tỉ lệ

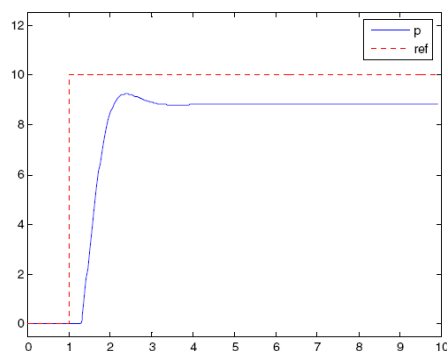
$e$ : sai lệch:  $e = SP - PV$

Sơ đồ khối của khâu P:



Hàm truyền:  $G_p(s) = K_p$

Nếu chỉ có khâu P thì trong mọi trường hợp sai số tĩnh luôn xuất hiện, trừ khi giá trị đầu vào của hệ thống bằng 0 hoặc đã bằng với giá trị mong muốn. Trong hình sau thể hiện sai số tĩnh xuất hiện khi thay đổi giá trị đặt.



Hình 2. 7: Đáp ứng của khâu P

Nếu giá trị khâu P quá lớn sẽ làm cho hệ thống mất ổn định.

### 2.2.2.2. Khâu I

Khâu I cộng thêm tổng các sai số trước đó vào giá trị điều khiển. Việc tính tổng các sai số được thực hiện liên tục cho đến khi giá trị đạt được bằng với giá trị đặt, và kết quả là khi hệ cân bằng thì sai số bằng 0.

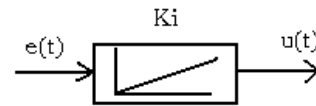
Khâu I được tính theo công thức:  $I_{out} = K_i \int_0^t e(\tau) d\tau$

Với:  $I_{OUT}$ : giá trị ngõ ra khâu I

$K_i$ : hệ số tích phân

$e$ : sai số:  $e = SP - PV$

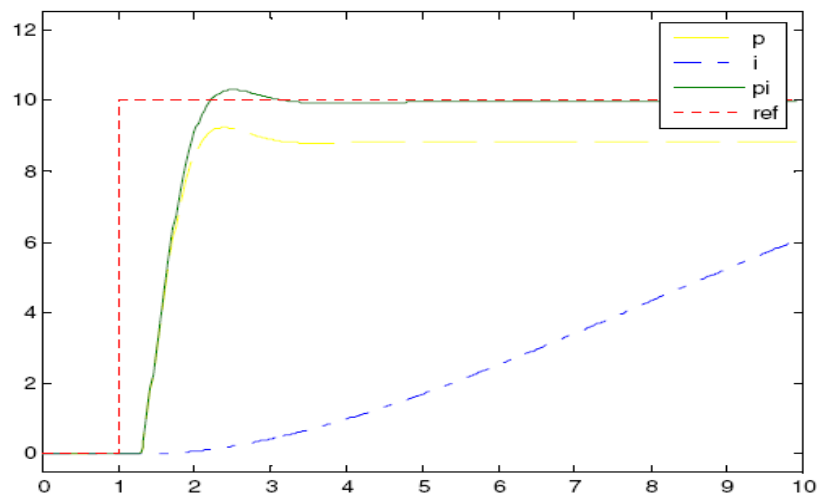
Sơ đồ khối khâu I:



$$\text{Hàm truyền: } G(s) = \frac{U(s)}{E(s)} = \frac{K_I}{s} = \frac{I}{T_I s}$$

Khâu I thường đi kèm với khâu P, hợp thành bộ điều khiển PI. Nếu chỉ sử dụng khâu I thì đáp ứng của hệ thống sẽ chậm và thường bị dao động.

Hình sau chỉ ra sự khác biệt giữa khâu I và PI:



Hình 2. 8: Đáp ứng của khâu I và PI



Ta có thể nhận thấy là khâu I làm cho đáp ứng của hệ thống bị chậm đi rất nhiều, còn khâu PI giúp triệt tiêu sai số xác lập.

### 2.2.2.3. Khâu D

Khâu D cộng thêm tốc độ thay đổi sai số vào giá trị điều khiển ở ngõ ra. Nếu sai số thay đổi nhanh thì sẽ tạo ra thành phần cộng thêm vào giá trị điều khiển. Điều này cải thiện đáp ứng của hệ thống, giúp trạng thái của hệ thống thay đổi nhanh chóng và mau chóng đạt được giá trị mong muốn.

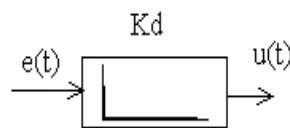
Khâu D được tính theo công thức:  $D_{out} = K_d \frac{de}{dt}$

Với:  $D_{OUT}$ : ngõ ra khâu D

$K_D$ : hệ số vi phân

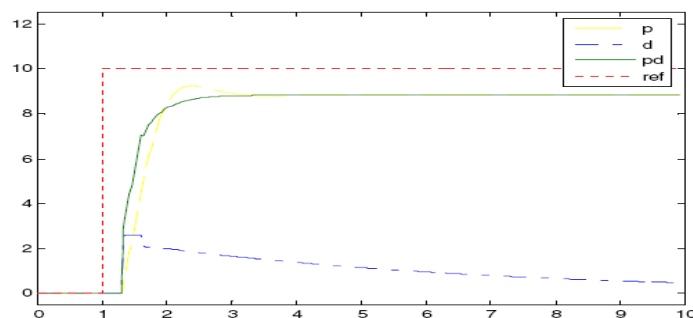
e: sai số:  $e = SP - PV$

Sơ đồ khối khâu D:



Hàm truyền:  $G(s) = \frac{U(s)}{E(s)} = K_d s$

Khâu D thường đi kèm với khâu P thành bộ PD, hoặc với PI để thành bộ PID.



Hình 2. 9 : Đáp ứng của khâu D và PD

Theo hình trên, bộ PD tạo đáp ứng có thời gian tăng trưởng nhỏ hơn so với bộ P. Nếu giá trị D quá lớn sẽ làm cho hệ thống không ổn định.

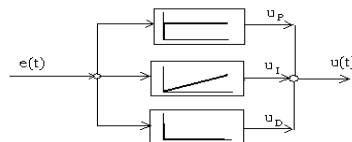
#### 2.2.2.4. Tổng hợp ba khâu – Bộ điều khiển PID

Bộ điều khiển PID là cấu trúc ghép song song giữa 3 khâu P, I và D.

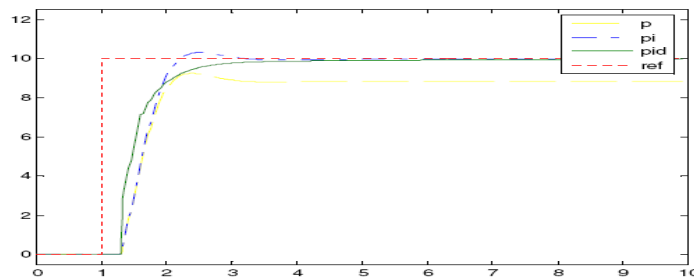
Phương trình vi phân của bộ PID lý tưởng:

$$u(t) = K_p e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt}$$

Sơ đồ khối:



Đáp ứng của bộ PID:



Hình 2. 10: Đáp ứng của khâu P, PI và PID

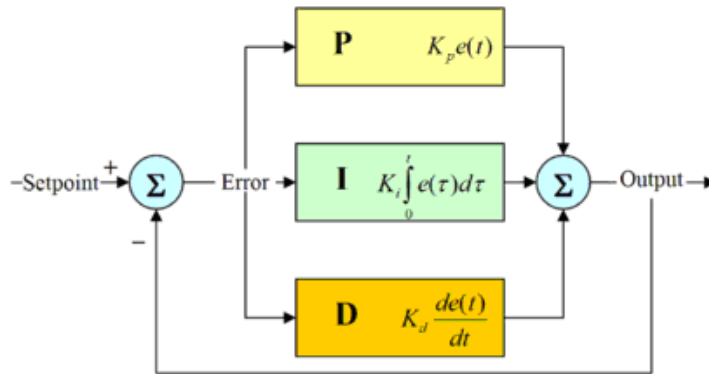
#### 2.2.2.5. Rời rạc hóa bộ điều khiển PID:

Bộ điều khiển số không thể lấy mẫu liên tục theo thời gian, nó cần được rời rạc ở một vài mức. Khi cho hệ số lấy mẫu ngắn bên trong thời gian vi phân có thể đạt được xấp xỉ một sai phân có giới hạn và tích phân qua việc lấy tổng. Chúng ta sẽ quan tâm mỗi dạng ở một thời điểm, và sai số được tính ở mỗi khoảng lấy mẫu:

$$e(n) = X(n) - Y(n)$$

Bộ PID rời rạc đọc sai số, tính toán và xuất ngõ ra điều khiển theo một khoảng thời gian xác định (không liên tục) – thời gian lấy mẫu T. Thời gian lấy mẫu cần nhỏ hơn đơn vị thời gian của hệ thống.

Không giống các thuật toán điều khiển đơn giản khác, bộ điều khiển PID có khả năng xuất tín hiệu ngõ ra dựa trên giá trị trước đó của sai số cũng như tốc độ thay đổi sai số. Điều này giúp cho quá trình điều khiển chính xác và ổn định hơn.



Hình 2. 11: Sơ đồ khối PID

Hàm truyền của hệ thống:

$$\frac{u}{e}(s) = H(s) = K_p \left( 1 + \frac{1}{T_i s} + T_d s \right)$$

Hàm chuyển đổi:

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\sigma) d\sigma + T_d \frac{de(t)}{dt} \right)$$

Tính gần đúng theo công thức:

$$\int_0^t e(\sigma) d\sigma \approx T \sum_{k=0}^n e(k) \quad \frac{de(t)}{dt} \approx \frac{e(n) - e(n-1)}{T} \quad t = nT$$

Với n là bước rời rạc tại t.

Kết quả thu được:  $u(n) = K_p e(n) + K_i \sum_{k=0}^n e(k) + K_d (e(n) - e(n-1))$

Với:  $K_i = \frac{K_p T}{T_i} \quad K_d = \frac{K_p T_d}{T}$

### 2.2.3. Điều chỉnh vòng lặp

Điều chỉnh một vòng điều khiển là điều chỉnh các thông số điều khiển của nó (độ lợi/dải tỉ lệ, độ lợi tích phân/reset, độ lợi vi phân/tốc độ) tới giá trị đáp ứng điều khiển tối ưu. Độ ổn định (dao động biên) là một yêu cầu căn bản, nhưng ngoài ra, các hệ thống khác nhau, có những hành vi khác nhau, những ứng dụng khác nhau có những yêu cầu khác nhau, và vài yêu cầu lại mâu thuẫn với nhau. Hơn nữa, vài quá trình có một mức độ phi tuyến nào đấy khiến các thông số làm việc tốt ở điều kiện đầy tải sẽ không làm việc khi quá trình khởi động từ không tải; điều này có thể khắc phục bằng chương trình độ lợi (sử dụng các thông số khác nhau cho những khu vực hoạt động khác nhau). Các bộ điều khiển PID thường cung cấp các điều khiển có thể chấp nhận được thậm chí không cần điều chỉnh, nhưng kết quả nói chung có thể được cải thiện bằng cách điều chỉnh kỹ lưỡng, và kết quả có thể không chấp nhận được nếu điều chỉnh kém.

Điều chỉnh PID là một bài toán khó, ngay cả khi chỉ có 3 thông số và về nguyên tắc là dễ miêu tả, bởi vì nó phải thỏa mãn các tiêu chuẩn phức tạp nằm trong Những hạn chế của điều khiển PID. Vì vậy có nhiều phương pháp khác nhau để điều chỉnh vòng lặp, và các kỹ thuật phức tạp hơn là đề tài cho nhiều phát minh sáng chói; phần này miêu tả vài phương pháp thủ công truyền thống để điều chỉnh vòng lặp.

#### 2.2.3.1 Độ ổn định

Nếu các thông số của bộ điều khiển PID (độ lợi của khâu tỉ lệ, tích phân và vi phân) được chọn sai, đầu vào quá trình điều khiển có thể mất ổn định, vì các khác biệt đầu ra của nó, có hoặc không có dao động, và được giới hạn chỉ bởi sự bão hòa hoặc đứt gãy cơ khí. Sự không ổn định được gây ra bởi sự dư thừa độ lợi, nhất là khi xuất hiện độ trễ lớn.

Nói chung, độ ổn định của đáp ứng (ngược với độ bất định) phải thỏa mãn và quá trình phải không được dao động vì bất kỳ sự kết hợp nào giữa các điều kiện

quá trình và điểm đặt, mặc dù đôi khi ổn định biên có thể được chấp nhận hoặc yêu cầu.

### **2.2.3.2 Tối ưu hóa hành vi**

Tối ưu hóa hành vi trong thay đổi quá trình hoặc thay đổi điểm đặt khác nhau tùy thuộc vào ứng dụng.

Hai yêu cầu cơ bản là ổn định (trệt tiêu nhiễu-ổn định tại một điểm đặt cho trước) và tự hiệu chỉnh lệnh (thực hiện các thay đổi điểm đặt)-hai yêu cầu đó tùy thuộc vào việc các biến điều khiển theo dõi giá trị mong muốn có tốt hay không. Các tiêu chuẩn đặc biệt về tự hiệu chỉnh lệnh bao gồm thời gian khởi động và thời gian xác lập. Một vài quá trình phải ngăn không cho phép các biến quá trình vọt lố quá điểm đặt nếu, thí dụ, điều này có thể mất an toàn. Các quá trình khác phải tối thiểu hóa năng lượng tiêu hao khi tiến tới một điểm đặt mới.

### **2.2.3.3 Tổng quan các phương pháp**

Có nhiều phương pháp khác nhau để điều chỉnh vòng lặp PID. Những phương pháp hữu hiệu nhất thường bao gồm những triển khai của vài dạng mô hình xử lý, sau đó chọn P, I, và D dựa trên các thông số của mô hình động học. Các phương pháp điều chỉnh thủ công tương đối không hiệu quả lắm, đặc biệt nếu vòng lặp có thời gian đáp ứng được tính bằng phút hoặc lâu hơn.

Lựa chọn phương pháp thích hợp sẽ phụ thuộc phần lớn vào việc có hay không vòng lặp có thể điều chỉnh "offline", và đáp ứng thời gian của hệ thống. Nếu hệ thống có thể thực hiện offline, phương pháp điều chỉnh tốt nhất thường bao gồm bắt hệ thống thay đổi đầu vào từng bước, tín hiệu đo lường đầu ra là một hàm thời gian, sử dụng đáp ứng này để xác định các thông số điều khiển.

Lựa chọn phương pháp điều chỉnh		
Phương pháp	Ưu điểm	Khuyết điểm
Điều chỉnh thủ công	Không cần hiểu biết về toán. Phương pháp online.	Yêu cầu nhân viên có kinh nghiệm.
Ziegler–Nichols	<i>Phương pháp chứng minh. Phương pháp online.</i>	<i>Làm rối loạn quá trình, một số thử nghiệm và lỗi, phải điều chỉnh nhiều lần</i>
Các công cụ phần mềm	Điều chỉnh chắc chắn. Phương pháp online hoặc offline. Có thể bao gồm phân tích các van và cảm biến. Cho phép mô phỏng trước khi tải xuống để thực thi.	Giá cả cao, và phải huấn luyện.
Cohen-Coon	xử lý các mô hình tốt.	Yêu cầu kiến thức toán học. Phương pháp offline. Chỉ tốt đối với các quá trình bậc một.

Bảng 2.1: Lựa chọn phương pháp điều chỉnh

#### 2.2.3.4 Điều chỉnh thủ công

Nếu hệ thống phải duy trì trạng thái online, một phương pháp điều chỉnh là thiết đặt giá trị đầu tiên của  $K_i$  và  $K_d$  bằng không. Tăng dần  $K_p$  cho đến khi đầu ra của vòng điều khiển dao động, sau đó  $K_p$  có thể được đặt tới xấp xỉ một nửa giá trị đó để đạt được đáp ứng "1/4 giá trị suy giảm biên độ". Sau đó tăng  $K_i$  đến giá trị phù hợp sao cho đủ thời gian xử lý. Tuy nhiên,  $K_i$  quá lớn sẽ gây mất ổn định. Cuối cùng, tăng  $K_d$ , nếu cần thiết, cho đến khi vòng điều khiển nhanh có thể chấp nhận được nhanh chóng lấy lại được giá trị đặt sau khi bị nhiễu. Tuy nhiên,  $K_d$  quá lớn sẽ gây đáp ứng dư và vọt lố. Một điều chỉnh cấp tốc của vòng điều khiển PID thường hơi quá lố một ít khi tiến tới điểm đặt nhanh chóng; tuy nhiên, vài hệ thống

không chấp nhận xảy ra vọt lố, trong trường hợp đó, ta cần một hệ thống vòng kín giảm lố, thiết đặt một giá trị  $K_p$  nhỏ hơn một nửa giá trị  $K_p$  gây ra dao động.

Tác động của việc tăng một thông số độc lập					
Thông số	Thời gian khởi động	Quá độ	Thời gian xác lập	Sai số ổn định	Độ ổn định <sup>[3]</sup>
$K_p$	Giảm	Tăng	Thay đổi nhỏ	Giảm	Giảm cấp
$K_i$	Giảm <sup>[4]</sup>	Tăng	Tăng	Giảm đáng kể	Giảm cấp
$K_d$	Giảm ít	Giảm ít	Giảm ít	Về lý thuyết không tác động	Cải thiện nếu $K_d$ nhỏ

Bảng 2.2: Tác động của việc tăng một thông số độc lập

### 2.2.3.5 Phương pháp Ziegler–Nichols

Một phương pháp điều chỉnh theo kinh nghiệm khác là phương pháp Ziegler–Nichols, được đưa ra bởi John G. Ziegler và Nathaniel B. Nichols. Giống phương pháp trên, độ lợi  $K_i$  và  $K_d$  lúc đầu được gán bằng không. Độ lợi P được tăng cho đến khi nó tiến tới độ lợi tới hạn,  $K_u$ , ở đầu ra của vòng điều khiển bắt đầu dao động.  $K_u$  và thời gian giao động  $P_u$  được dùng để gán độ lợi như sau:

Phương pháp Ziegler–Nichols			
Dạng điều khiển	$K_p$	$K_i$	$K_d$
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2K_u/P_u$	-
PID	$0.60K_u$	$2K_u/P_u$	$K_u P_u/8$

Bảng 2. 3: Thông số điều chỉnh bằng phương pháp Ziegler–Nichols

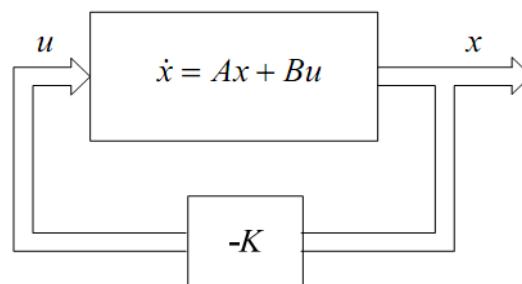
## 2.3 Cơ sở kiến thức chung điều khiển tối ưu

Một hệ điều khiển được thiết kế ở chế độ làm việc tốt nhất là hệ luôn ở trạng thái tối ưu theo một tiêu chuẩn chất lượng nào đó ( đạt được giá trị cực trị ) . Trạng thái tối ưu có đạt được hay không tùy thuộc vào yêu cầu chất lượng đặt ra, vào sự hiểu biết về đối tượng và các tác động lên đối tượng , vào điều kiện làm việc của hệ điều khiển

Khảo sát vấn đề duy trì trạng thái của hệ thống ở giá trị là 0, chống tác động nhiễu, đồng thời với cực tiểu tiêu hao năng lượng

$$(5.1) \quad \begin{aligned} \dot{x} &= Ax + Bu, x(0) = x^0 \\ y &= Cx \\ \min J &= \frac{1}{2} \int_0^{\infty} [x^T Q x + u^T R u] dt, \end{aligned}$$

### Sơ đồ thiết kế cần thiết



#### Trong đó:

Q là ma trận đối xứng xđd hay bán xđd, thường là ma trận chéo

R là ma trận đối xứng xđd, thường là ma trận chéo

Chọn luật điều khiển hồi tiếp trạng thái  $u = -Kx$ , K là hằng số, thay vào biểu thức của J

$$J = \frac{1}{2} \int_0^{\infty} x^T (Q + K^T R K) x dt$$

Tính K dùng phương trình Lyapunov, chọn hàm Lyapunov là J:

$$V(x(t)) = \frac{1}{2} \int_t^{\infty} x^T (Q + K^T R K) x dt = \frac{1}{2} x^T P x$$



$$V(x(0)) = J = x^T(0)Px(0)$$

Đạo hàm theo thời gian:

$$\dot{V}(x) = \frac{1}{2} x^T (Q + K^T RK) x \Big|_t^\infty = \frac{1}{2} x^T(\infty) [Q + K^T RK] x(\infty) - \frac{1}{2} x^T(t) [Q + K^T RK] x(t)$$

Giả sử chọn K để hệ ổn định,  $x(\infty) \rightarrow 0$

$$\dot{V}(x) = -\frac{1}{2} x^T(t) [Q + K^T RK] x(t)$$

Mặt khác

$$\dot{V}(x) = \frac{1}{2} (\dot{x}^T Px + x^T P \dot{x}) = \frac{1}{2} x^T [(A - BK)^T P + P(A - BK)] x$$

Suy ra 
$$\frac{1}{2} x^T [(A - BK)^T P + P(A - BK)] x = -\frac{1}{2} x^T (Q + K^T RK) x$$

Ma trận P thỏa phương trình Lyapunov

$$(A - BK)^T P + P(A - BK) = -(Q + K^T RK) \quad (2.1)$$

## 2.4 Các bước giải phương trình tối ưu

Giải phương trình Lyapunov ta được các phần tử của ma trận P theo các phần tử của ma trận K chưa biết. Sau đó ta tính  $J = V(x(0)) = \frac{1}{2} x^T(0)Px(0)$  là hàm theo các phần tử của ma trận K

Để J cực tiểu ta giải phương trình  $\frac{\partial J}{\partial k_{ij}} = 0$  hay  $\frac{\partial J}{\partial k_{ij}} = 0$

Suy ra ma trận K, luật điều khiển  $u = -Kx$

Xét ổn định của ma trận A-BK

Nếu muốn điều chỉnh ngõ ra  $y=cx$  ta chọn

$$J = \frac{1}{2} \int_0^\infty x^T (C^T QC + K^T RK) x dt$$

Đặt  $R = \Gamma^T \Gamma$ ,  $\Gamma$  là ma trận vuông không suy biến

Phương trình Lyapunov viết lại là:

$$(A^T - K^T B^T)P + P(A - BK) + Q + K^T \Gamma^T \Gamma K = 0$$

$$A^T P + PA + [\Gamma K - (\Gamma^T)^{-1} B^T P]^T [\Gamma K - (\Gamma^T)^{-1} B^T P] - P B R^{-1} B^T P + Q = 0$$

Lấy đạo hàm phương trình theo  $k_{ij}$  và dùng tính chất  $\frac{\partial P}{\partial k_{ij}} = 0$

Ta suy ra:  $\frac{\partial}{\partial k_{ij}}[(\Gamma K - (\Gamma^T)^{-1} B^T P)^T (\Gamma K - (\Gamma^T)^{-1} B^T P)] = 0$

Cực tiểu xảy ra khi số hạng trong ngoặc là 0

$$\begin{aligned}\Gamma K &= (\Gamma^T)^{-1} B^T P \\ K &= \Gamma^{-1} (\Gamma^T)^{-1} B^T P = R^{-1} B^T P\end{aligned}\tag{2.2}$$

Phương trình Lyapunov trở thành phương trình đại số Riccati

$$A^T P + PA - PBR^{-1}B^T P + Q = 0\tag{2.3}$$

Trong phương trình trên không chứa K

Đây là kết quả rất quan trọng trong lý thuyết điều khiển hiện đại. Phương trình (2.3) là phương trình đại số Riccati (algebraic Riccati equation) (ARE)

Thủ tục thiết kế để tìm hồi tiếp K của LQR như sau:

Lựa chọn thông số ma trận Q và R

Giải phương trình Riccati cho P

Tìm hồi tiếp biến trạng thái (SVFB) sử dụng  $K = R^{-1} B^T P$

Thực thi trong MATLAB bằng hàm `lqr(A,B,Q,R)`.

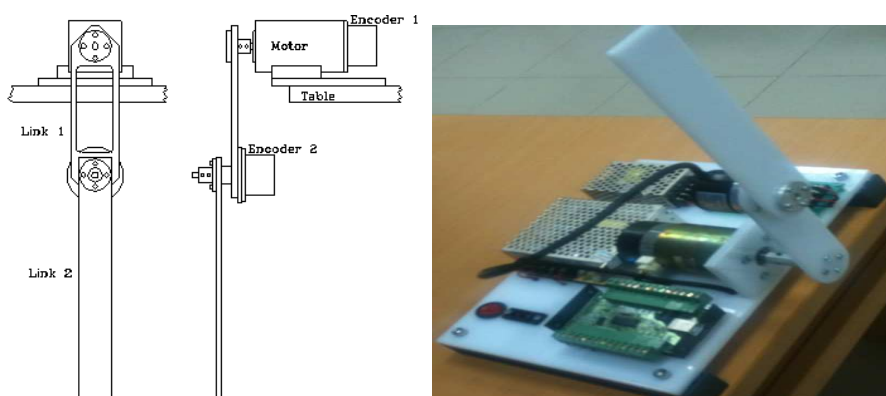
## CHƯƠNG 3: PHƯƠNG TRÌNH TOÁN HỌC HỆ PENDUBOT

Chương này giới thiệu sơ lược về hệ thống Pendubot, quá trình xây dựng mô hình toán học của hệ thống Pendubot tuyến tính và phi tuyến, khảo sát hệ Pendubot tuyến tính và phi tuyến khi có bộ điều khiển.

### 3.1. Hệ thống Pendubot

Hệ thống Pendubot là một vấn đề điều khiển cổ điển nó được sử dụng trong các trường đại học trên khắp thế giới, hệ thống Pendubot là mô hình phù hợp để kiểm tra các thuật toán điều khiển hệ phi tuyến, điều khiển tuyến tính, điều khiển logic mờ, điều khiển thông minh... để trở lại ổn định.

Mô hình Pendubot gồm nhiều phần: Link 1 gắn vào động cơ DC quay quanh trục nằm ngang, và Link 2 nối tiếp với Link 1 quay tự do, được thể hiện theo hình sau:



Hình 3.1: Hệ thống Pendubot

Hệ Pendubot không ổn định, nó luôn ngã xuống, trừ khi có lực tác động của động cơ DC một cách thích hợp cả hai Link để quay quay. Bài toán đặt ra là điều động cơ DC điều khiển cho Link 1 và Link 2 thẳng đứng và để giữ cho Link 2 luôn được giữ thẳng bằng khi có một ngoại lực tác động vào bất kỳ trên hai Link. Hệ Pendubot được giữ thẳng bằng nhờ có cân bằng phụ thuộc vào cả hai thông số động năng và thế năng, bộ điều khiển PID ổn định lâu dài khi có ngoại lực tác động vào hệ Pendubot.

Mô hình mô phỏng của hệ Pendubot dựa theo phương trình động lực của hệ thống Pendubot. Đồng thời bộ điều khiển cơ bản cũng sẽ được hình trình bày mô phỏng trong nghiên cứu này. Mục đích của việc trình bày hệ Pendubot trong mô phỏng là sự mô tả mô hình sẽ giống đặc điểm như mô hình thực. Nó sẽ kiểm tra bộ điều chỉnh mẫu trong môi trường mô phỏng

### 3.2 Cơ sở chung

Phương pháp Lagrange dựa trên mối quan hệ tổng năng lượng của hệ thống với lực tổng quát trong một hệ tọa độ tổng quát.

Giả sử đã xác định được một tập hợp các biến  $\lambda, i=1.....n$  với tư cách là tọa độ tổng quát để mô tả vị trí các khâu của một tay máy n bậc tự do. Lagrange của cơ hệ là hàm số của các tọa độ tổng quát.

$$L = K - V \quad (4.1)$$

Trong đó K và V tương ứng là động năng và thế năng của hệ.

Từ đó, ta có công thức Lagrange như sau:

$$\frac{d}{dt} \left[ \frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} = \tau \quad (4.2)$$

$$\tau = \begin{bmatrix} \tau_m \\ 0 \end{bmatrix} : \text{là lực tổng quát liên kết với hệ tọa độ tổng quát}$$

$q, \dot{q}$  : vector góc và vận tốc góc hệ thống

Động năng của hệ thống là hàm toàn phương của  $\dot{q}$  ( $\dot{q}$  là vector vận tốc góc)

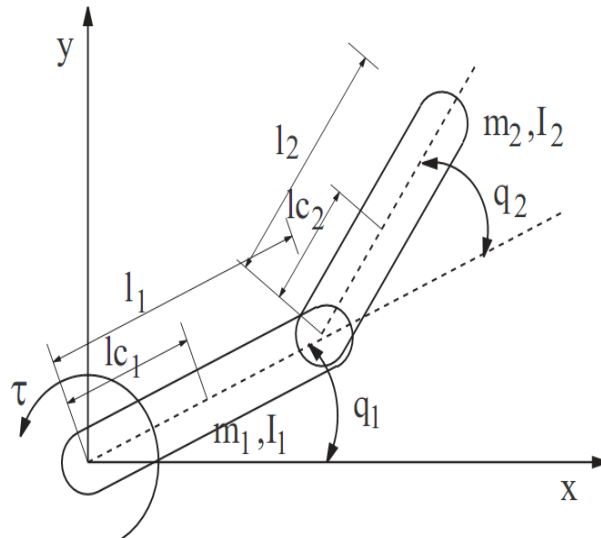
$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (4.3)$$

Ở đây D(q): là ma trận đối xứng định nghĩa dương với mỗi  $q \in \mathbb{R}^2$

Thế năng của Pendubot là hàm của vị trí góc  $V = V(q)$ , và độc lập với  $\dot{q}$ .

### 3.3 Phương trình toán hệ thống

#### 3.3.1 Giới thiệu các thông số của hệ thống.



Hình 3.2: Hệ trục tọa độ Pendubot

Những mô hình toán của hệ thống trong luận văn này dựa trên trục tọa độ như (Hình 3.1) phía trên, từ hệ trục này ta xây dựng mô hình tại vị trí Mid vị trí Top.

Những mô hình của hệ thống đều dựa trên hệ trục tọa độ như (Hình 3.1) với hai góc  $q_1$  và  $q_2$  tương ứng với link1 và link2. Góc  $q_1$  là góc hợp bởi trục x và link1, góc  $q_2$  là góc hợp bởi link1 và link2.

Những tham số của hệ thống Pendubot:

$q_1$  : Góc của link1 (rad)

$q_2$  : Góc của link2 (rad)

$\dot{q}_1$  : Vận tốc góc link1

$\dot{q}_2$  : Vận tốc góc link2

$\ddot{q}_1$  : Gia tốc góc của link1

$\dot{q}_2$  : Gia tốc gốc của link2

$D(q)$  : Ma trận quán tính

$C(q, \dot{q})$  : Coriolis/ vector hướng tâm

$G(q)$  : Vector trọng lực

$d_{11}$  : Tham số ma trận quán tính

$d_{12}$  : Tham số ma trận quán tính

$d_{21}$  : Tham số ma trận quán tính

$d_{22}$  : Tham số ma trận quán tính

$h$  : tham số của vector hướng tâm

$\theta_1$  : Tham số của vector trọng lực

$\theta_2$  : Tham số của vector trọng lực

$m_1$  : Tổng trọng lượng của link 1

$\tau$  : Moment quán tính của motor

$\beta_1$  : Tham số động lực học của Pendubot

$\beta_2$  : Tham số động lực học của Pendubot

$\beta_3$  : Tham số động lực học của Pendubot

$\beta_4$  : Tham số động lực học của Pendubot

$\beta_5$  : Tham số động lực học của Pendubot

$l_1$  : Chiều dài link 1

$l_{c1}$  : Khoảng từ trục của động cơ đến trọng tâm link 1

$I_1$  : Moment quán tính của link 1

- $m_2$  : Tổng trọng lượng của link 2
- $l_2$  : Chiều dài link 2
- $l_{c2}$  : Khoảng cách từ trục encoder trọng tâm link 2
- $I_2$  : Moment quán tính của link 2
- $g$  : Gia tốc trọng trường
- $C_n$  : Ma trận điều khiển được
- $A$  : Ma trận không gian trạng thái được tuyến tính
- $B$  : Ma trận moment đầu vào được tuyến tính
- $x_1$  : Biến không gian trạng thái cho góc của link1
- $x_2$  : Biến không gian trạng thái cho vận tốc góc của link1
- $x_3$  : Biến không gian trạng thái cho góc của link2
- $x_4$  : BIẾN không gia trạng thái cho vận tốc góc link2
- $K_1$  : Động năng link1
- $K_2$  : Động năng link2
- $K_{11}$  : Động năng tịnh tiến link1
- $K_{12}$  : Động năng quay link1
- $K_{21}$  : Động năng tịnh tiến link2
- $K_{22}$  : Động năng quay link2

### 3.3.2 Phương trình toán

Sử dụng phương trình động lực học Lagrangian và hệ trục tọa độ như (Hình 3.2) để thiết lập mô hình toán cho hệ thống Pendubot, phương trình được diễn tả dưới dạng vector như sau:

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau \quad (4.4)$$

Diễn tả theo phương trình trên hệ số ma sát đã được bỏ qua. Trong phương trình trên  $\tau$  là lực xoắn được áp vào link 1 và  $q$  là vector vị trí góc.  $D(q)$  là ma trận quán tính,  $C(q, \dot{q})$  là ma trận Coriolis và  $G(q)$  là vector trọng lực.

Theo phương trình động lực học EULER\_LAGRANGIAN ta có:

$$\frac{d}{dt} \left[ \frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} = \tau$$

Với  $L = K - V$

Theo hệ trục tọa độ của hệ thống động năng  $K$  của hệ thống được tính như sau:

$$K = K_1 + K_2$$

Với:  $K_1 = k_{11} + k_{12}$

$$K_{11} = \frac{1}{2} m_1 v_1^2$$

$$K_{12} = \frac{1}{2} I_1 \dot{q}_1^2$$

$$\rightarrow K_1 = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} I_1 \dot{q}_1^2$$

Với  $K_2 = k_{21} + k_{22}$

$$K_{21} = \frac{1}{2} m_2 v_2^2$$

$$K_{22} = \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2$$

$$\rightarrow K_2 = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2$$

Ta có:  $v_1 = \begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix}$  và  $v_2 = \begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix}$

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2$$

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$$

$$x_1 = l_{c1} \cos(q_1)$$



$$y_1 = l_{c1} \sin(q_1)$$

$$x_2 = l_1 \cos(q_1) + l_{c1} \cos(q_1 + q_2)$$

$$y_2 = l_2 \sin(q_1) + l_{c2} \sin(q_1 + q_2)$$

$$\dot{x}_1 = -l_{c1} \sin(q_1)$$

$$\dot{y}_1 = l_{c1} \cos(q_1)$$

$$\dot{x}_2 = -l_1 \sin(q_1) - l_{c1}(\dot{q}_1 + \dot{q}_2) l_{c2} \sin(q_1 + q_2)$$

$$\dot{y}_2 = l_2 \cos(q_1) + l_{c2}(\dot{q}_1 + \dot{q}_2) l_{c2} \cos(q_1 + q_2)$$

Tổng động năng của của hai link

$$\begin{aligned} K &= \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 \\ &= \frac{1}{2} m_1 ((-l_{c1} \sin(q_1))^2 + (l_{c1} \cos(q_1))^2) + \frac{1}{2} m_2 (-l_1 \sin(q_1) - l_{c1}(\dot{q}_1 + \dot{q}_2) l_{c2} \sin(q_1 + q_2))^2 \\ &\quad + (l_2 \cos(q_1) + l_{c2}(\dot{q}_1 + \dot{q}_2) l_{c2} \cos(q_1 + q_2))^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \\ &\quad \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 \end{aligned}$$

Thế năng của hệ thống

$$\begin{aligned} V &= m_1 y_1 g + m_2 y_2 g = m_1 l_{c1} \sin(q_1) g + m_2 l_2 \sin(q_1) + l_{c2} \sin(q_1 + q_2) g \\ \Rightarrow L &= K - V = \frac{1}{2} m_1 ((-l_{c1} \sin(q_1))^2 + (l_{c1} \cos(q_1))^2) + \frac{1}{2} m_2 (-l_1 \sin(q_1) - l_{c1}(\dot{q}_1 + \dot{q}_2) l_{c2} \sin(q_1 + q_2))^2 \\ &\quad + (l_2 \cos(q_1) + l_{c2}(\dot{q}_1 + \dot{q}_2) l_{c2} \cos(q_1 + q_2))^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2 - (m_1 l_{c1} \sin(q_1) g + m_2 l_2 \sin(q_1) + l_{c2} \sin(q_1 + q_2) g) \end{aligned}$$

Tính toán theo công thức (3.1) và đưa về dạng ma trận

Viết dưới dạng ma trận:  $D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$

Với  $\tau = \begin{bmatrix} \tau_m \\ 0 \end{bmatrix}$

Động năng của hệ thống

$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q}$$

$$\text{Với } D(q) = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \quad (4.5)$$

$$d_{11} = m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_1 + I_2$$

$$d_{12} = m_2 (l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_2 \quad (4.6)$$

$$d_{21} = m_2 (l_{c2}^2 + 2l_1 l_{c2} \cos q_2) + I_2$$

$$d_{22} = m_2 l_{c2}^2 + I_2$$

$$C(q, \dot{q}) = \begin{bmatrix} h\dot{q}_2 & h\dot{q}_2 + h\dot{q}_1 \\ -h\dot{q}_2 & 0 \end{bmatrix} \quad (4.7)$$

$$\text{Với } h = -m_2 l_1 l_{c2} \sin q_2 \quad (4.8)$$

$$g(q) = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix}$$

$$\phi_1 = (m_1 l_{c1} + m_2 l_1) g \cos q_1 + m_2 l_{c2} g \cos(q_1 + q_2)$$

$$\phi_2 = m_2 l_{c2} g \cos(q_1 + q_2) \quad (4.9)$$

Ta đặt thông số hệ thống như sau:

$$\beta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$$

$$\beta_2 = m_2 l_{c2}^2 + I_2 \quad (4.10)$$

$$\beta_3 = m_2 l_1 l_{c2}$$

$$\beta_4 = m_1 l_{c1} + m_2 l_1$$

$$\beta_5 = m_2 l_{c2}$$

Thế những thông số trên vào các ma trận ma trận  $D(q)$ ,  $C(q, \dot{q})$ ,  $G(q)$  :

$$D(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos q_2 & \beta_2 + 2\beta_3 \cos q_2 \\ \beta_2 + 2\beta_3 \cos q_2 & \beta_2 \end{bmatrix} \quad (4.11)$$

$$C(q, \dot{q}) = \begin{bmatrix} -\beta_3 \sin(q_2) \dot{q}_2 & -\beta_3 \sin(q_2) \dot{q}_2 - \beta_3 \sin(q_2) \dot{q}_1 \\ \beta_3 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \quad (4.12)$$

$$g(q) = \begin{bmatrix} \beta_4 g \cos q_1 + \beta_5 g \cos(q_1 + q_2) \\ \beta_5 g \cos(q_1 + q_2) \end{bmatrix} \quad (4.13)$$

### 3.4 Giới thiệu phương pháp nhận dạng thông số.

#### 3.4.1 Phương pháp năng lượng.

Sau khi tìm được mô hình toán của hệ thống Pendubot, bước tiếp theo là tìm là chúng ta tìm năm thông số của mô hình trong phương trình (4.10).

Có nhiều phương pháp nhận dạng thông số, trong luận văn này giới thiệu phương pháp nhận diện bằng năng lượng và phương pháp tính trực tiếp.

Chương trình này nhận dạng on-line sử dụng định lý năng lượng để hình thành các phương trình có thể được giải các tham số chưa biết bằng phương pháp bình phương tối thiểu.

Động năng Pendubot là:

$$K = \frac{1}{2} \dot{q}^T D(q) \dot{q}$$

$$\text{Với } D(q) = \begin{bmatrix} \beta_1 + \beta_2 + 2\beta_3 \cos q_2 & \beta_2 + 2\beta_3 \cos q_2 \\ \beta_2 + 2\beta_3 \cos q_2 & \beta_2 \end{bmatrix} \quad (4.14)$$

Từ hai phương trình trên động năng Pendubot được viết lại như sau

$$K = \frac{1}{2} \dot{q}_1^2 \beta_1 + (\frac{1}{2} \dot{q}_1^2 + \dot{q}_1 \dot{q}_2 + \frac{1}{2} \dot{q}_2^2) \beta_2 + (\cos q_2 \dot{q}_1^2 + \cos q_2 \dot{q}_1 \dot{q}_2) \beta_3$$

Thế năng của Pendubot

$$V = \beta_4 g \sin q_1 + \beta_5 g \sin(q_1 + q_2)$$

Từ các phương trình ở trên ta thấy rằng các phương trình năng lượng động năng và thế năng cả hai phương trình là tuyến tính trong các thông số quán tính. Một cách đơn giản để viết những phương trình này là

$$\begin{aligned} K &= \sum_{i=1}^5 \frac{\partial K}{\partial \beta_i} \beta_i = \sum_{i=1}^5 AK_i \beta_i \\ V &= \sum_{i=1}^5 \frac{\partial V}{\partial \beta_i} \beta_i = \sum_{i=1}^5 AV_i \beta_i. \end{aligned} \quad (4.15)$$

Pendubot với những biến mới là:

$$\begin{aligned} AK_1 &= \frac{1}{2} \dot{q}_1^2 \\ AK_2 &= \frac{1}{2} \dot{q}_1^2 + \dot{q}_1 \dot{q}_2 + \frac{1}{2} \dot{q}_2^2 \\ AK_3 &= \cos q_2 \dot{q}_1^2 + \cos q_2 \dot{q}_1 \dot{q}_2, \\ AK_4 &= 0 \\ AK_5 &= 0 \end{aligned} \quad (4.16)$$

$$\begin{aligned} AV_1 &= 0 \\ AV_2 &= 0 \\ AV_3 &= 0 \\ AV_4 &= g \sin q_1 \\ AV_5 &= g \sin(q_1 + q_2) \end{aligned} \quad (4.17)$$

Định lý năng lượng với trạng thái mà việc lực được áp vào hệ thống bằng với sự thay đổi tổng năng lượng của tổng hệ thống biểu thức như sau:

$$\int_{t_1}^{t_2} \mathbf{T}^T \dot{\mathbf{q}} dt = (K(t_2) + V(t_2)) - (K(t_1) + V(t_1)) = L(t_2) - L(t_1)$$

Với  $L(t_i)$  là tổng năng lượng thời gian  $t_i$ ,  $L(t_i) = K(t_i) + V(t_i)$ , và  $\mathbf{T}$  là vector của moment được áp vào tại khớp.  $\mathbf{T}$  bao gồm moment motor và lực ma sát

$$\mathbf{T} = \mathbf{t} + \mathbf{G}_f.$$

Trong nghiên cứu này ta bỏ qua lực ma sát và phương trình viết lại

$$\mathbf{T} = \mathbf{t}$$

Với  $K$  và  $V$  là tuyến tính trong thông số quán tính, khác nhau trong tổng năng lượng

$$L(t_2) - L(t_1) = \mathbf{A}\mathbf{L}^T \boldsymbol{\beta}$$

Với  $\mathbf{A}\mathbf{L}^T = [DL_1(t_2)-DL_1(t_1) \dots DL_5(t_2)-DL_5(t_1)]$

Và  $DL_i(t_k) = DK_i(t_k) + DV_i(t_k)$ .

Phương trình năng lượng với dạng sau

$$\int_{t_1}^{t_2} \mathbf{T}^T \dot{\mathbf{q}} dt = \mathbf{A}\mathbf{L}^T \boldsymbol{\beta}.$$

Định nghĩa vector mới  $d$ ,  $d^T = \mathbf{A}\mathbf{L}^T$ , phương trình  $K^{\text{th}}$  liên hệ đến khoảng thời gian  $(t_{K-1}, t_K)$  được viết như sau

$$\left( \int_{t_{K-1}}^{t_K} \mathbf{T}^T \dot{\mathbf{q}} dt \right)_K = d_K^T \boldsymbol{\beta}.$$

Phương trình  $k$  có thể được kết nối vào trong tiêu chuẩn trên xác định phương trình ma trận,  $Ax=b$ , và được giải bằng kỹ thuật bình phương tối thiểu. Từ  $AL_i(0)=0$  for  $(i=1, \dots, 5)$ , phương trình có thể được viết lại như sau

$$\left( \int_{t_0}^{t_K} \mathbf{T}^T \dot{\mathbf{q}} dt \right)_K = d_K^T \boldsymbol{\beta}$$

Với phương trình  $K^{\text{th}}$  bây giờ cho trong khoảng thời gian  $(t_0, t_K)$ .

Thực thi phương pháp nhận dạng này chúng ta viết chương trình điều khiển Pendubot với tín hiệu vòng lặp hở tại thời gian được ghi tương ứng của hệ thống. Với những phương trình có được dùng Matlab để nhận dạng

Chương trình nhận dạng bằng matlab:

%q1, dq1, q2 và dq2 là vector vị trí và vận tốc

g=386; (SI Units = 9.8)

dL1 = (.5\*dq1.^2);

dL2 = (.5\*dq1.^2 + dq1.\*dq2 + .5\*dq2.^2);

dL3 = (cos(q2).\*(dq1.^2 + dq1.\*dq2));

```

dL4 = (g*sin(q1));
dL5 = (g*sin(q1+q2));
taudq1 = tau.*dq1;    %tau điều khiển vòng lặp hở
for i = 1:(length(dL1)-10),
DL(i,1) = dL1(i+10)-dL1(1);
DL(i,2) = dL2(i+10)-dL2(1);
DL(i,3) = dL3(i+10)-dL3(1);
DL(i,4) = dL4(i+10)-dL4(1);
DL(i,5) = dL5(i+10)-dL5(1);
Itq(i,1) = trapz(t(1:i+10,1),taudq1(1:i+10,1));
end

Peta = lsqnonneg(DL,Itq) %non-negative least squares solution to Ax=b.

```

### 3.4.2 Phương pháp đo trực tiếp các thông số của hệ thống

Những thông số sau là cần thiết diễn tả hệ thống:

- Những thông số của các link có thể đo trực tiếp hoặc được tính bằng công thức từng thành phần riêng.

- Momemt của mỗi link có thể được tính bằng công thức như sau: Là momemt của từng thành phần trên link đó

$$I_1 = I_{link1} + I_{sensor}$$

$$I_{link1} = 1/12 m_{link1} (l_1^2 + w_1^2) = 1/12 \times 0.005 \times (0.2^2 + 0.003^2) = 1.667 \cdot 10^{-4} (\text{kg.m}^2)$$

$$I_{sensor} = M_{sensor} L_{sensor}^2 = 0.17 \times 0.15^2 = 3.825 \times 10^{-3} (\text{kg.m}^2)$$

$$I_1 = I_{link1} + I_{sensor} = 1.667 \cdot 10^{-4} + 3.825 \times 10^{-3} = 3.9917 \times 10^{-3} (\text{kg.m}^2)$$

- Ta cũng tính được  $I_2 = 6.49 \times 10^{-4} (\text{kg.m}^2)$

Với các thông số ta đo trực tiếp hoặc tính theo công thức ta được:

$m_1$ (kg)	0.22
$l_1$ (m)	0.187
$l_{c1}$ (m)	0.15
$I_1$ (kg.m <sup>2</sup> )	$3.9917 \times 10^{-3}$
$m_2$ (kg)	0.038
$l_2$ (m)	0.325
$l_{c2}$ (m)	0.205
$I_2$ (kg.m <sup>2</sup> )	$6.49 \times 10^{-4}$
$M_{\text{sensor}}$ (kg)	0.17
$L_{\text{sensor}}$ (m)	0.15

Thế vào các thông số:

$$\beta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1 = 0.01122$$

$$\beta_2 = m_2 l_{c2}^2 + I_2 = 0.00193$$

$$\beta_3 = m_2 l_1 l_{c2} = 0.00156$$

$$\beta_4 = m_1 l_{c1} + m_2 l_1 = 0.0406$$

$$\beta_5 = m_2 l_{c2} = 0.0078$$

### 3.5 Diễn tả không gian trạng thái hệ thống

Bởi vì ma trận  $D(q)$  là ma trận định nghĩa dương, ma trận ngược  $D(q)^{-1}$  của nó là tồn tại

Hệ thống phương trình hoạt động có thể được viết lại như sau:

$$\begin{aligned}
\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} &= D(q)^{-1} \tau - D(q)^{-1} C(q, \dot{q}) \dot{q} - D(q)^{-1} g(q) \\
x_1 &= q_1, x_2 = \dot{q}_1, x_3 = q_2, x_4 = \dot{q}_2 \\
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= \ddot{q}_1 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \ddot{q}_2
\end{aligned} \tag{4.18}$$

Tính các thành phần trong (4.14) :

$$\begin{aligned}
D(q)^{-1} &= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} \begin{bmatrix} \beta_2 & -\beta_2 - \beta_3 \cos q_2 \\ -\beta_2 - \beta_3 \cos q_2 & \beta_1 + \beta_2 + 2\beta_3 \cos q_2 \end{bmatrix} \\
D(q)^{-1} \tau &= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} \begin{bmatrix} \beta_2 \tau_m \\ (-\beta_2 - \beta_3 \cos q_2) \tau_m \end{bmatrix} \\
D(q)^{-1} C(q, \dot{q}) \dot{q} &= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} \begin{bmatrix} \beta_2 & -\beta_2 - \beta_3 \cos q_2 \\ -\beta_2 - \beta_3 \cos q_2 & \beta_1 + \beta_2 + 2\beta_3 \cos q_2 \end{bmatrix} \times \\
&\quad \begin{bmatrix} -\beta_3 \sin(q_2) \dot{q}_2 & -\beta_3 \sin(q_2) \dot{q}_2 - \beta_3 \sin(q_2) \dot{q}_1 \\ \beta_3 \sin(q_2) \dot{q}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} \\
&= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} \times \begin{bmatrix} -\beta_2 \beta_3 \sin(q_2) (\dot{q}_2 + \dot{q}_1)^2 - \beta_3^2 \sin(q_2) \cos(q_2) \dot{q}_1^2 \\ \beta_3 \sin(q_2) (\beta_2 + \beta_3 \cos q_2) (\dot{q}_2 + \dot{q}_1)^2 + \beta_2 \beta_3 \sin(q_2) \dot{q}_1^2 \end{bmatrix} \\
D(q)^{-1} g(q) &= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} \begin{bmatrix} \beta_2 & -\beta_2 - \beta_3 \cos q_2 \\ -\beta_2 - \beta_3 \cos q_2 & \beta_1 + \beta_2 + 2\beta_3 \cos q_2 \end{bmatrix} \times \\
&\quad \begin{bmatrix} \beta_4 g \cos q_1 + \beta_5 g \cos(q_1 + q_2) \\ \beta_5 g \cos(q_1 + q_2) \end{bmatrix} \\
&= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} \begin{bmatrix} \beta_2 \beta_4 g \cos q_1 - \beta_2 \beta_5 g \cos q_2 \cos(q_1 + q_2) \\ \beta_5 g (\beta_1 + \beta_3 \cos q_2) \cos(q_1 + q_2) - \beta_4 g (\beta_2 + \beta_3 \cos q_2) \cos q_1 \end{bmatrix} \\
\text{Với } \ddot{q} &= \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} \\
\ddot{q}_1 &= \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 q_2} [\beta_2 \tau_m + \beta_2 \beta_3 \sin(q_2) (\dot{q}_2 + \dot{q}_1)^2 + \\
&\quad \beta_3^2 (\sin q_2) (\cos q_2) \dot{q}_1^2 - \beta_2 \beta_4 \cos q_1 + \beta_3 \beta_5 \cos q_2 \cos(q_1 + q_2)] \tag{4.19}
\end{aligned}$$



$$\ddot{q}_2 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} [-\beta_2 \tau_m - \tau_m \beta_3 \cos(q_2) - \beta_3 \sin(q_2)(\beta_2 + \beta_3 \cos q_2)(\dot{q}_2 + \dot{q}_1)^2 - (\beta_1 + \beta_3 \cos q_2)[\beta_2 (\sin q_2) \dot{q}_1^2 - \beta_5 g \cos(q_1 + q_2)] + \beta_4 g \cos q_1 (\beta_2 + \beta_3 \cos q_2)] \quad (4.20)$$

Biến trạng thái định nghĩa lại như sau:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \ddot{q}_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \ddot{q}_2 \end{aligned} \quad (4.21)$$

Phương trình không gian trạng thái được diễn tả như sau:

$$\dot{x} = Ax + Bu \quad (4.22)$$

Với

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_3} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_3} & 0 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix}$$

$$f_1(x) = x_2$$

$$f_2(x) = \frac{1}{\beta_1\beta_2 + \beta_3^2 \cos^2 x_3} \left[ \beta_2 \tau_m + \beta_2 \beta_3 + \sin(x_3)(x_2 + x_4)^2 + \beta_3^2 (\sin(x_3))x_2^2 \right] \\ - \beta_2 \beta_4 \cos x_1 + \beta_3 \beta_5 \cos x_3 \cos(x_1 + x_3)$$

$$f_3(x) = x_3$$

$$f_4(x) = \frac{1}{\beta_1 \beta_2 + \beta_3^2 \cos^2 x_3} \left[ -\beta_2 \tau_m - \tau_m \beta_3 \cos(x_3) - \beta_3 (\sin(x_3)(\beta_2 + \beta_3 \cos x_3 (x_2 + x_4)^2 - (\beta_1 + \beta_3 \cos x_3) [\beta_3 (\sin x_3 \cos(x_1 + x_3))] + \beta_4 g \cos x_1 (\beta_2 + \beta_3 \cos x_3)) \right]$$

$$f_1(x) = 0$$

$$f_3(x) = 0$$

$$f_2(x) = \frac{\beta_3}{\beta_1 \beta_2 - \beta_3^2 \cos^2 x_3}$$

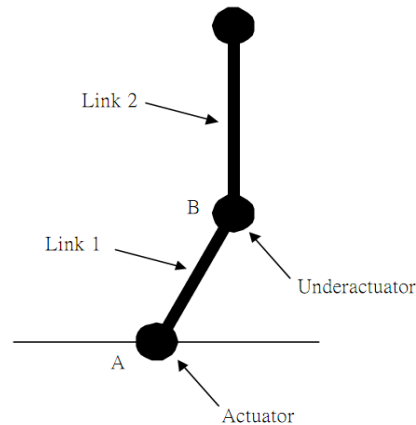
$$f_4(x) = \frac{\beta_3 + \beta_3 \cos x_3}{\beta_1 \beta_2 - \beta_3^2 \cos^2 x_3} \quad (4.23)$$

Các thông số từ  $\beta_1$  đến  $\beta_5$  được định nghĩa như u moment tín hiệu vào điều khiển và  $x_1$  đến  $x_4$  là trạng thái của hệ thống Pendubot.

### 3.6 Những điểm cân bằng của hệ thống

Nhiệm vụ trong luận văn này là điều khiển cân bằng hệ thống nên những điểm cân bằng được tìm là cần thiết.

Hệ thống máy có cơ cấu chấp hành bên dưới (underactuator) như (Hình 3.3) tạo nên những điểm cân bằng dựa trên cả thông số động năng và thế năng của chúng. Nếu hệ thống Pendubot được thiết lập song song với gia tốc trọng trường hệ thống có những điểm cân bằng như (Hình 3.4)



Hình 3.3 Cơ cấu chấp hành và cơ cấu chấp hành bên dưới của Pendubot

Hệ thống cân bằng khi :

$$\ddot{q}_1 = \ddot{q}_2 = 0;$$

$$\dot{q}_1 = \dot{q}_2 = 0; \text{ và } \tau_m = 0;$$

Từ phương trình (4.10), (4.11), (4.12), (4.13) và (4.18 )

Chúng ta tìm được điểm cân bằng

$$\begin{cases} \beta_4 g \cos q_1 + \beta_5 g \cos(q_1 + q_2) = 0 \\ \beta_5 g \cos(q_1 + q_2) = 0 \end{cases}$$

Với  $q_1$  và  $q_2 \in [-\pi, \pi]$

Ta có:  $\beta_4 g \cos q_1 = 0 \rightarrow \cos q_1 = 0$

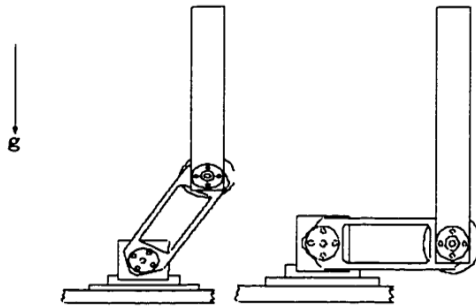
Khi  $q_1 = \pi/2$  và  $q_1 = -\pi/2$

$$\beta_5 g \cos(q_1 + q_2) = 0 \rightarrow \cos(q_1 + q_2) = 0$$

Khi  $q_2 = 0$  và  $q_2 = \pi$

Như vậy hệ thống có 4 điểm cân bằng tương ứng  $(q_1, \dot{q}_1, q_2, \dot{q}_2)$

$(\pi/2, 0, 0, 0)$  ;  $(-\pi/2, 0, \pi, 0)$  ;  $(-\pi/2, 0, 0, 0)$  ;  $(\pi/2, 0, \pi, 0)$



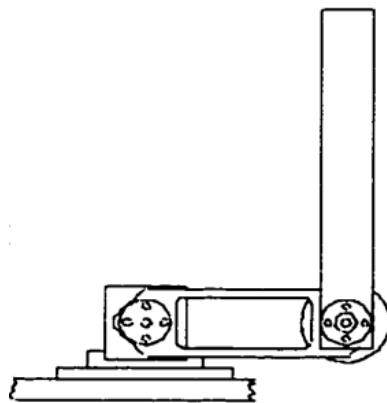
Hình 3.4 Những điểm cân bằng có thể

### 3.7 Tính chất có thể điều khiển được và không điều khiển được

Những điểm cân bằng trên là có thể điều khiển được. Tuy nhiên ta có 4 vị trí không thể điều khiển được như là:

$$q_1 = 0, q_2 = \pi/2 \text{ hoặc là } -\pi/2 \text{ và } q_1 = -\pi, q_2 = \pi/2 \text{ hoặc là } -\pi/2$$

Tuy nhiên, chúng ta dễ dàng hiểu tính chất vật lý của hệ thống vì sao không thể điều khiển được khi  $q_1 = 0, \pi, -\pi$  được mô phỏng như (Hình 3).



Hình 3.5 Đây là vị trí không điều khiển được

Định nghĩa ma trận có thể điều khiển được như sau:

$$C_n = [ B \quad AB \quad A^2B \quad A^3B ]$$

Ma trận A và B được học viên định nghĩa ở phần sau (. Phương trình A.4 và A.9 tương ứng, sau khi thế những tham số hệ thống và những điểm cân bằng

$$q_1 = 0, \quad q_2 = \pi/2 \text{ hoặc là } -\pi/2 \text{ và } \dot{q}_1 = -\pi, \quad \dot{q}_2 = \pi/2 \text{ hoặc là } -\pi/2$$

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_2} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_2} & 0 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix}$$

Thế những điểm cân bằng trên vào 2 ma trận A và B ta được

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 26.0265 & 0 & -7.1909 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 89.1266 \\ 0 \\ -89.1266 \end{bmatrix}$$

Từ ma trận A và B ta tính được ma trận Cn như sau:

$$Cn = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 89.1266 & 89.1266 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -89.1266 & -89.1266 & 0 & 0 \end{bmatrix}$$

Ma trận Cn có hạng là  $2 < 4$  hạng của ma trận A, do vậy những điểm cân bằng trên là không điều khiển được như (Hình 3.5)

## CHƯƠNG 4: THIẾT KẾ BỘ ĐIỀU KHIỂN

### 4.1 Thiết kế điều khiển hệ thống Pendubot dùng giải thuật LQR

#### 4.1.1 Tuyến tính hóa phương trình chuyển động

Để thiết kế điều khiển cân bằng phương trình phi tuyến (4.19) của Pendubot ta tiến hành tuyến tính hóa tại điểm làm việc của hệ thống.

Phương trình biến trạng thái có dạng:

$$\dot{x} = Ax + Bu$$

$$y = Cx$$

Ma trận Jacobian A, B, C có dạng như sau:

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_3} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_3} & 0 \end{bmatrix}$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} = D(q)^{-1} \tau - D(q)^{-1} C(q, \dot{q}) \dot{q} - D(q)^{-1} g(q)$$

$$x_1 = q_1, x_2 = \dot{q}_1, x_3 = q_2, x_4 = \dot{q}_2$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \ddot{q}_1$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \ddot{q}_2$$

Phương pháp xấp xỉ tuyến tính dựa trên sự khai triển hàm phi tuyến theo chuỗi Taylor xung quanh điểm làm việc và sự duy trì các điều kiện tuyến tính. Đối với n biến  $x_1, x_2, \dots, x_n$  có thể biểu diễn ngắn gọn như sau:

$$y = f(x_1, x_2, \dots, x_n)$$

$$\bar{y} = f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$$

$$y - \bar{y} = \frac{\partial f}{\partial x_1} \bigg|_{\substack{x_1=\bar{x}_1 \\ x_2=\bar{x}_2 \\ \dots \\ x_n=\bar{x}_n}} (x_1 - \bar{x}_1) + \frac{\partial f}{\partial x_2} \bigg|_{\substack{x_1=\bar{x}_1 \\ x_2=\bar{x}_2 \\ \dots \\ x_n=\bar{x}_n}} (x_2 - \bar{x}_2) + \dots + \frac{\partial f}{\partial x_n} \bigg|_{\substack{x_1=\bar{x}_1 \\ x_2=\bar{x}_2 \\ \dots \\ x_n=\bar{x}_n}} (x_n - \bar{x}_n)$$

Ta có các điểm cân bằng sau:

$$(\pi/2, 0, 0, 0); (-\pi/2, 0, \pi, 0); (-\pi/2, 0, 0, 0); (\pi/2, 0, \pi, 0)$$

Theo định nghĩa biến trạng thái như sau:

$$x = [x_1 \ x_2 \ x_3 \ x_4]' = [q_1 \ \dot{q}_1 \ q_2 \ \dot{q}_2]'$$

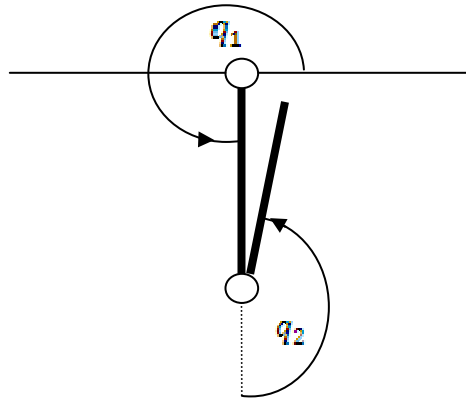
$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 x_3} [\beta_2 \tau_m + \beta_2 \beta_3 \sin(x_3) (x_2 + x_4)^2 + \beta_3^2 (\sin(x_3)) (\cos(x_3)) x_2^2 - \beta_2 \beta_4 \cos x_1 + \beta_3 \beta_5 \cos x_3 \cos(x_1 + x_3)]$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{1}{\beta_1 \beta_2 - \beta_3^2 \cos^2 x_3} [-\beta_2 \tau_m - \tau_m \beta_3 \cos(x_3) - \beta_3 \sin(x_3) (\beta_2 + \beta_3 \cos x_3) (x_2 + x_4)^2 - (\beta_1 + \beta_3 \cos x_3) [\beta_3 (\sin x_3) \dot{q}_1^2 - \beta_5 g \cos(x_1 + x_3)] + \beta_4 g \cos x_1 (\beta_2 + \beta_3 \cos x_3)]$$

#### 4.1.2 Điểm cân bằng vị trí MID:



Hình 5.1 : Vị trí cân bằng MID

Tại vị trí cân bằng Mid  $(-\pi/2, 0, \pi, 0)$ , vì vậy ta có:

$$x_1 = q_1 + \pi/2$$

$$x_2 = \dot{q}_1$$

$$x_3 = q_2 - \pi$$

$$x_4 = \dot{q}_2$$

Thế các biến trên vào phương trình phi tuyến sau đó lấy đạo hàm để tìm ma trận A và B như sau:

**Phương trình tuyến tính hóa vị trí cân bằng Mid  $(-\pi/2, 0, \pi, 0)$**

□

$$x_2 = \dot{x}_1$$

$$\dot{x}_2 = \frac{1}{\beta_1\beta_2 - \beta_2^2 \cos^2 x_3} [\beta_2 T + \beta_2\beta_3 \sin(x_3)(x_4 + x_2)^2 + \beta_3^2 \sin(x_3)\cos(x_3)x_2^2 - \beta_2\beta_4 g \cos(x_1) + \beta_3\beta_5 g \cos(x_3)\cos(x_1 + x_3)]$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{1}{\beta_1\beta_2 - \beta_2^2 \cos^2 x_3} [-\beta_2 T - T\beta_3 \cos(x_3) - \beta_3 \sin(x_3)[(\beta_2 + \beta_3 \cos(x_3))(x_4 + x_2)^2 + (\beta_1 + \beta_3 \cos(x_3))x_2^2] - \beta_5 g \cos(x_1 + x_3)(\beta_1 + \beta_3 \cos(x_3)) + \beta_4 g \cos(x_1)(\beta_2 + \beta_3 \cos(x_3))]$$



$$y - \bar{y} = \frac{\mathcal{F}}{\partial x} \bigg|_{\substack{x_1 = \bar{x}_1 \\ x_2 = \bar{x}_2 \\ x_n = \bar{x}_n}} (x_1 - \bar{x}_1) + \frac{\mathcal{F}}{\partial u} \bigg|_{\substack{x_1 = \bar{x}_2 \\ x_2 = \bar{x}_2 \\ x_n = \bar{x}_n}} (x_2 - \bar{x}_2) + \dots + \frac{\mathcal{F}}{\partial u} \bigg|_{\substack{x_1 = \bar{x}_2 \\ x_2 = \bar{x}_2 \\ x_n = \bar{x}_n}} (x_n - \bar{x}_n) \quad (\text{A.1})$$

$$u_r = \theta_4 g \cos(x_{r1}) \quad (\text{A.2})$$

$$x_{r1} + x_{r3} = \pi/2 \quad (\text{A.3})$$

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{\partial f_2}{\partial x_1} & 0 & \frac{\partial f_2}{\partial x_2} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\partial f_4}{\partial x_1} & 0 & \frac{\partial f_4}{\partial x_2} & 0 \end{bmatrix} \quad (\text{A.4})$$

Vói

$$\frac{\partial f_2}{\partial x_1} = \frac{g[\beta_2 \beta_4 \sin(x_1) - \beta_2 \beta_5 \cos x_2 \sin(x_1 + x_2)]}{\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2} \quad (\text{A.5})$$

$$\frac{\partial f_2}{\partial x_2} = \frac{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)[\beta_2 \beta_3 \cos x_2 (x_4 + x_2)^2 + \beta_2^2 \cos(2x_2) x_2^2 - \beta_2 \beta_3 g \sin(x_1 + 2x_2)]}{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)^2} \quad (\text{A.6})$$

$$- \frac{\beta_2^2 \sin(2x_2)[\beta_2 T + \beta_2 \beta_3 \sin(x_2)(x_4 + x_2)^2 + \beta_2^2 \sin(x_2) \cos(x_2) x_2^2 - \beta_2 \beta_4 g \cos(x_1) + \beta_2 \beta_3 g \cos(x_2) \cos(x_1 + x_2)]}{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)^2}$$

$$\frac{\partial f_4}{\partial x_1} = \frac{g\beta_3 \sin(x_1 + x_2)[\beta_1 + \beta_2 \cos x_2] - [\beta_2 + \beta_2 \cos x_2]\beta_4 g \sin(x_1)}{\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2} \quad (\text{A.7})$$

$$\frac{\partial f_4}{\partial x_2} = \frac{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)[\beta_2 T \sin(x_2) - \beta_2 \cos(x_2)](x_4 + x_2)^2 (\beta_2 + \beta_2 \cos x_2) + (\beta_1 + \beta_2 \cos x_2) x_2^2}{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)^2} \\ + \frac{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)[\beta_2^2 \sin^2 x_2 [(x_4 + x_2)^2 + x_2^2] + \beta_2 g [\beta_1 \sin(x_1 + x_2) + \sin(x_1 + 2x_2)] - \beta_2 \beta_4 g \sin(2x_2) \cos x_2]}{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)^2}$$

$$- \frac{\beta_2^2 \sin(2x_2)[-\beta_2 T - T \beta_2 \cos(x_2) - \beta_2 \sin(x_2)[(\beta_2 + \beta_2 \cos(x_2))(x_4 + x_2)^2 + (\beta_1 + \beta_2 \cos(x_2))x_2^2] - \beta_2 g \cos(x_1 + x_2)(\beta_1 + \beta_2 \cos(x_2)) + \beta_4 g \cos(x_1)(\beta_2 + \beta_2 \cos(x_2))]}{(\beta_1 \beta_2 - \beta_2^2 \cos^2 x_2)^2} \quad (\text{A.8})$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ \frac{\partial f_2}{\partial u} \\ 0 \\ \frac{\partial f_4}{\partial u} \end{bmatrix} \quad (\text{A.9})$$

$$\begin{aligned}\frac{\partial f_2}{\partial u} &= \frac{-\beta_2}{\beta_1\beta_2 - \beta_3^2 \cos^2 x_2} \\ \frac{\partial f_4}{\partial u} &= \frac{\beta_2 + \beta_3 \cos x_2}{\beta_1\beta_2 - \beta_3^2 \cos^2 x_2}\end{aligned}\tag{A.10}$$

Vậy ta có:

$$\frac{\partial f_2}{\partial x_1} = (1/(\beta_1\beta_2 - \beta_3^2))g(\beta_3\beta_5 - \beta_2\beta_4)$$

$$\frac{\partial f_2}{\partial x_1} = (1/(\beta_1\beta_2 - \beta_3^2))g(\beta_3\beta_5 g x_3)$$

$$\frac{\partial f_4}{\partial x_1} = (1/(\beta_1\beta_2 - \beta_3^2))g(\beta_5(\beta_1 - \beta_3) - g^*\beta_4(\beta_2 - \beta_3))$$

$$\frac{\partial f_4}{\partial x_3} = (1/(\beta_1\beta_2 - \beta_3^2))g^*\beta_5(\beta_1 - \beta_3)x_3$$

$$\frac{\partial f_2}{\partial u} = \beta_2/(\beta_1\beta_2 - \beta_3^2)$$

$$\frac{\partial f_2}{\partial u} = (\beta_2 - \beta_3)/(\beta_1\beta_2 - \beta_3^2)$$

$m_1$ (kg)	0.22
$l_1$ (m)	0.187
$l_{c1}$ (m)	0.15
$I_1$ (kg.m <sup>2</sup> )	0.00475
$m_2$ (kg)	0.038
$l_2$ (m)	0.325
$l_{c2}$ (m)	0.205
$I_2$ (kg.m <sup>2</sup> )	6.49x10 <sup>-4</sup>

$$\text{Và } \beta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1$$

$$\beta_2 = m_2 l_{c2}^2 + I_2$$

$$\beta_3 = m_2 l_1 l_{c2}$$

$$\beta_4 = m_1 l_{c1} + m_2 l_1$$

$$\beta_5 = m_2 l_{c2}$$

Thế các tham số trên vào phương trình ta tìm được các peta:

$$\beta_1 = 0.01122$$

$$\beta_2 = 0.00193$$

$$\beta_3 = 0.00156$$

$$\beta_4 = 0.0406$$

$$\beta_5 = 0.0078$$

Vậy ta tìm được các ma trận sau:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 33.7820 & 0 & -6.2103 & 0 \\ 0 & 0 & 0 & 1 \\ -46.1230 & 0 & -38.4561 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 100.4110 \\ 0 \\ -19.2498 \end{bmatrix}$$

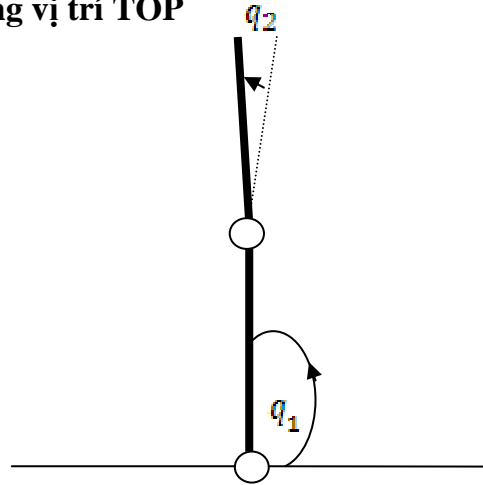
Ma trận  $Q = \begin{bmatrix} 100 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}$

$$R = 1$$

Với  $K = \text{lqr}(A, B, Q, R)$

$$K = [38.3174 \quad 3.8413 \quad 17.9151 \quad -4.1102]$$

#### 4.1.3 Điểm cân bằng vị trí TOP



Hình 4.2 Vị trí cân bằng Top

Tại vị trí cân bằng TOP  $(\pi/2, 0, 0, 0)$ , vì vậy ta có:

$$x_1 = q_1 - \pi/2$$

$$x_2 = q_1$$

$$x_3 = q_2$$

$$x_4 = q_2$$

Thế các biến trên vào phương trình phi tuyến sau đó lấy đạo hàm để tìm ma trận A và B như Phương trình tuyến tính hóa vị trí cân bằng Mid  $(-\pi/2, 0, \pi, 0)$ .

Ta có:

$$\frac{\partial f_2}{\partial x_1} = (1/(\beta_1\beta_2 - \beta_3^2))g(-\beta_3\beta_5 + \beta_2\beta_4)$$

$$\frac{\partial f_2}{\partial x_1} = -(1/(\beta_1\beta_2 - \beta_3^2))g(\beta_3\beta_5gx_3)$$

$$\frac{\partial f_4}{\partial x_1} = (1/(\beta_1\beta_2 - \beta_3^2))g(\beta_5(\beta_1 - \beta_3) - g^*\beta_4(\beta_2 - \beta_3))$$

$$\frac{\partial f_4}{\partial x_3} = (1/(\beta_1\beta_2 - \beta_3^2))g^*\beta_5(\beta_1 - \beta_3)x_3$$

$$\frac{\partial f_2}{\partial u} = \beta_2/(\beta_1\beta_2 - \beta_3^2)$$

$$\frac{\partial f_2}{\partial u} = (-\beta_2 - \beta_3)/(\beta_1\beta_2 - \beta_3^2)$$

Ma trận A, B, Q, R trong vị trí Top

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -33.47820 & 0 & 6.2103 & 0 \\ 0 & 0 & 0 & 1 \\ 21.4410 & 0 & -50.8766 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 100.4110 \\ 0 \\ -181.5722 \end{bmatrix}$$

$$\text{Ma trận } Q = \begin{bmatrix} 20 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

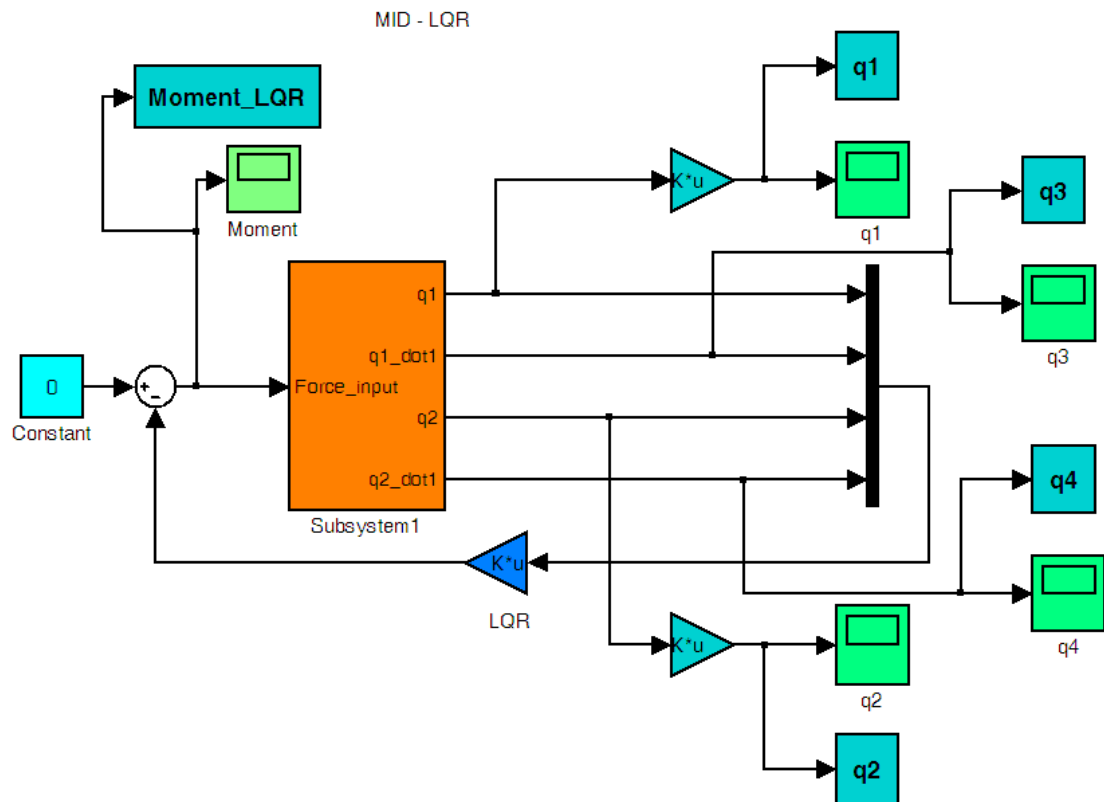
$$R = 1$$

Với  $K = \text{lqr}(A, B, Q, R)$

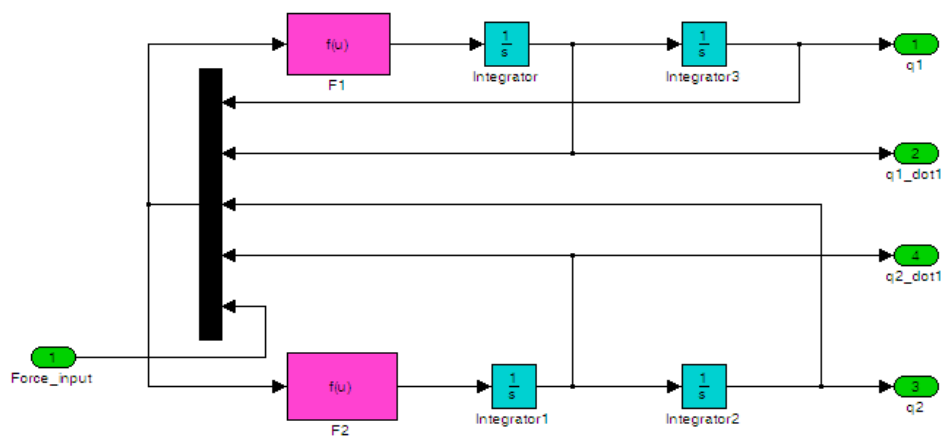
$$K = [-7.2723 \quad 2.9289 \quad -11.8899 \quad -1.2342]$$

## 4.2 Kết quả mô phỏng vị trí MID dùng LQR

Sơ đồ mô phỏng vị trí Mid



Hình 4.3: Sơ đồ khối mô phỏng vị trí Mid



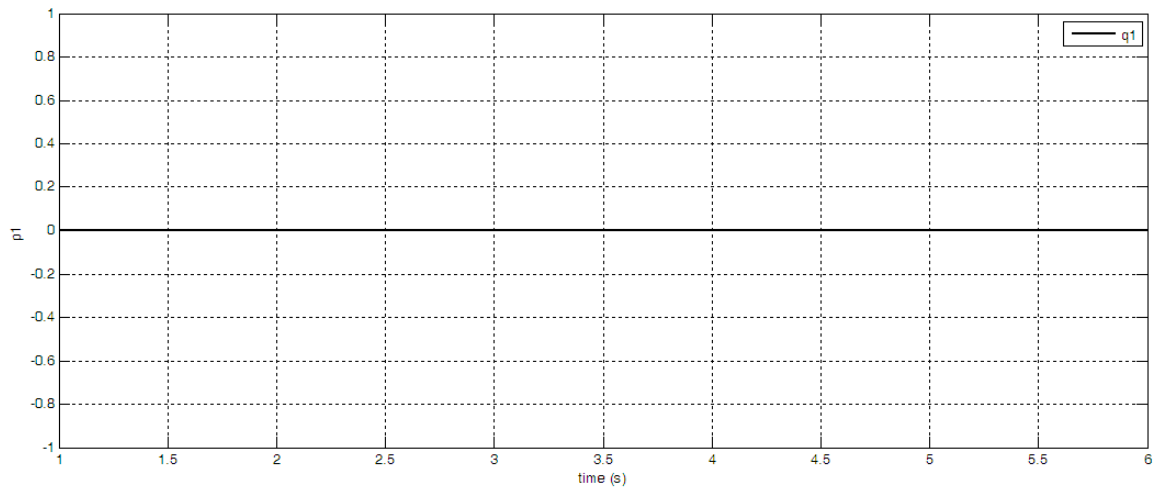
Hình 4.4: Sơ đồ subsystem1 vị trí Mid

Với f1 và f2 là 2 hàm

$$\ddot{q}_1 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} [\beta_2 \tau_m - \beta_2 \beta_3 \sin(q_2)(\dot{q}_2 + \dot{q}_1)^2 + \beta_3^2 (\sin q_2)(\cos q_2) \dot{q}_1^2 + \beta_2 \beta_4 \sin q_1 - \beta_3 \beta_5 \cos q_2 \sin(q_1 + q_2)]$$

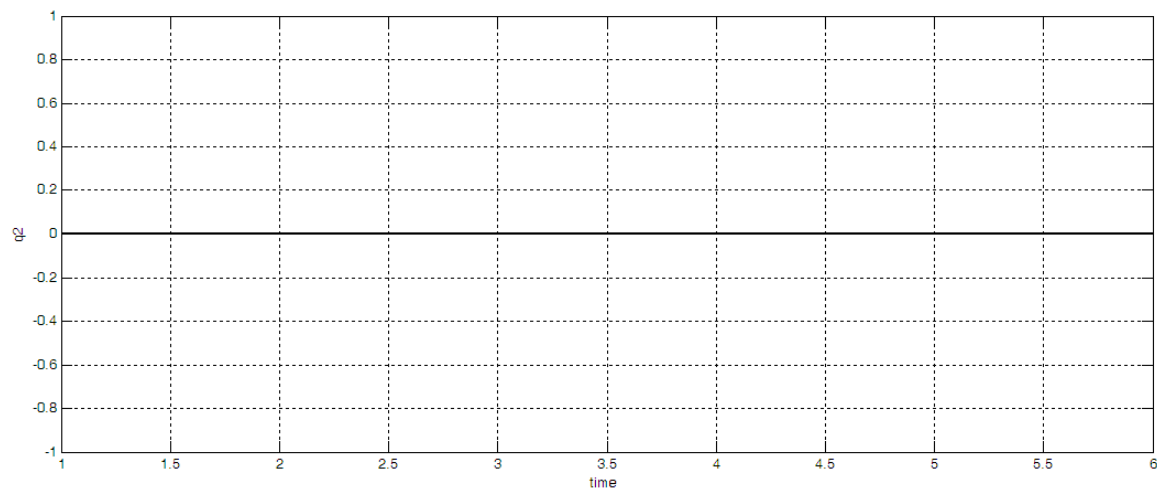
$$\ddot{q}_2 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} \{-\beta_2 \tau_m + \tau_m \beta_3 \cos(q_2) + \beta_3 \sin(q_2)[(\beta_2 - \beta_3 \cos q_2)(\dot{q}_2 + \dot{q}_1)^2 + (\beta_1 - \beta_3 \cos q_2) \dot{q}_1^2] - \beta_5 g \sin(q_1 + q_2)(\beta_1 - \beta_3 \cos q_2) - \beta_4 g \sin q_1 (\beta_2 - \beta_3 \cos q_2)\}$$

**q<sub>1</sub> (độ)**



Hình 4.5: Mô phỏng tín hiệu ra của q<sub>1</sub>

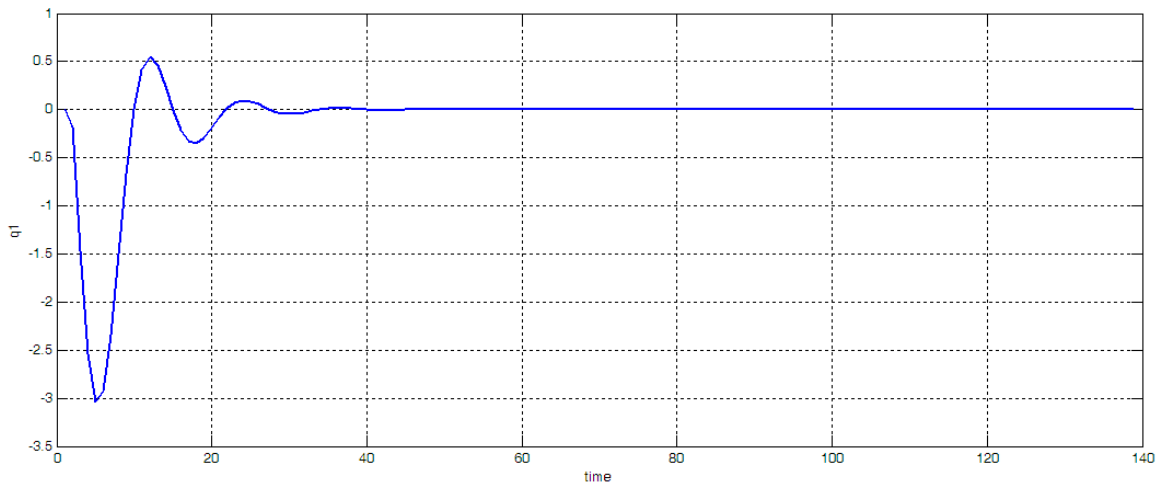
**q<sub>2</sub> (độ)**



Hình 4.6: Mô phỏng tín hiệu ra của q<sub>2</sub>

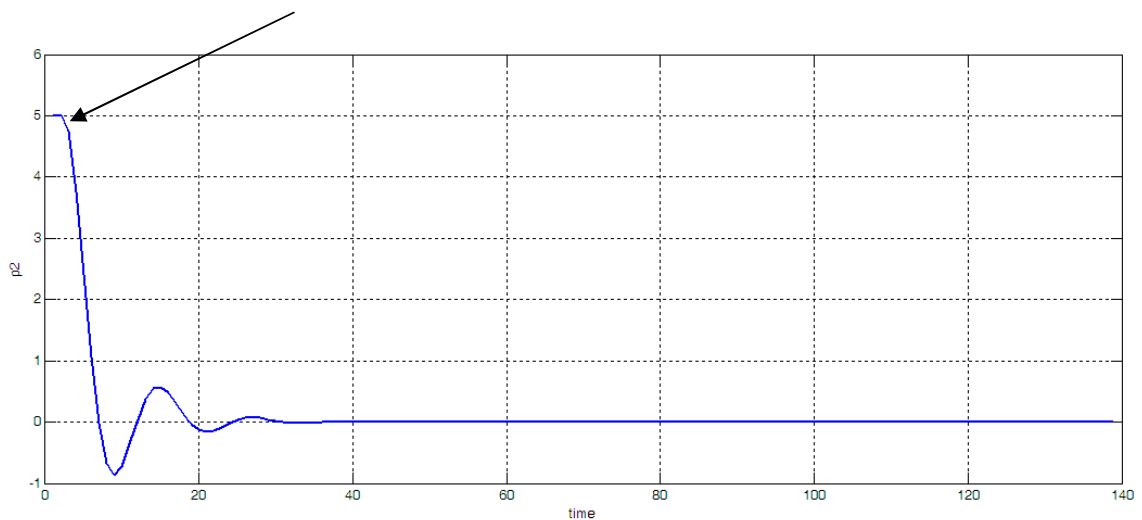
Nhận xét: Hệ thống ổn định khi không có tác động nhiễu và góc đặt ban đầu của  $q_2$  không lệch so với link2, hai link cân bằng tại vị trí Mid với hai góc đặt ban đầu là  $0^0$  hệ thống ổn định tại vị trí cân bằng  $(-\pi/2, 0, \pi, 0)$

**$q_1$  (độ) với góc đặt ban đầu của  $q_2$  là  $5^0$**



Hình 4.7: Tín hiệu ra của góc  $q_1$

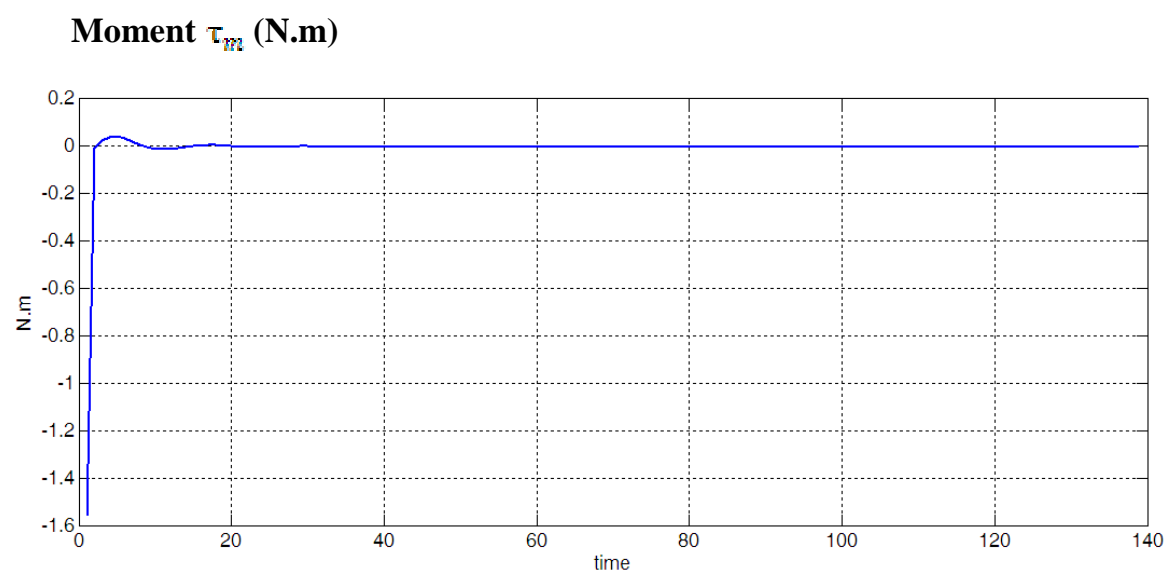
**$q_2$  (độ) với góc đặt ban đầu  $5^0$**



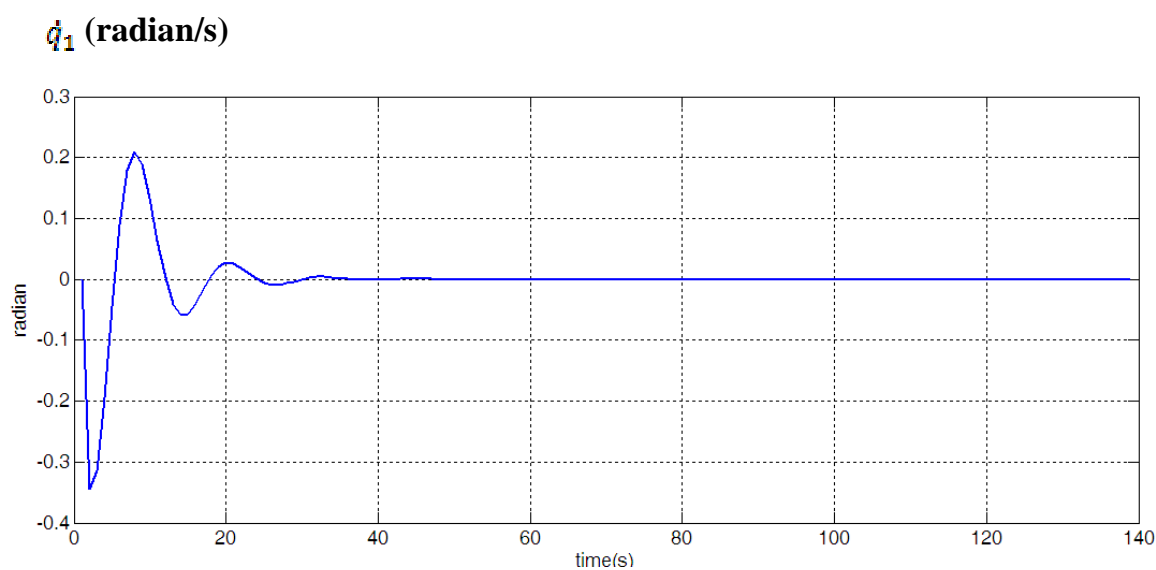
Hình 4.8: Tín hiệu ra của góc  $q_2$

Nhận xét: Khi đặt góc ban đầu là  $q_2 = 5^0$  lệch so với link2 ta thấy hệ thống link1 bắt đầu dao động nhưng chiều của link1 ngược so với link2 để giữ ổn định cho link1 không rơi khỏi vị trí cân bằng.



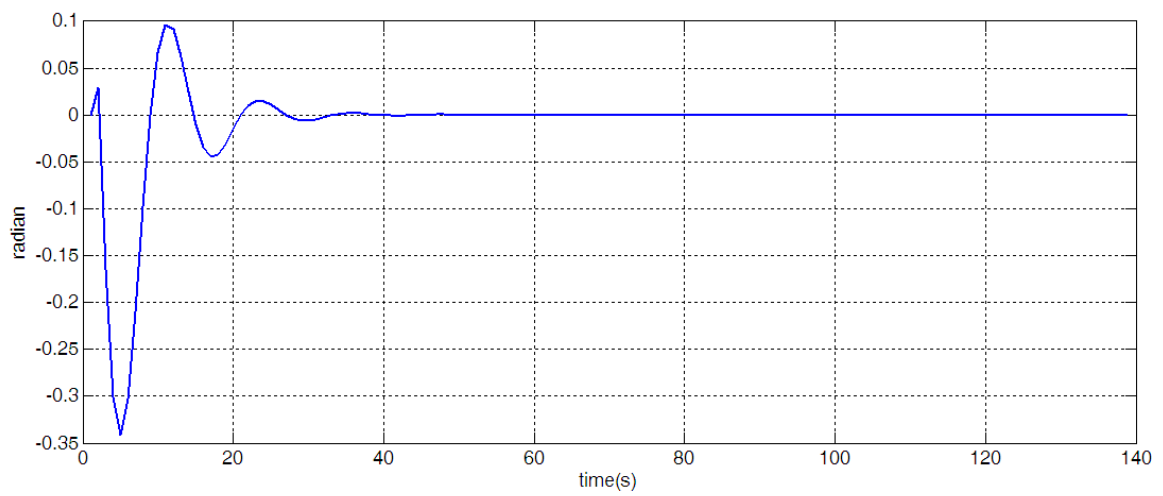


Hình 4.9 Tín hiệu ra moment  $\tau_m$



Hình 4.10: Tín hiệu ra  $\dot{q}_1$

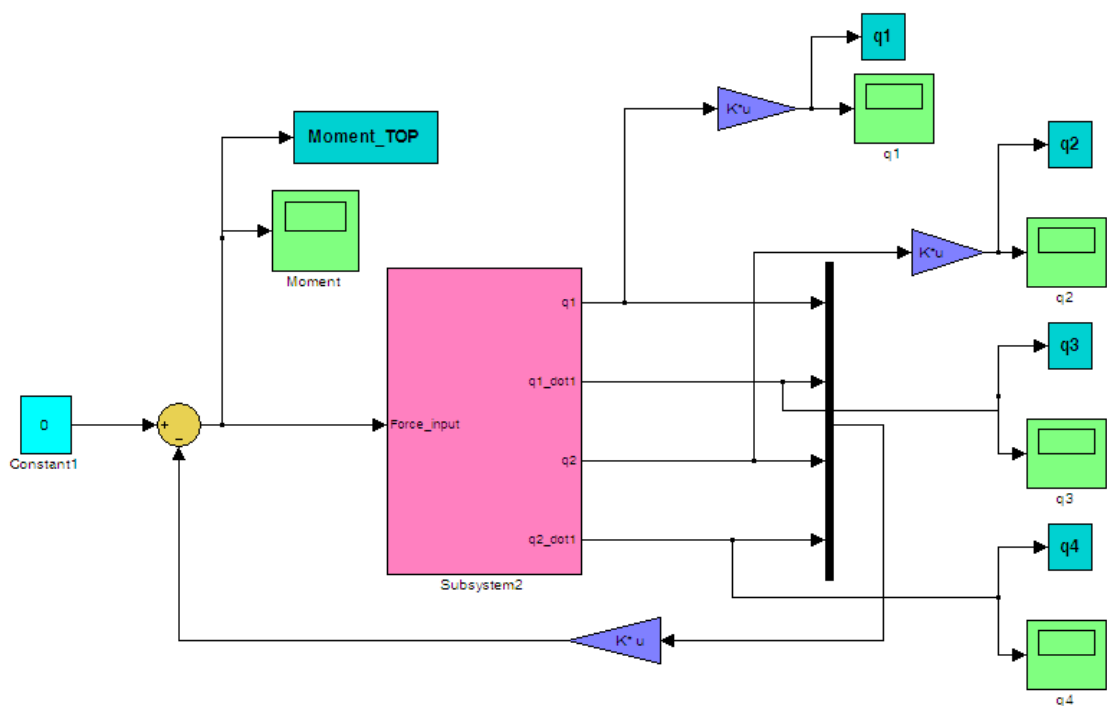
$\dot{q}_2$  (radian/s)



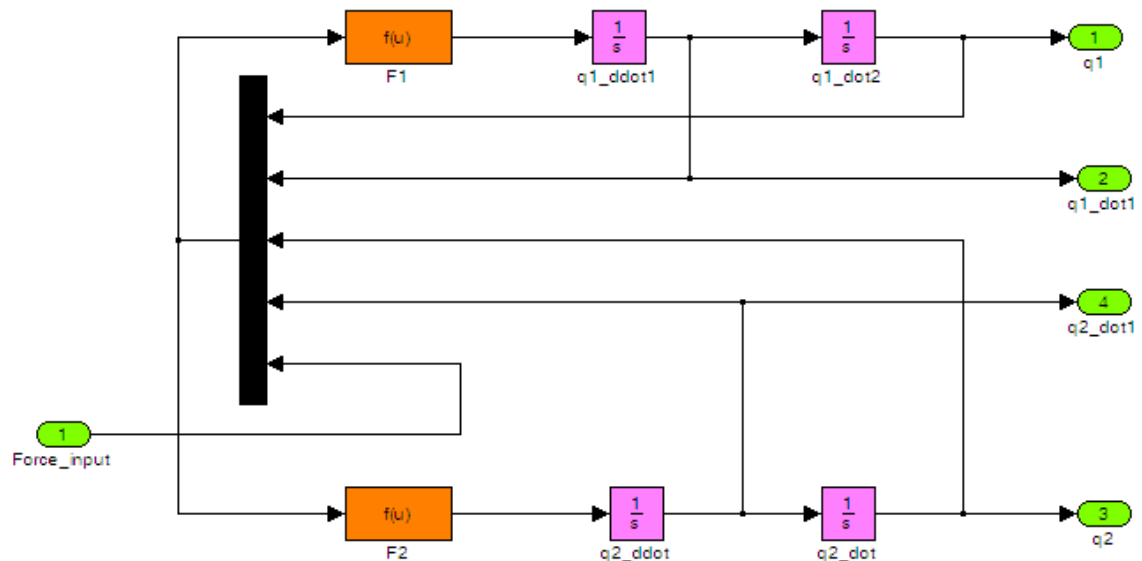
Hình 4.11: Tín hiệu ra của  $\dot{q}_2$

### 4.3 Kết quả mô phỏng vị trí TOP dùng LQR

Sơ đồ mô phỏng vị trí TOP dùng LQR



Hình 4.12: Sơ đồ khối mô phỏng vị trí Top



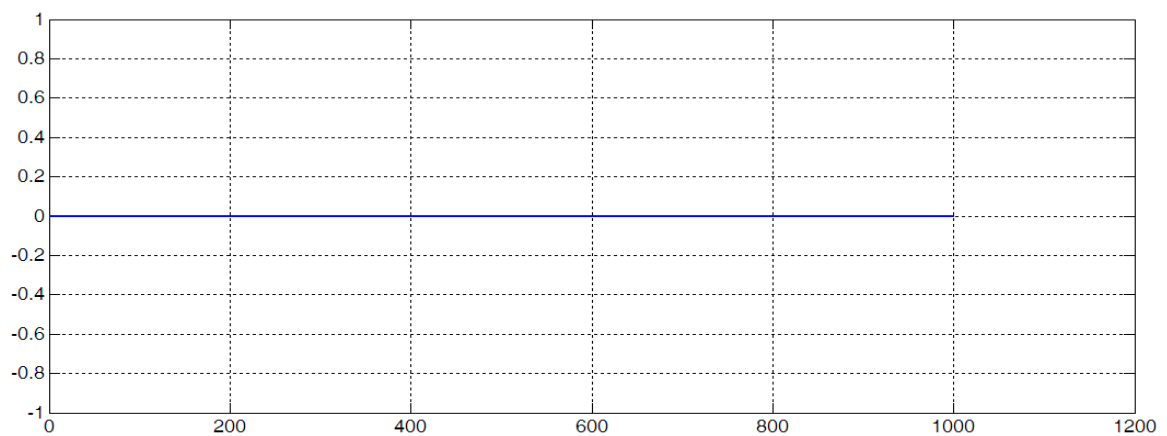
Hình 4.13: Sơ đồ subsystem2 vị trí Top

Với  $f_1$  và  $f_2$  là 2 hàm

$$\ddot{q}_1 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} [\beta_2 \tau_m + \beta_2 \beta_3 \sin(q_2) (\dot{q}_2 + \dot{q}_1)^2 + \beta_3^2 (\sin q_2) (\cos q_2) \dot{q}_1^2 - \beta_2 \beta_4 \sin q_1 + \beta_3 \beta_5 \cos q_2 \sin(q_1 + q_2)]$$

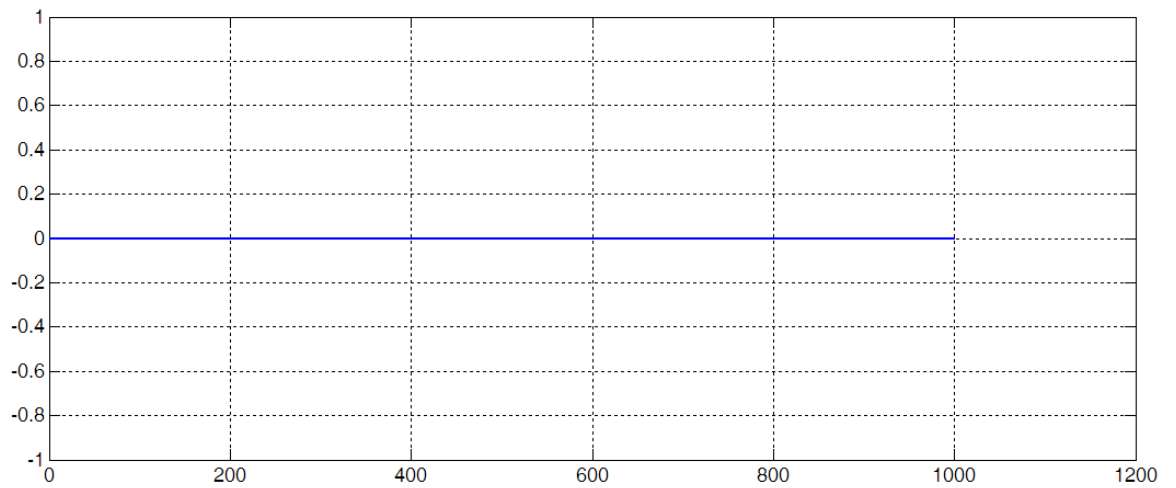
$$\ddot{q}_2 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} \{-\beta_2 \tau_m - \tau_m \beta_3 \cos(q_2) - \beta_3 \sin(q_2) [(\beta_2 + \beta_3 \cos q_2) (\dot{q}_2 + \dot{q}_1)^2 + (\beta_1 + \beta_3 \cos q_2) \dot{q}_1^2] - \beta_5 g \sin(q_1 + q_2) (\beta_1 + \beta_3 \cos q_2) + \beta_4 g \sin q_1 (\beta_2 + \beta_3 \cos q_2)\}$$

$q_1(\text{radian})$



Hình 4.14: Tín hiệu ra của  $q_1$

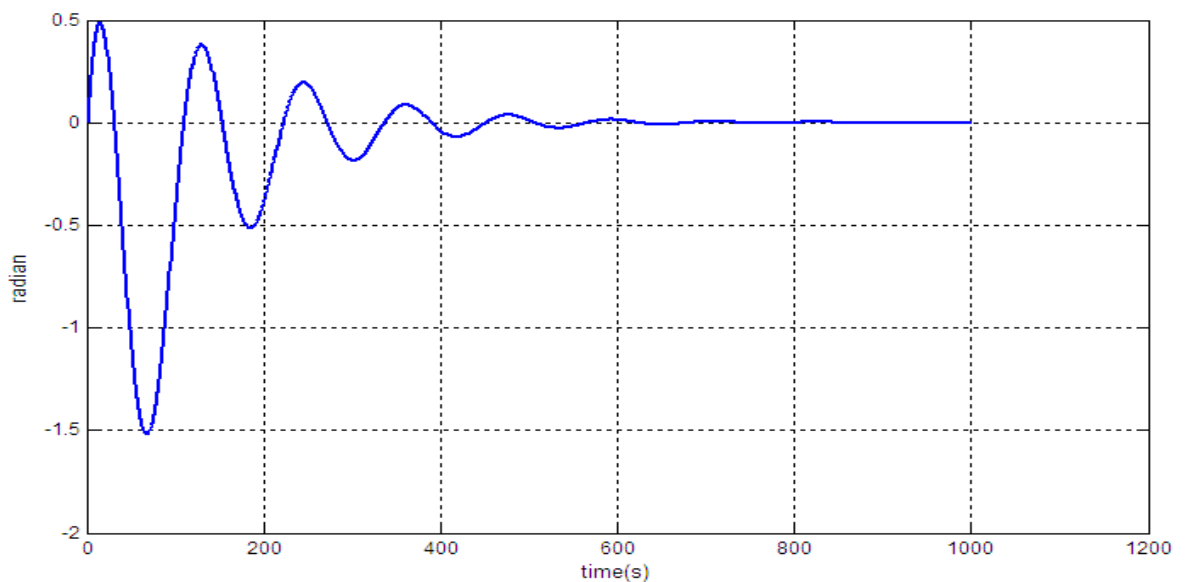
$q_2$  (radian)



Hình 4.15: Tín hiệu ra của  $q_2$

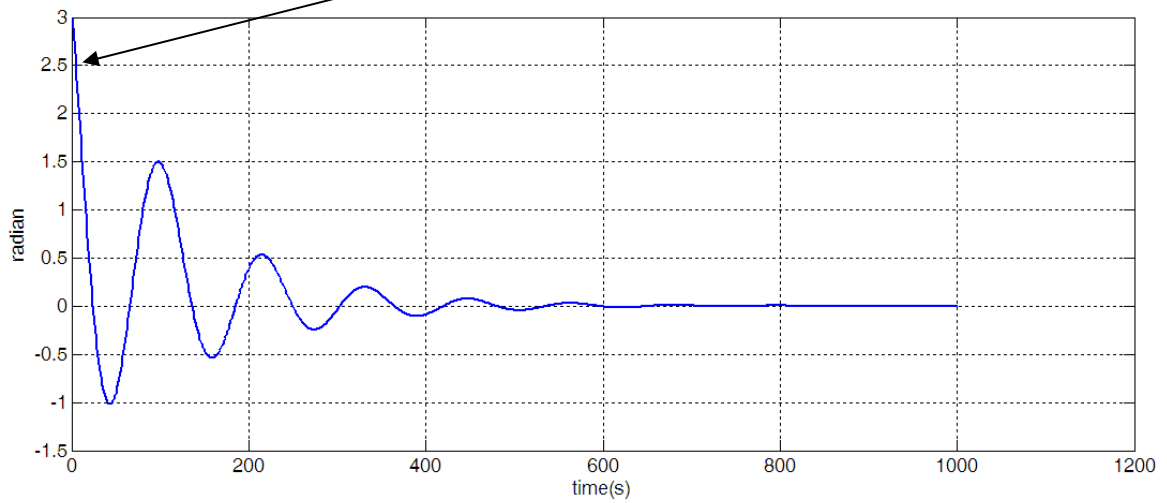
Nhận xét: Hệ thống ổn định khi không có tác động nhiễu và góc đặt ban đầu của  $q_2$  không lệch so với link2, hai link cân bằng tại vị trí Mid với hai góc đặt ban đầu là  $0^0$  hệ thống ổn định tại vị trí cân bằng ( $\pi/2, 0, 0, 0$ )

$q_1$  (radian) với góc đặt ban đầu  $q_2$  là  $3^0$



Hình 4.16: Tín hiệu ra của  $q_1$

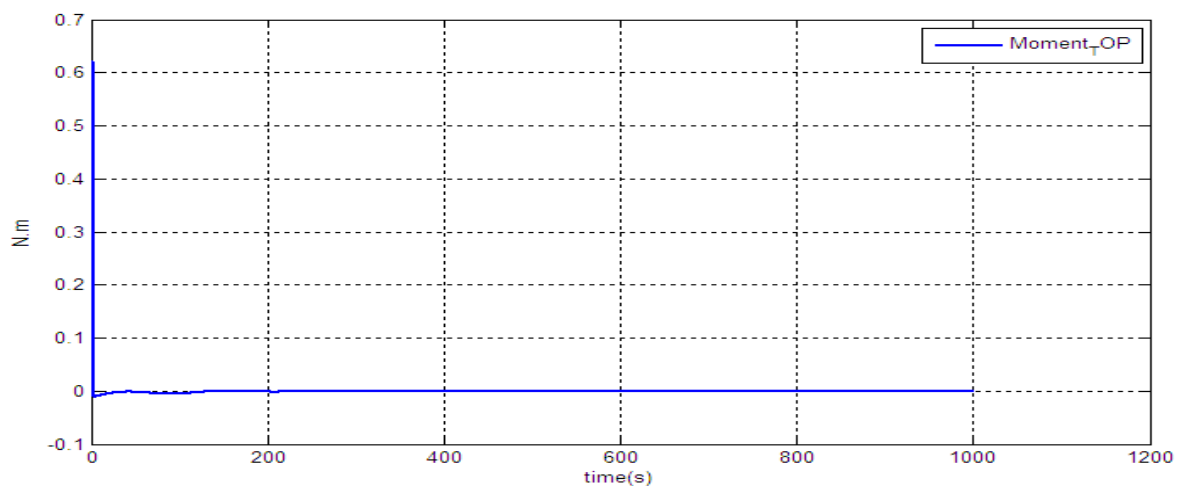
**Đặt link2 lệch một góc  $q_2 = 3^\circ$  so với link1 kết quả mô phỏng**



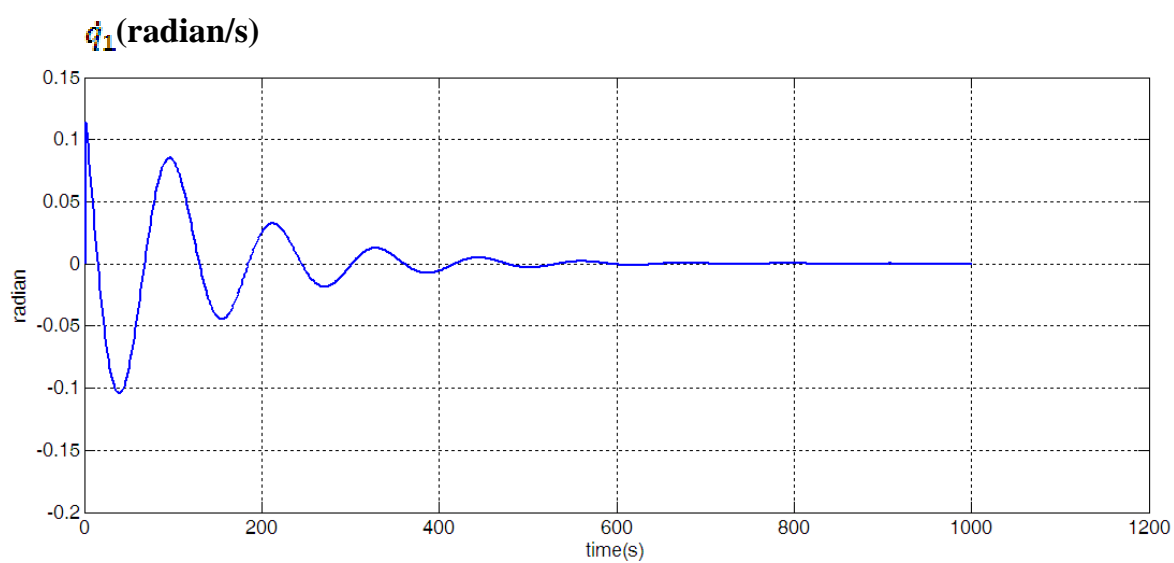
Hình 4.17: Tín hiệu ra của  $q_2$

Nhận xét: Khi đặt góc ban đầu là  $q_2 = 3^\circ$  lệch so với link2 ta thấy hệ thống link1 bắt đầu dao động nhưng chiều của link1 ngược so với link2 để giữ ổn định cho link1 không rơi khỏi vị trí cân bằng. Nhưng hệ thống dao động mạnh hơn vị trí cân bằng Mid, và góc link1 lệch  $3^\circ$  so với link2 nên vị trí cân bằng vị trí Top khó cân bằng hơn vị trí Mid.

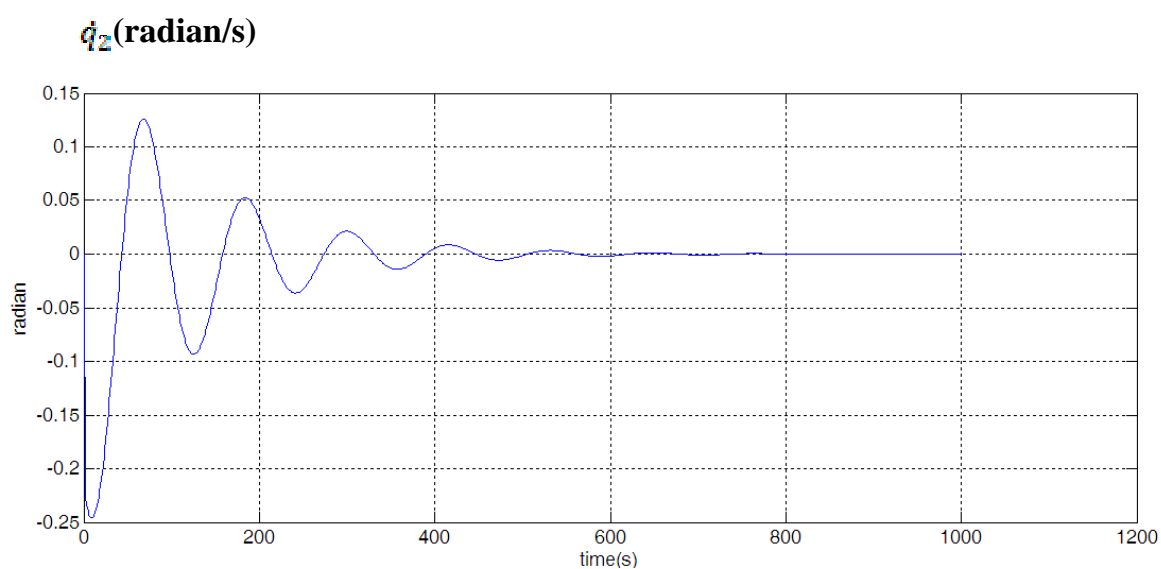
**Moment  $\tau_m$  (N.m)**



Hình 4.18: Tín hiệu ra moment  $\tau_m$



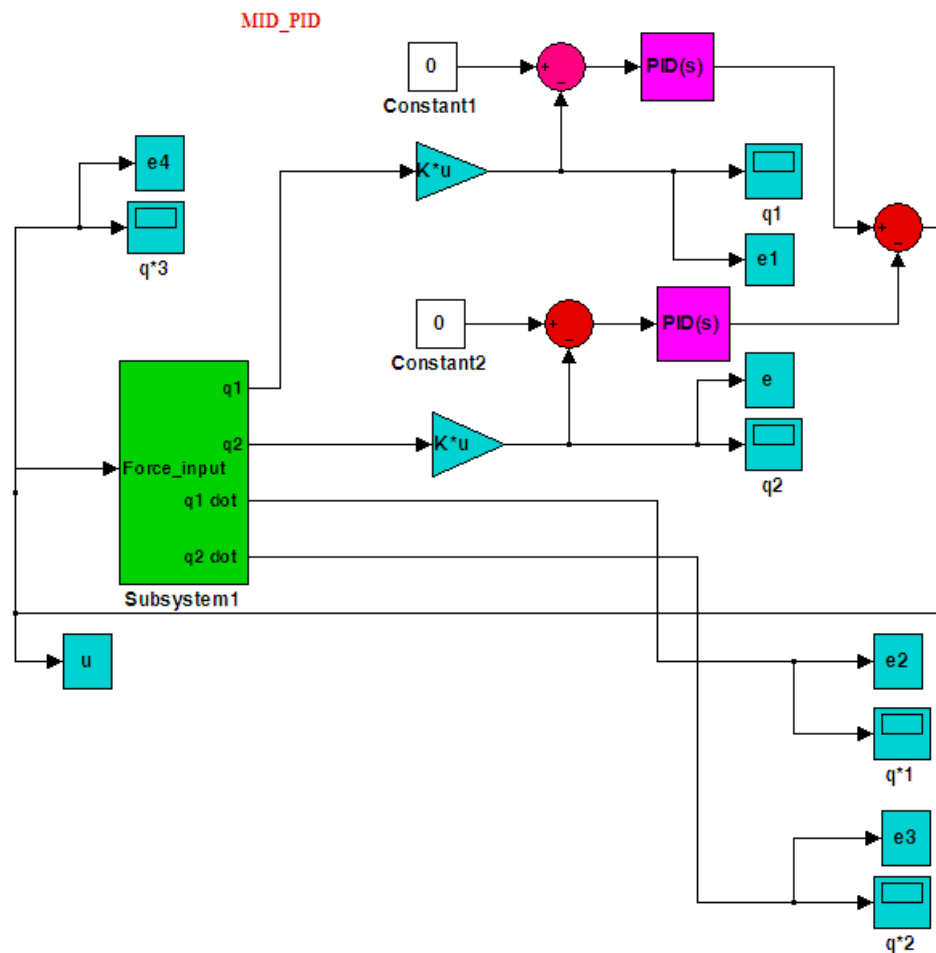
Hình 4.19: Tín hiệu ra của  $q_1$



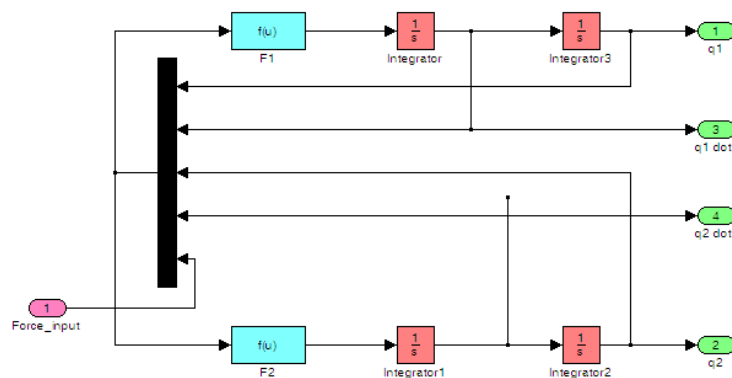
Hình 4.20: Tín hiệu ra của  $q_2$

#### 4.4 Kết quả mô phỏng vị trí MID dùng PID

Sơ đồ mô phỏng vị trí Mid PID



Hình 4.21: Sơ đồ khối mô phỏng vị trí Mid-PID



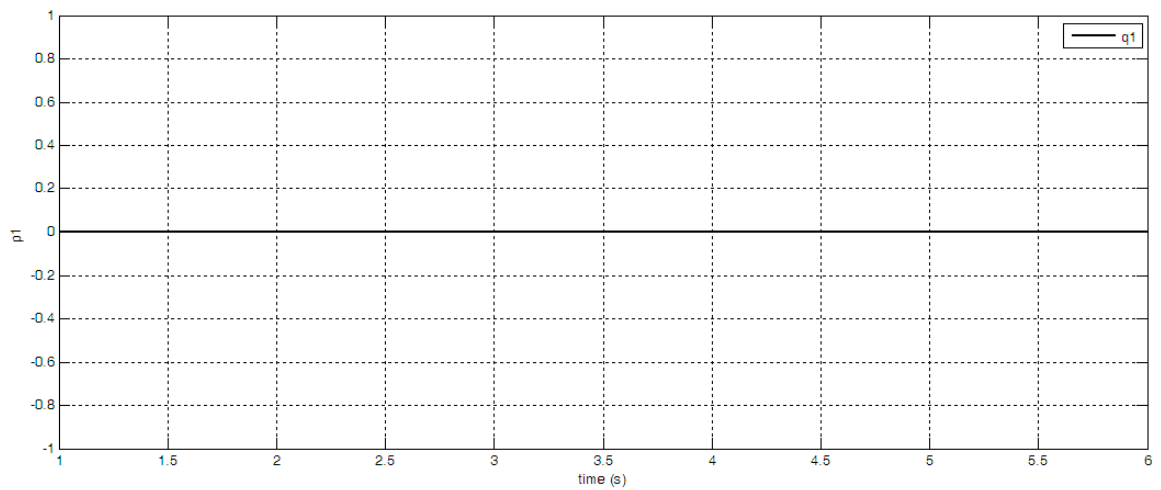
Hình 4.22: Sơ đồ subsystem1 vị trí Mid-PID

Với f1 và f2 là 2 hàm

$$\ddot{q}_1 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} [\beta_2 \tau_m - \beta_2 \beta_3 \sin(q_2)(\dot{q}_2 + \dot{q}_1)^2 + \beta_3^2 (\sin q_2)(\cos q_2) \dot{q}_1^2 + \beta_2 \beta_4 \sin q_1 - \beta_3 \beta_5 \cos q_2 \sin(q_1 + q_2)]$$

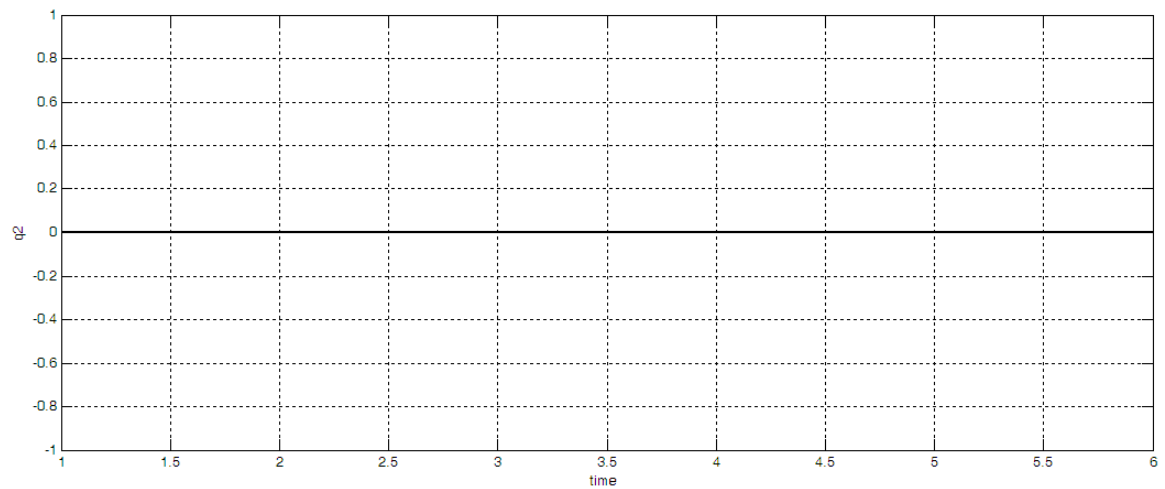
$$\ddot{q}_2 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} \{-\beta_2 \tau_m + \tau_m \beta_3 \cos(q_2) + \beta_3 \sin(q_2)[(\beta_2 - \beta_3 \cos q_2)(\dot{q}_2 + \dot{q}_1)^2 + (\beta_1 - \beta_3 \cos q_2) \dot{q}_1^2] - \beta_5 g \sin(q_1 + q_2)(\beta_1 - \beta_3 \cos q_2) - \beta_4 g \sin q_1 (\beta_2 - \beta_3 \cos q_2)\}$$

**q<sub>1</sub> (độ)**



Hình 4.23: Mô phỏng tín hiệu ra của q<sub>1</sub>

**q<sub>2</sub> (độ)**

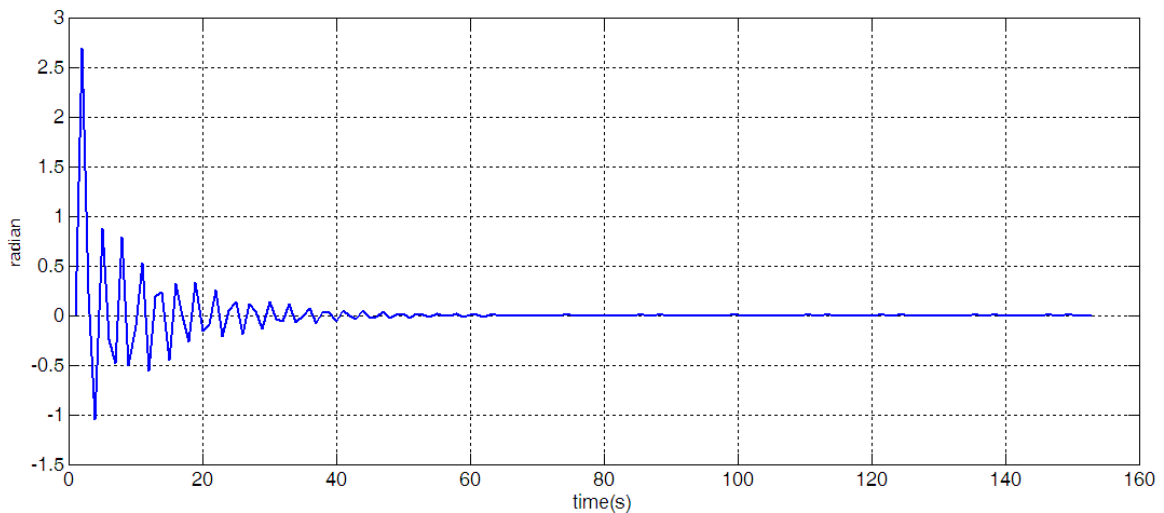


Hình 4.24: Mô phỏng tín hiệu ra của q<sub>2</sub>



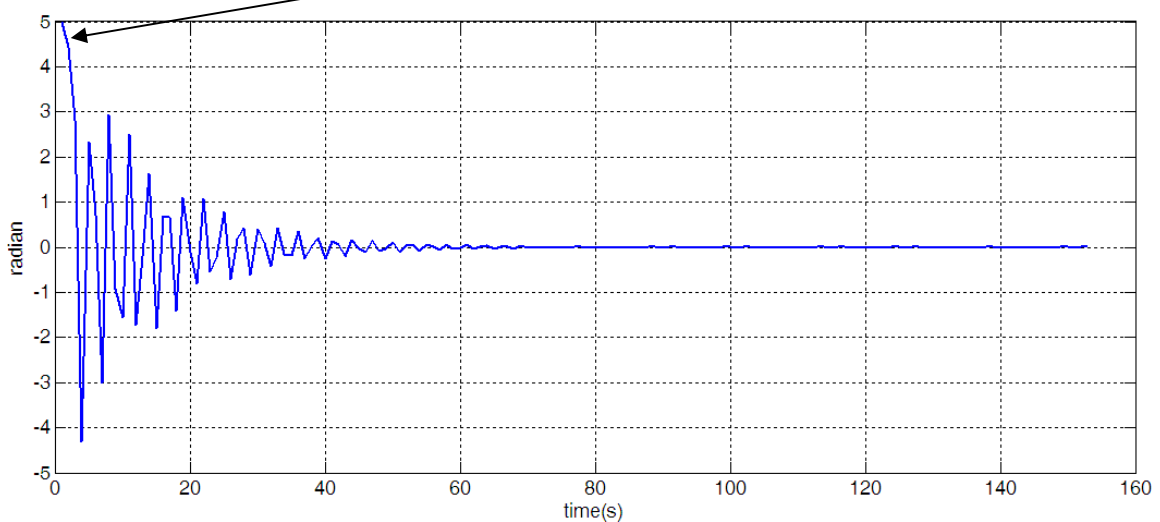
Nhận xét: Hệ thống ổn định khi không có tác động nhiễu và góc đặt ban đầu của  $q_2$  không lệch so với link2, hai link cân bằng tại vị trí Mid với hai góc đặt ban đầu là  $0^0$  hệ thống ổn định tại vị trí cân bằng  $(-\pi/2, 0, \pi, 0)$

**$q_1$  (độ) với góc đặt ban đầu của  $q_2$  là  $5^0$**



Hình 4.25: Tín hiệu ra của góc  $q_1$

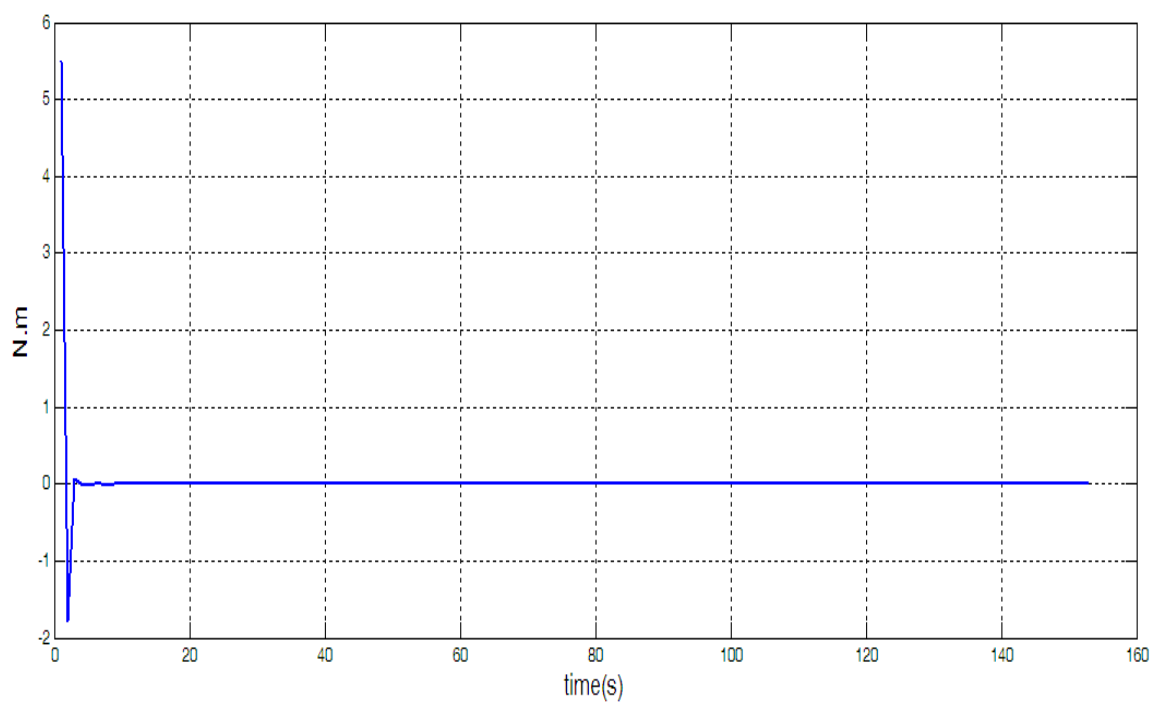
**$q_2$  (độ) với góc đặt ban đầu  $5^0$**



Hình 4.26: Tín hiệu ra của góc  $q_2$

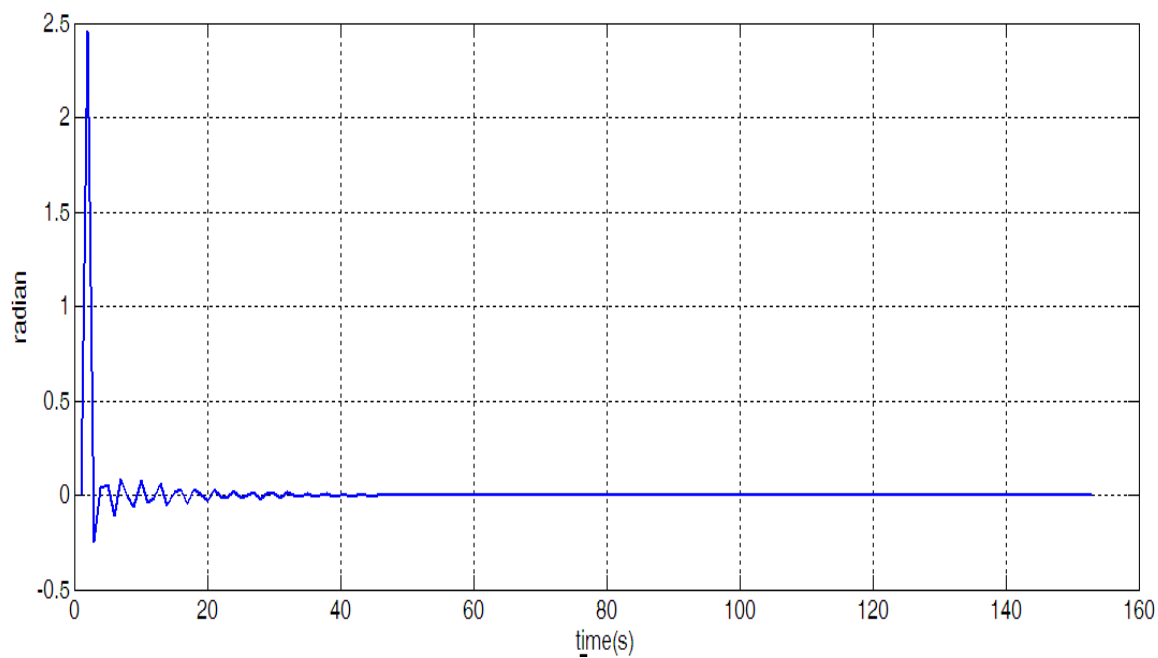
Nhận xét: Khi đặt góc ban đầu là  $q_2 = 5^0$  lệch so với link2 ta thấy hệ thống link1 bắt đầu dao động nhưng chiều của link1 ngược so với link2 để giữ ổn định cho link1 không rơi khỏi vị trí cân bằng.

**Moment  $\tau_m$  (N.m)**



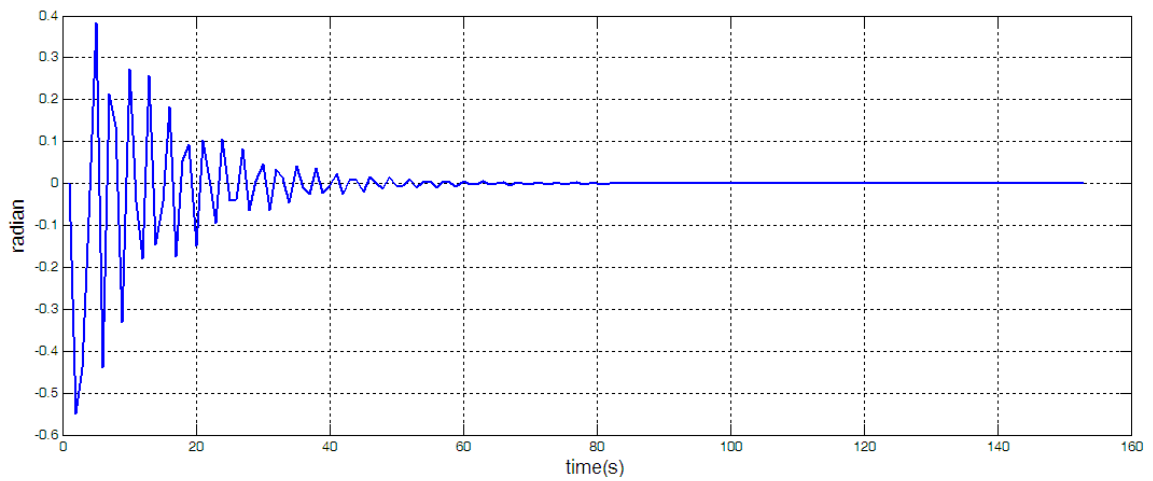
Hình 4.27 Tín hiệu ra môment  $\tau_m$

**$\dot{q}_1$  (radian/s)**



Hình 4.28: Tín hiệu ra  $\dot{q}_1$

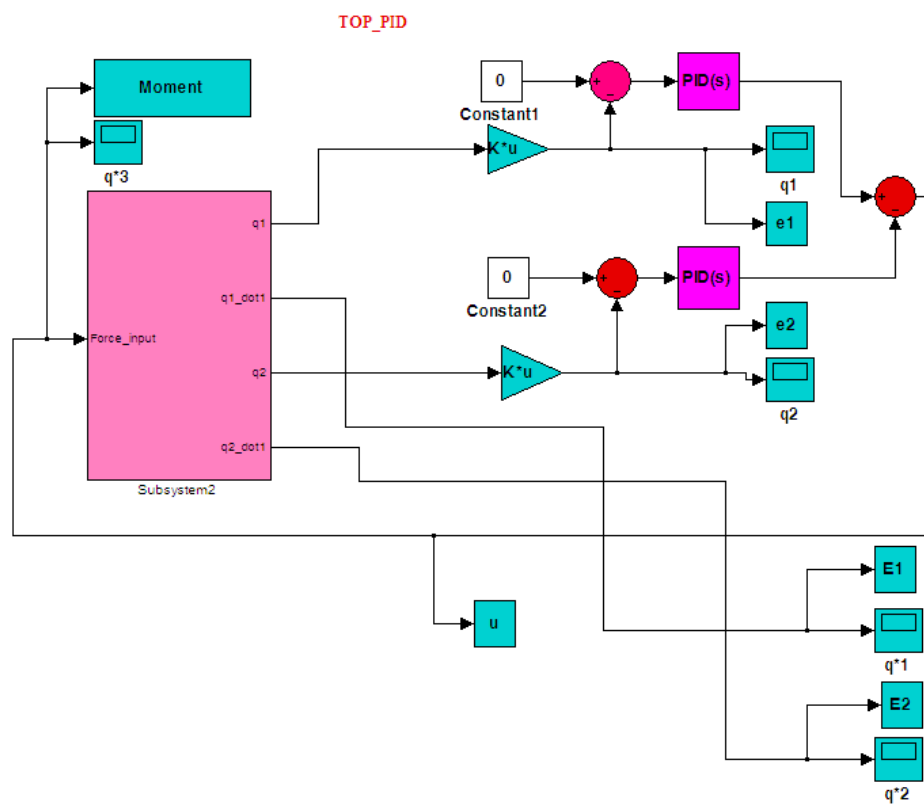
$\dot{q}_2$  (radian/s)



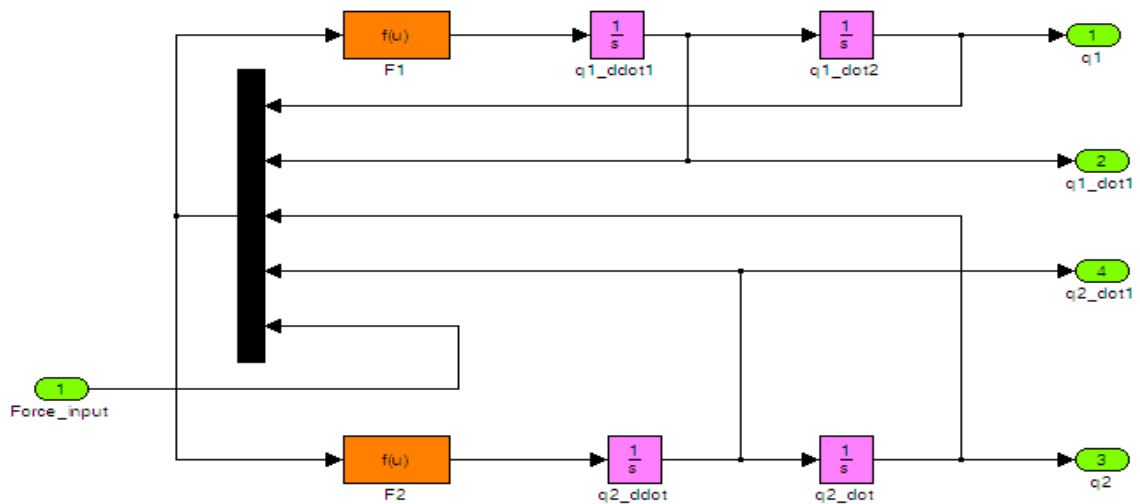
Hình 4.29 Tín hiệu ra của  $\dot{q}_2$

#### 4.4 Kết quả mô phỏng vị trí TOP dùng PID

Sơ đồ mô phỏng vị trí TOP dùng PID



Hình 4.30: Sơ đồ khối mô phỏng vị trí Top



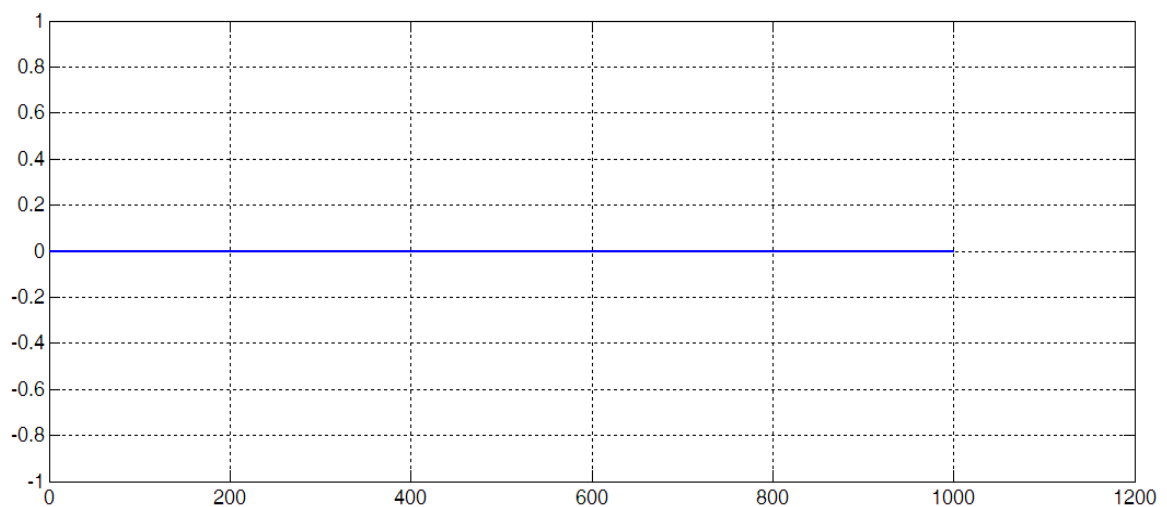
Hình 4.31 Sơ đồ subsystem2 vị trí Top PID

Với f1 và f2 là 2 hàm

$$\ddot{q}_1 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} [\beta_2 \tau_m + \beta_2 \beta_3 \sin(q_2) (\dot{q}_2 + \dot{q}_1)^2 + \beta_3^2 (\sin q_2) (\cos q_2) \dot{q}_1^2 - \beta_2 \beta_4 \sin q_1 + \beta_3 \beta_5 \cos q_2 \sin(q_1 + q_2)]$$

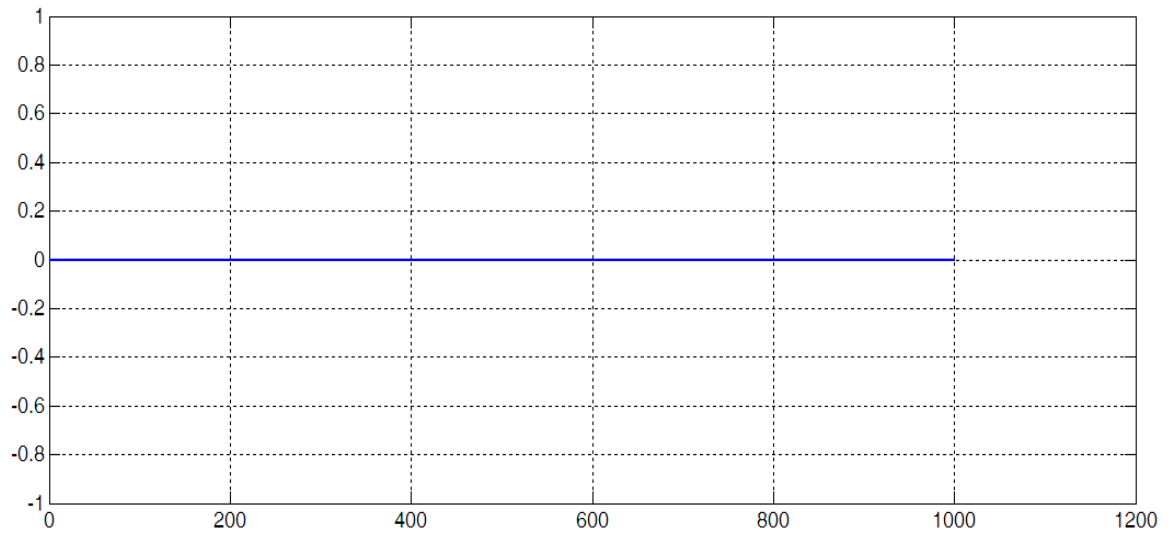
$$\ddot{q}_2 = \frac{1}{\beta_1\beta_2 - \beta_3^2 \cos^2 q_2} \{-\beta_2 \tau_m - \tau_m \beta_3 \cos(q_2) - \beta_3 \sin(q_2) [(\beta_2 + \beta_3 \cos q_2) (\dot{q}_2 + \dot{q}_1)^2 + (\beta_1 + \beta_3 \cos q_2) \dot{q}_1^2] - \beta_5 g \sin(q_1 + q_2) (\beta_1 + \beta_3 \cos q_2) + \beta_4 g \sin q_1 (\beta_2 + \beta_3 \cos q_2)\}$$

$q_1$ (radian)



Hình 4.32 Tín hiệu ra của  $q_1$

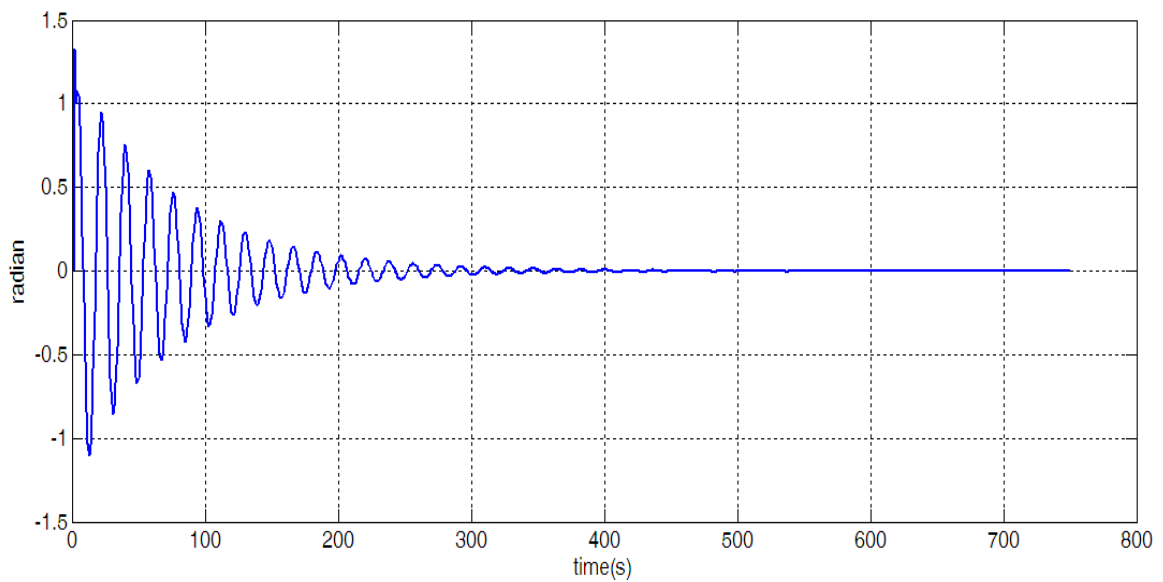
**$q_2$  (radian)**



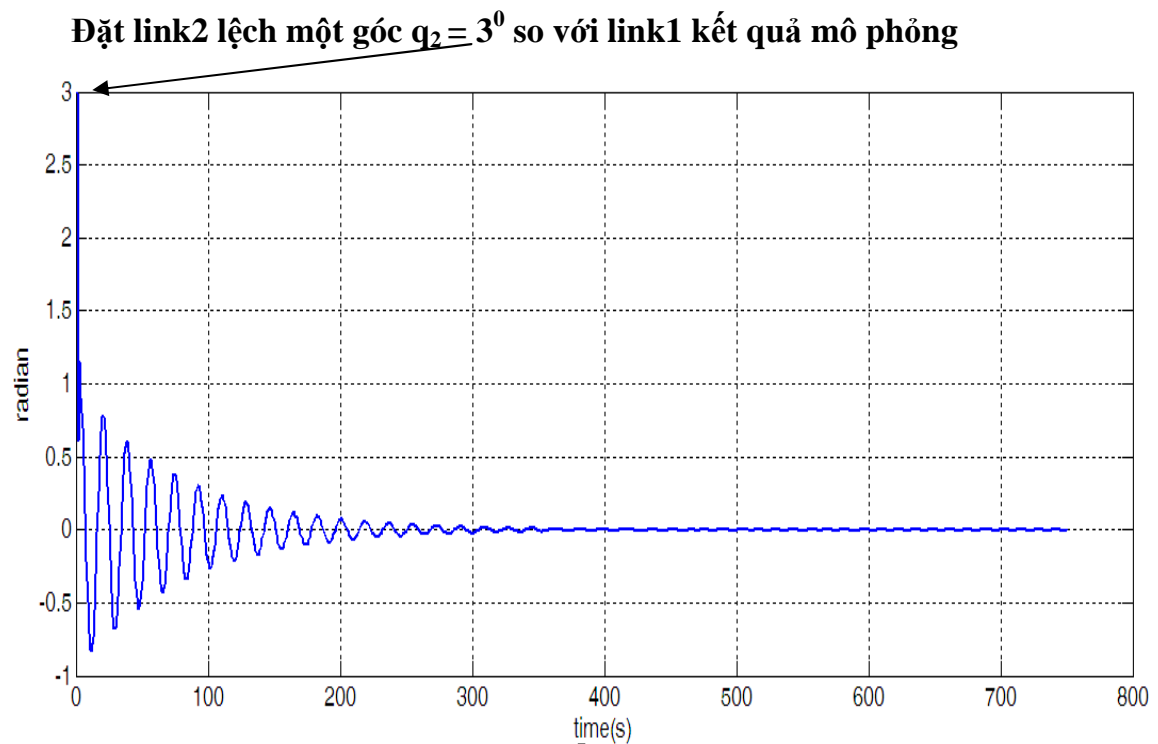
Hình 4.33 Tín hiệu ra của  $q_2$

Nhận xét: Hệ thống ổn định khi không có tác động nhiễu và góc đặt ban đầu của  $q_2$  không lệch so với link2, hai link cân bằng tại vị trí Mid với hai góc đặt ban đầu là  $0^\circ$  hệ thống ổn định tại vị trí cân bằng ( $\pi/2, 0, 0, 0$ )

**$q_1$  (radian) với góc đặt ban đầu  $q_2$  là  $3^\circ$**



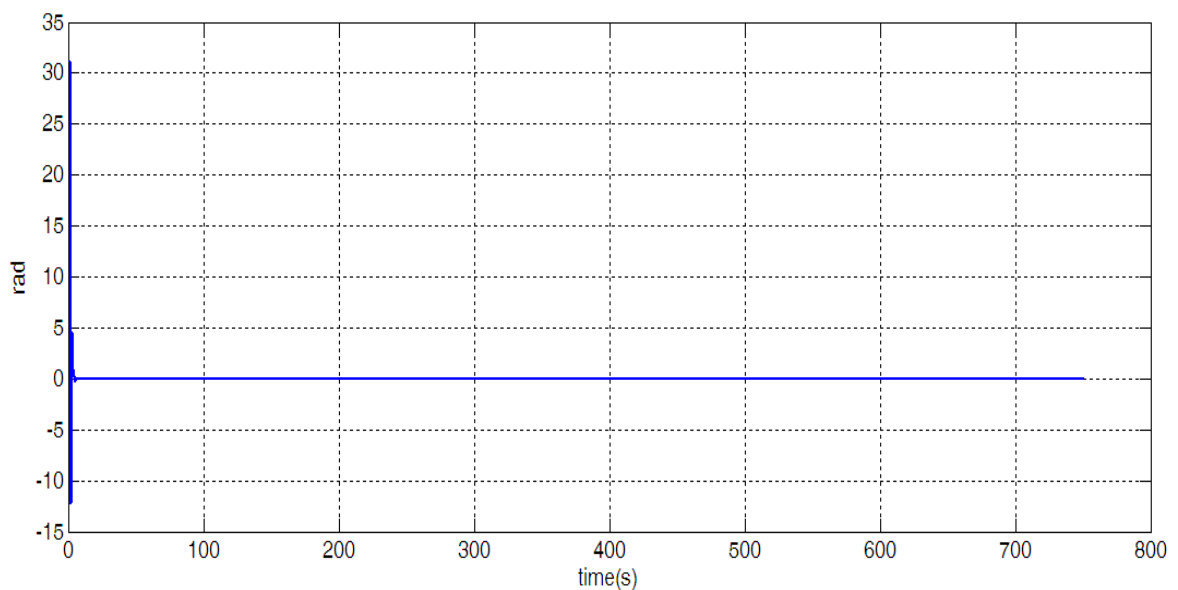
Hình 4.34 Tín hiệu ra của  $q_1$



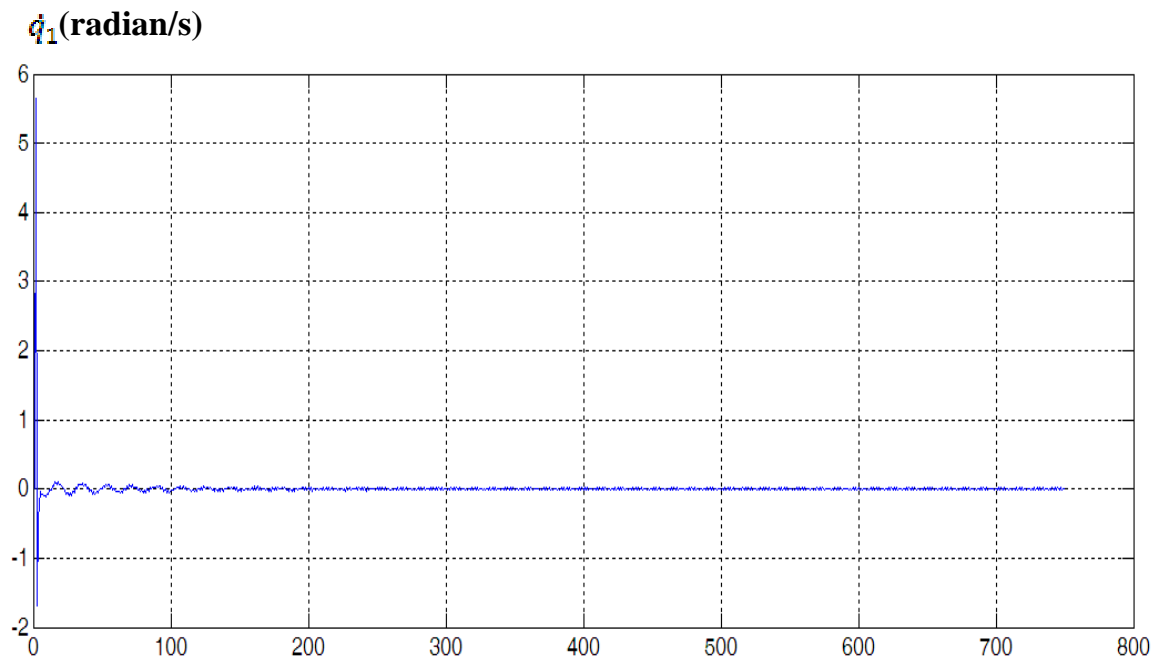
Hình 4.35 Tín hiệu ra của  $q_2$

Nhận xét: Khi đặt góc ban đầu là  $q_2 = 3^0$  lệch so với link2 ta thấy hệ thống link1 bắt đầu dao động nhưng chiều của link1 ngược so với link2 để giữ ổn định cho link1 không rơi khỏi vị trí cân bằng. Nhưng hệ thống dao động mạnh hơn vị trí cân bằng Mid, và góc link1 lệch  $3^0$  so với link2 nên vị trí cân bằng vị trí Top khó cân bằng hơn vị trí Mid.

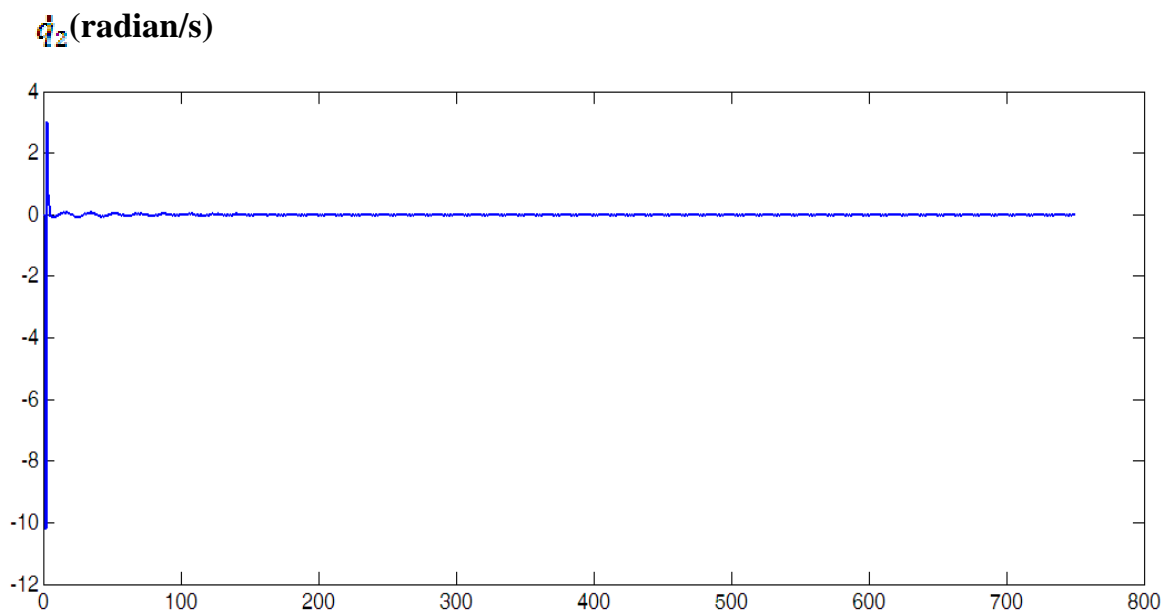
**Moment  $\tau_m$  (N.m)**



Hình 4.36 Tín hiệu ra moment  $\tau_m$



Hình 4.37 Tín hiệu ra của  $\dot{q}_1$



Hình 4.38 Tín hiệu ra của  $\dot{q}_2$

Nhận xét: Khi đặt góc ban đầu là  $q_2 = 3^\circ$  lệch so với link2 ta thấy hệ thống link1 bắt đầu dao động nhưng chiều của link1 ngược so với link2 để giữ ổn định cho link1 không rơi khỏi vị trí cân bằng.

#### 4.5 Kết luận

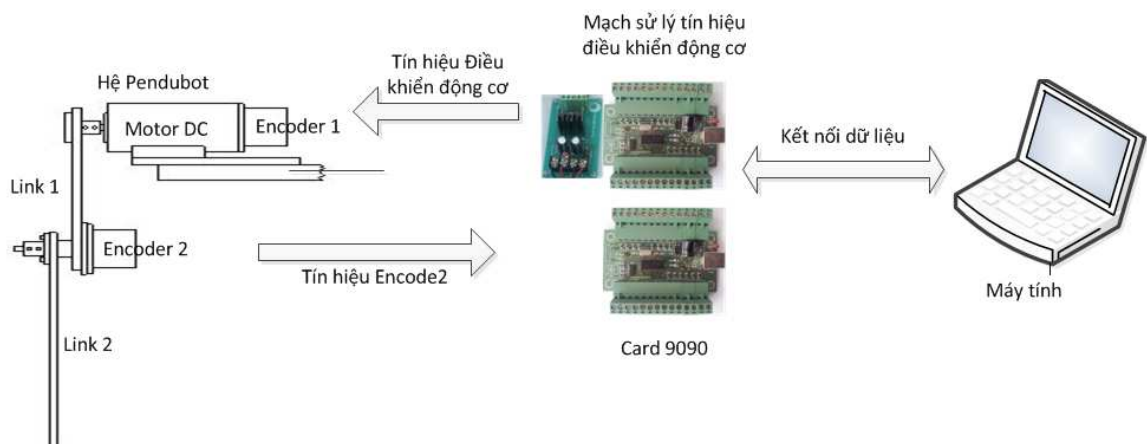
Giải thuật LQR áp dụng được với mọi đối tượng có số ngõ vào bằng với số ngõ ra (multi input –multi output hoặc single input-single output). Tuy nhiên đối với hệ có số ngõ vào ít hơn số ngõ ra như hệ Pendubot thì phải dùng PID thỏa hiệp, do đó hệ thống chắc chắn phải dao động 1 thời gian mới ổn định được. Điều này là chắc chắn không tránh khỏi. Đây là bất lợi của việc dùng PID cho hệ có số ngõ vào ít hơn số ngõ ra (1 động cơ điều khiển 2 góc lệch cánh tay pendubot và góc pendubot).

Bộ PID thỏa hiệp: Do một bộ PID chỉ nhận được một tín hiệu vào và một tín hiệu ra, do đó đối với hệ Pendubot có số ngõ vào ít hơn số ngõ ra nên tác giả phải dùng hai bộ PID lấy tín hiệu cho  $q_1$  và  $q_2$ . để đưa tín hiệu hồi tiếp về điều khiển bộ điều khiển hệ Pendubot.



## CHƯƠNG 5: THIẾT KẾ PHẦN CỨNG HỆ PENDUBOT

Chương này học viên trình bày về thiết kế phần cứng của hệ Pendubot, giới thiệu sơ lược về card 9090 để thu nhập dữ liệu điều khiển từ hệ thống thực vào máy tính, điều khiển giữ cho con lắc ổn định bằng bộ điều khiển PID. Hình 5.1 thể hiện sơ đồ hệ thống Pendubot, hệ thống thực nghiệm được giới thiệu trên hình 5.2.



Hình 5.1 Sơ đồ khối hệ thống điều khiển hệ Pendubot



Hình 5.2 Hệ thống thực nghiệm

### 5.1 Giới thiệu phần cứng hệ thống Pendubot:

Hệ thống Pendubot được thiết kế gồm 3 phần chính:

Phần cơ khí.

Phần mạch điện .

Phần thu nhập dữ liệu giữa máy tính và hệ thống con lắc thông qua Card 9090

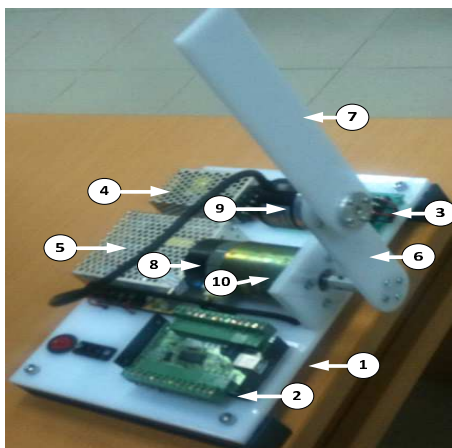
### 5.1.1 Phần cơ khí

Hệ Pendubot trên hình cấu tạo thiết kế khá đơn giản bao gồm:

Để làm bằng mica, trên mặt để thiết kế phần đỡ bộ Pendubot và hai Card giao tiếp 9090 được gắn xếp chồng lên nhau, mạch công suất điều khiển motor DC, bộ nguồn chuyển đổi 220VAC-12VDC dùng cho encoder 2. bộ nguồn chuyển đổi 220VAC-24VDC dùng cho động cơ DC

Đầu trục rotor động cơ DC còn lại được gắn Link1, và Link2 được kết nối với Link1 qua khớp nối gắn chung với encoder2 2000 xung để phản hồi vị trí Link2 có chức năng dùng để cân bằng ổn định .

Trục của Encoder 2 được gắn giữa khớp nối Link1 và Link2. Nhiệm vụ của hai encoder được sử dụng trong ứng dụng này là để xác định góc quay  $\theta_1$  ,  $\theta_2$  tương ứng với Link1 và Link2, sao cho Link1 ở vị trí bất kỳ vẫn giữ cho Link2 thẳng đứng khi tác động một lực bất kỳ lên Link 2



1. Bộ chân đế Pendubot
2. 2 Card 9090
3. Mạch công suất điều khiển động cơ DC
4. Bộ nguồn chuyển 220 VAC-12VDC dùng cho encoder 2
5. Bộ nguồn chuyển 220 VAC-12VDC dùng cho DC
6. Link 1
7. Link 2
8. Encoder 2 phản hồi vị trí link 2 12V 2000 xung
9. Encoder 1 phản hồi vị trí link 2 5V (card) 500 xung
10. Động cơ DC

Hình 5.3. Phần cơ khí mô hình hệ Pendubot

### 5.1.2 Phần mạch điện

Phần mạch điện của Hệ Pendubot do học viên thiết kế dựa trên sơ đồ mạch theo hình như sau:

#### Mạch động lực điều khiển động cơ một chiều:



Hình 5.4 Mạch động lực 3A-DCS HIPO

#### Đặc điểm:

Motor Driver 3A-DCS sử dụng chip LMD18200, là cầu H chịu đường dòng liên tục 3A (H-Bridge), và dòng tối đa chịu được 6A (peak current). Chip này được sản xuất dựa trên việc kết hợp nhiều công nghệ (bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure). Đặc biệt Motor Driver 3A-DCS có phản hồi dòng để dùng các ứng dụng điều khiển lực. Với kích thước tản nhiệt lớn đảm bảo việc đảo chiều motor diễn ra thường xuyên hoặc dòng điều khiển cao hơn 2.5A trong thời gian dài mà mạch vẫn hoạt động ổn định.

#### Cảm biến Encoder



Hình 5.5 Cảm biến Encoder của SHARP.

### Mô tả:

Cảm biến Encoder SHARP dùng để đo vị trí của động cơ (trong kỹ thuật robot có thể dùng để đo quãng đường đi của robot). Encoder SHARP có những đặc điểm sau:

- Tốc độ 2000 xung/phút;
- Đường kính trục 6mm, đường kính vỏ ngoài 45mm;
- Nguồn 12 VDC;
- Khối lượng 30g;
- 2 pha tín hiệu (Chanel A + Chanel B) có thể đếm xung lên, xuống;
- Độ bền cơ học cao.

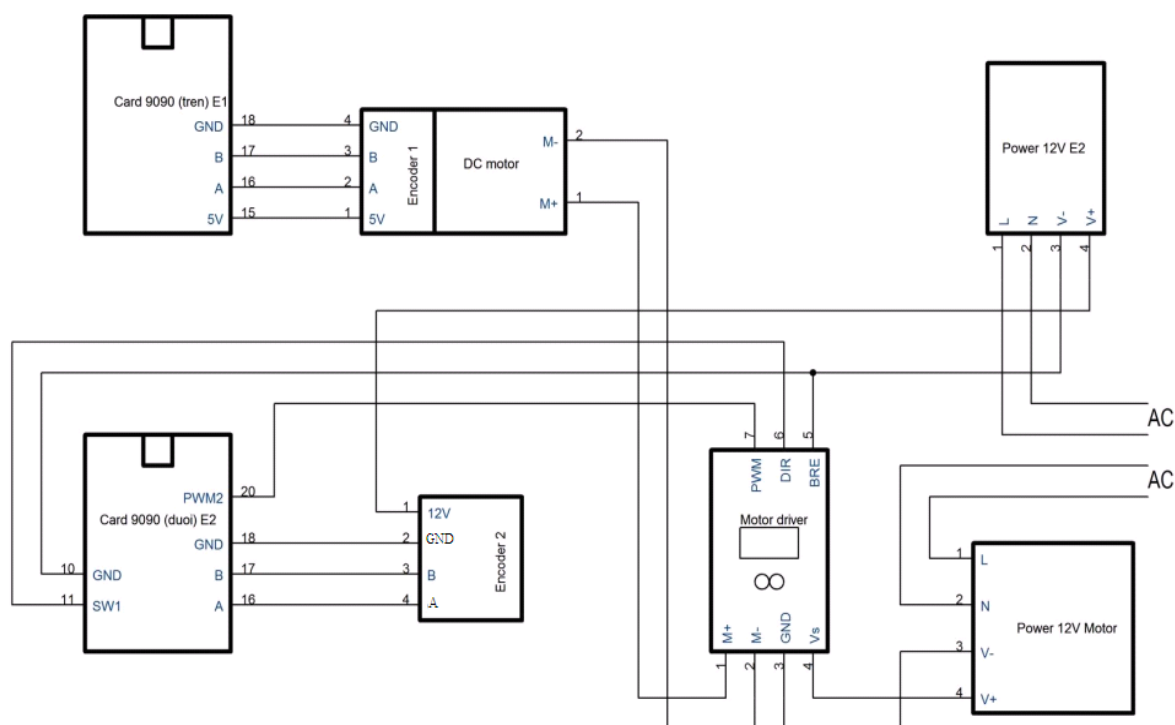
### **Động cơ DC**



Hình 5.6 Động cơ DC

Học viên thiết kế mô hình thực Pendubot được điều khiển cân bằng bởi động cơ Nisca 12 VDC, công suất 33,7 W, tốc độ không tải là 3234 vòng/phút, Mô-men xoắn 54,93(mN · m), có encoder 1 500 xung gắn trực tiếp vào một đầu trục Rotor động cơ DC.

## Sơ đồ điện đấu nối hai mạch giao tiếp máy tính PC điều khiển Hệ Pendubot



Hình 5.7 Sơ đồ điện đấu nối hai mạch giao tiếp máy tính PC điều khiển Hệ Pendubot.

### Mô tả sơ đồ:

Card giao tiếp 9090 E1 được nuôi nguồn điện và thu nhận tín hiệu qua cổng USB kết nối với máy vi tính, có chức năng thu nhận dữ liệu từ Encoder 1 (E1) thông qua kết nối chân các chân A, B tương ứng với Card 9090 E1 với Encoder 1.

Encoder 1 được gắn trên roto của động cơ DC và có chức năng thu nhận góc quay vị trí  $\theta_1$  của động cơ DC, được nuôi nguồn điện từ card 9090 E1 qua các chân tương ứng với GND và + 5V.

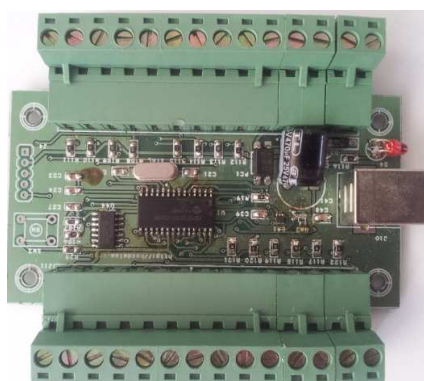
Card giao tiếp 9090 E2 được nuôi nguồn điện và thu nhận tín hiệu qua cổng USB kết nối với máy vi tính, có chức năng thu nhận dữ liệu từ Encoder 2 (E2) thông qua kết nối các chân A, B tương ứng với Card 9090 E2 với Encoder 2, chân GND được nối với -V của bộ nguồn Power 12V E2.

Encoder 2 được gắn trên khớp nối giữa Link 1 và Link 2 và có chức năng thu nhận góc quay vị trí  $\theta_2$  của Link 2 so với Link 1, được nuôi nguồn điện từ bộ nguồn Power 12V E2 qua chân +V, và chân GND của encoder 2 được nối với chân GND của card 9090.

Nguồn nối của Motor driver chân GND, Vs được lấy từ bộ Power 12V E2 chân V-, V+. Chức năng của Motor driver có tác dụng điều khiển động cơ DC qua chân PWM có giá trị 0-500 thể hiện độ mạnh yếu của động cơ DC, chân PWM được nhận tín hiệu từ chân PWM2 của card 9090 E2 có giá trị 0- 5V , nếu ở 0V thì Motor DC quay chiều nghịch và 5V motor DC quay chiều thuận. Ngoài ra chân DIR (Direction) của Motor driver được kết nối với chân SW1 (True hoặc False), sau đó thông qua Motor driver sẽ tương ứng với 2 chế độ quay thuận và quay nghịch của động cơ DC. Chân BRE (Break) được nối với mass. Chân nguồn DC motor được nối với DC Motor driver như sơ đồ hình 4.15.

## 5.2 Phần thu nhập dữ liệu thông qua card USB 9090

Các chức năng của card USB HDL 9090 như thu nhập dữ liệu từ các cảm biến, chế độ ON/OFF, điều chế xung PWM, .... Chức năng digital input giúp học viên dùng ứng dụng linh hoạt hơn.



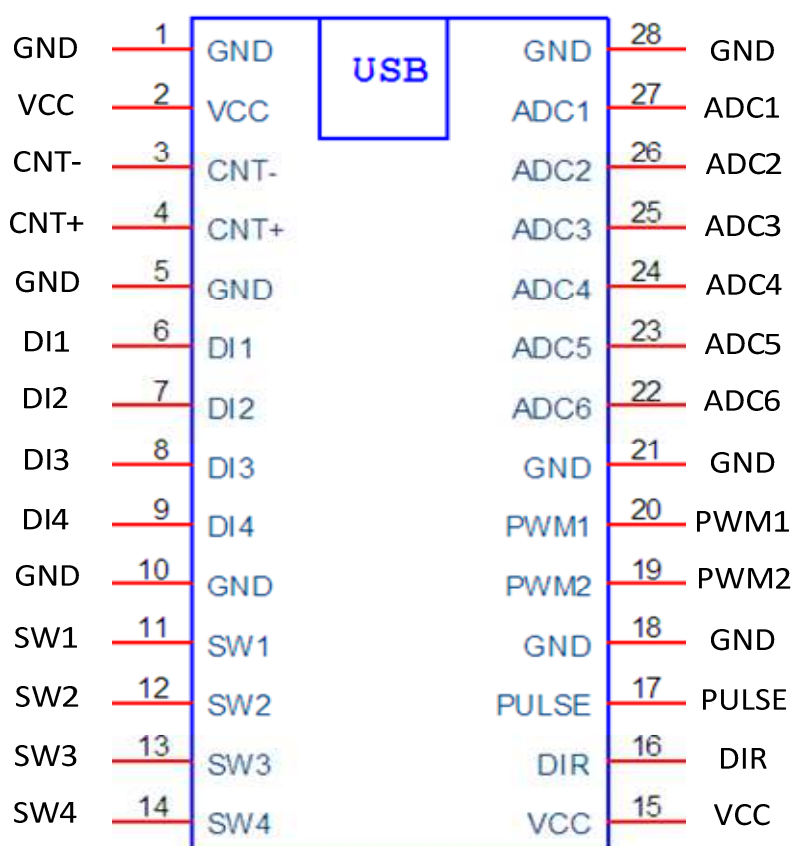
Hình 5.8 Card USB 9090

Chức năng của card USB 9090 trong ứng dụng này là cho phép máy tính có thể giao tiếp với các thiết bị ngoại vi, giúp cho bộ điều khiển trung tâm máy tính có

thể thu nhập dữ liệu, xử lý và xuất tín hiệu điều khiển đạt được độ chính xác và ổn định cao với những tính năng của nó như cho phép truy xuất dữ liệu qua cổng USB:

**Input:** Có các chân đếm Counter (CNT+/CNT-), Tín hiệu vào dạng số 0-5v (DI1-DI4), Set bộ đếm sung lên 5V hay đếm xuống 0V (DIR), Nhận tín hiệu tương tự Analog (ADC1-ADC6)

**Output:** có các chân Tín hiệu ra dạng số 0-5V (DI1-DI4), Xuất tín hiệu PWM để điều khiển (PWM1-PWM2)



Hình 5.9 Sơ đồ chân Card USB 9090

**Giải thích các chân của Card USB HDL 9090:**

Ký hiệu	Mô tả	Loại
GND	Ground – chân mass	Nguồn
VCC	Nguồn 5V lấy từ USB	Nguồn
CNT+/CNT-	Chân đếm Counter	Input
DI1-DI4	Tín hiệu vào dạng số 0-5v	Input
SW1-SW4	Tín hiệu ra dạng số 0-5V	output
DIR	Set bộ đếm sung lên 5V hay đếm xuống 0V	Input
PWM1-PWM2	Xuất tín hiệu PWM để điều khiển	Output
ADC1-ADC6	Nhận tín hiệu tương tự Analog	Input

Bảng 5.1 Bảng các thông số của Card USB HDL 9090

Có thể nói LabVIEW là một ngôn ngữ lập trình-mô phỏng rất mạnh mẽ. Còn rất nhiều toolkit khác mà trong giới hạn đề tài tác giả không thể trình bày hết, với những tính năng vượt trội của card 9090 như tốc độ cao hơn 5 lần và chính xác hơn 4 lần với các card thế hệ cũ. Chính vì vậy trong phần thực hiện đề tài này, học viên quyết định chọn card USB 9090 để thu nhập và xử lý dữ liệu

Trong các phần tiếp theo của đề tài học viên sẽ ứng dụng LabVIEW để giao tiếp với thiết bị phần cứng để điều khiển hệ pendubot với Link2 tự cân bằng thẳng đứng với  $\theta_2 = 0^\circ$  khi Link1 ở bất kỳ vị trí góc quay  $\theta_1 \neq 0^\circ$



## CHƯƠNG 6: ỨNG DỤNG LABVIEW ĐIỀU KHIỂN HỆ PENDUBOT

Chương này học viên trình bày phương pháp điều khiển hệ Pendubot thực bằng bộ điều khiển PID cải tiến trong môi trường Labview, và thực nghiệm trên mô hình thật.

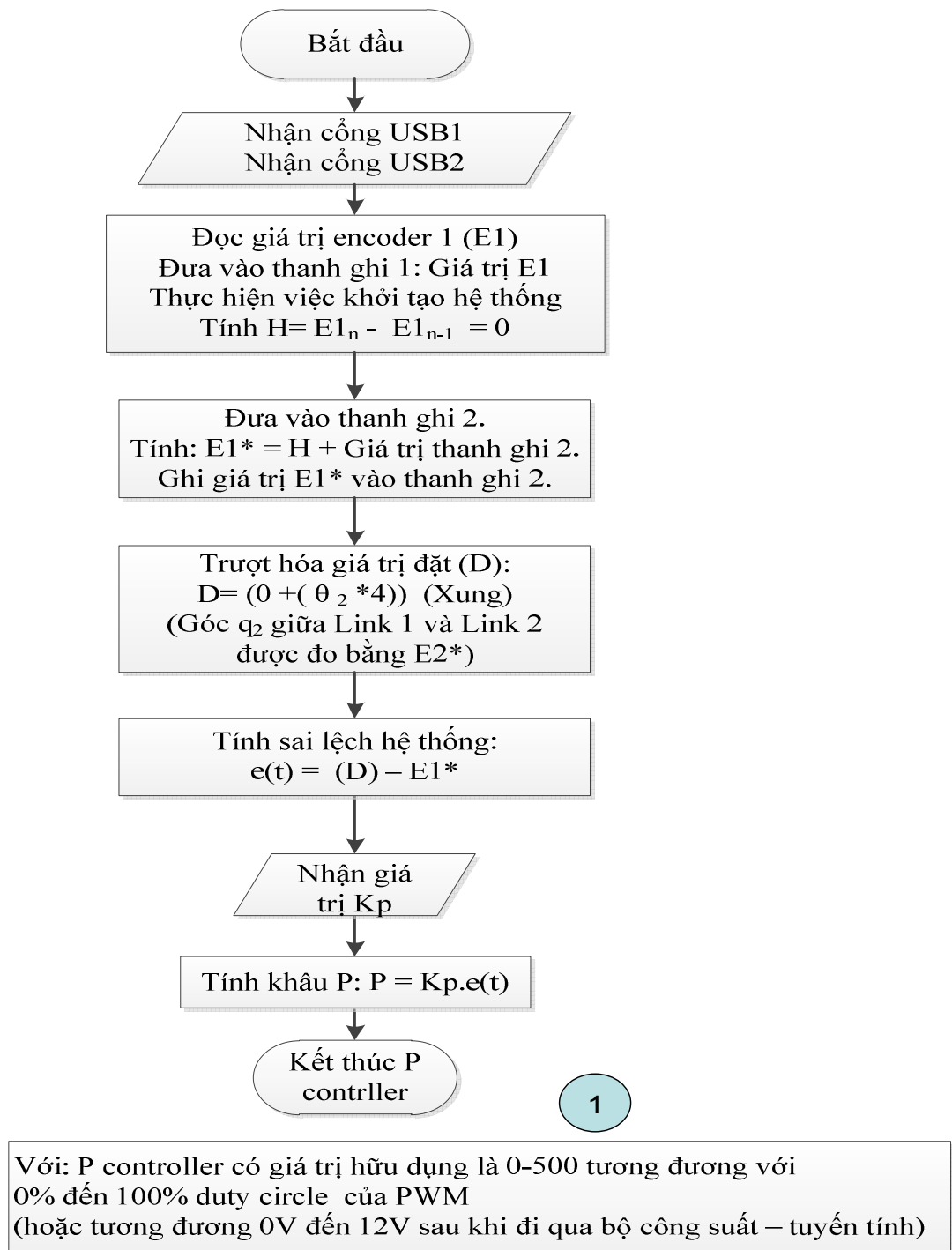
### 6.1 Thực nghiệm và khảo sát chương trình điều khiển hệ Pendubot

#### 6.1.1 Lưu đồ giải thuật điều khiển hệ Pendubot trên nền LabVIEW

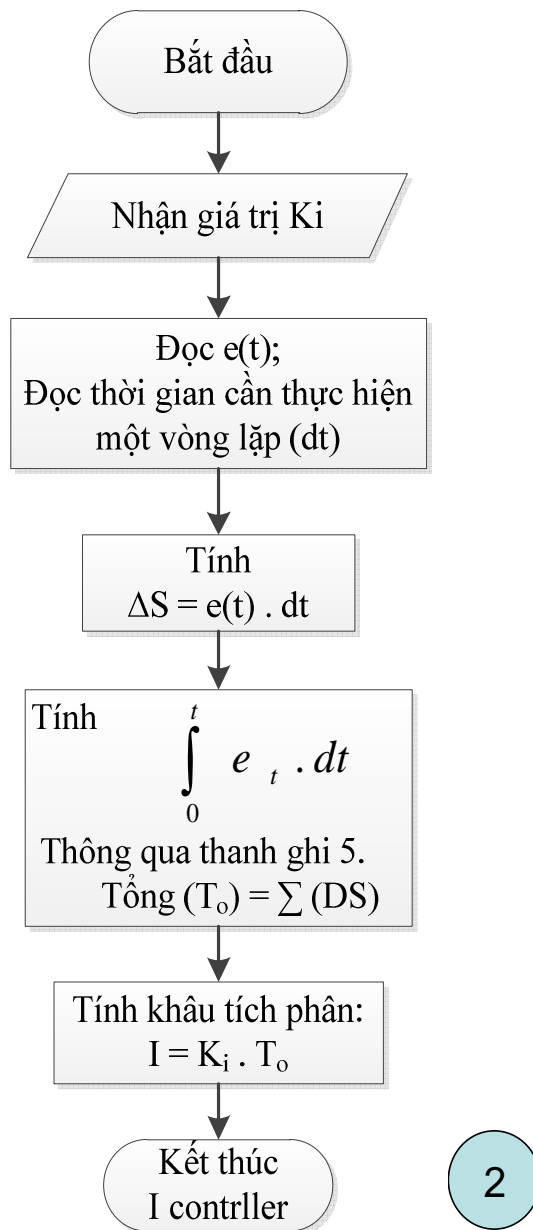
Bộ điều khiển PI là sự kết hợp của hai thành phần : Thành phần tỉ lệ (P) đưa ra giá trị  $P_{out}$  dựa trên giá trị sai lệch tại thời điểm hiện tại, thành phần tích phân (I) đưa ra giá trị  $I_{out}$  dựa trên tích lũy các giá trị sai lệch từ quá khứ đến thời điểm hiện tại.

Theo sơ đồ khối của hệ Pendubot hình 4.1, cho thấy hệ thống điều khiển sử dụng hai card 9090 để thu thập dữ liệu giao tiếp giữa hệ thống thực và máy tính. Do card 9090 chỉ có một ngõ counter nên mỗi card chỉ quan tâm đến một góc quay  $\theta_1, \theta_2$  tương ứng với vị trí của Link1 và Link2. Do đó học viên chỉ sử dụng bộ điều khiển PI với hồi tiếp từ góc lệch  $\theta_1, \theta_2$  tương ứng với vị trí của Link1 và Link2 để giữ cho Link 2 luôn thẳng đứng ổn định với góc Link 1 ở bất kỳ vị trí nào. Hình 5.4, 5.5, 5.6, là lưu đồ giải thuật điều khiển con lắc ngược bằng bộ điều khiển PI và bù mô-men xoay phi tuyến trên hệ thống thực.

**Lưu đồ giải thuật điều khiển con lắc ngược bằng bộ điều khiển PI và bù mô-men xoay phi tuyến trên hệ thống thực.**

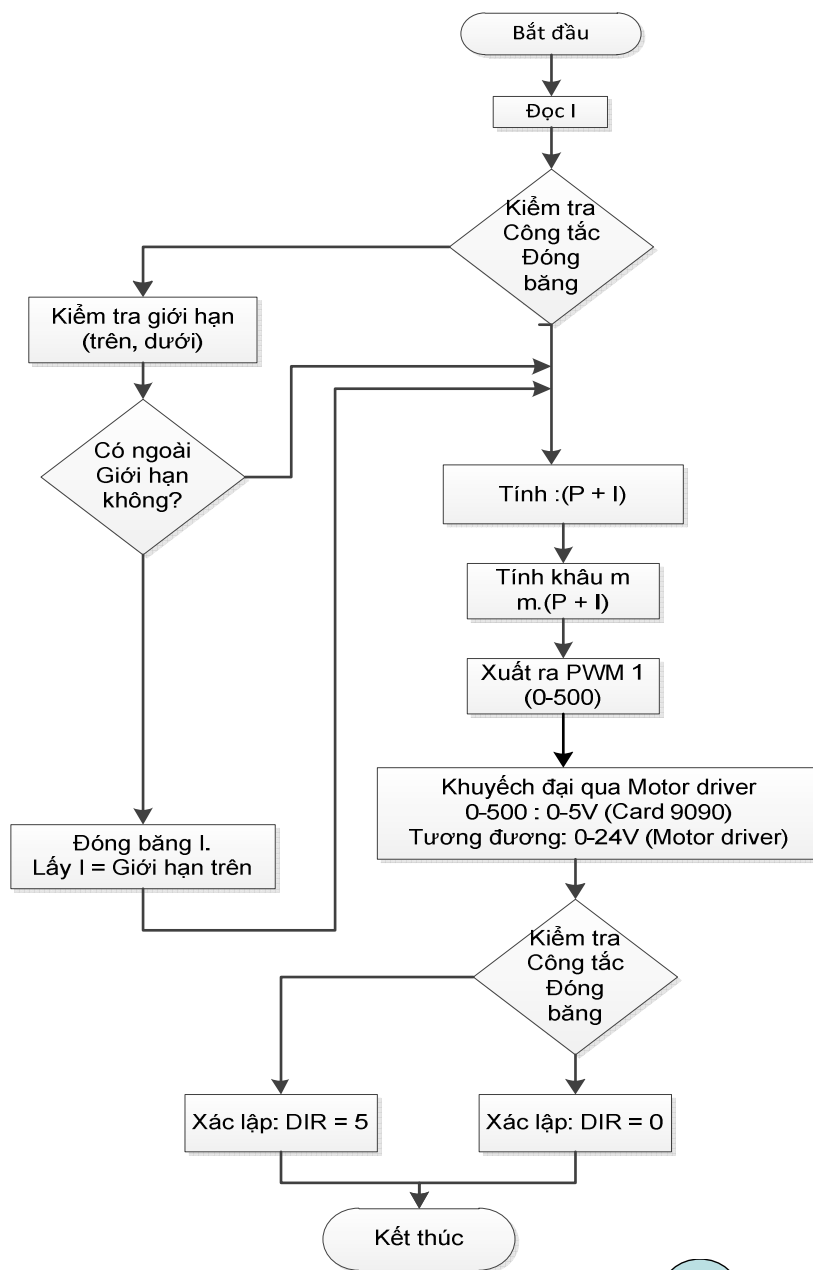


Hình 6.1 Sơ đồ giải thuật tính khâu P



Với: I controller có giá trị hữu dụng là 0-500 tương đương với 0% đến 100% duty circle của PWM (hoặc tương đương 0V đến 12V sau khi đi qua bộ công suất – tuyến tính)

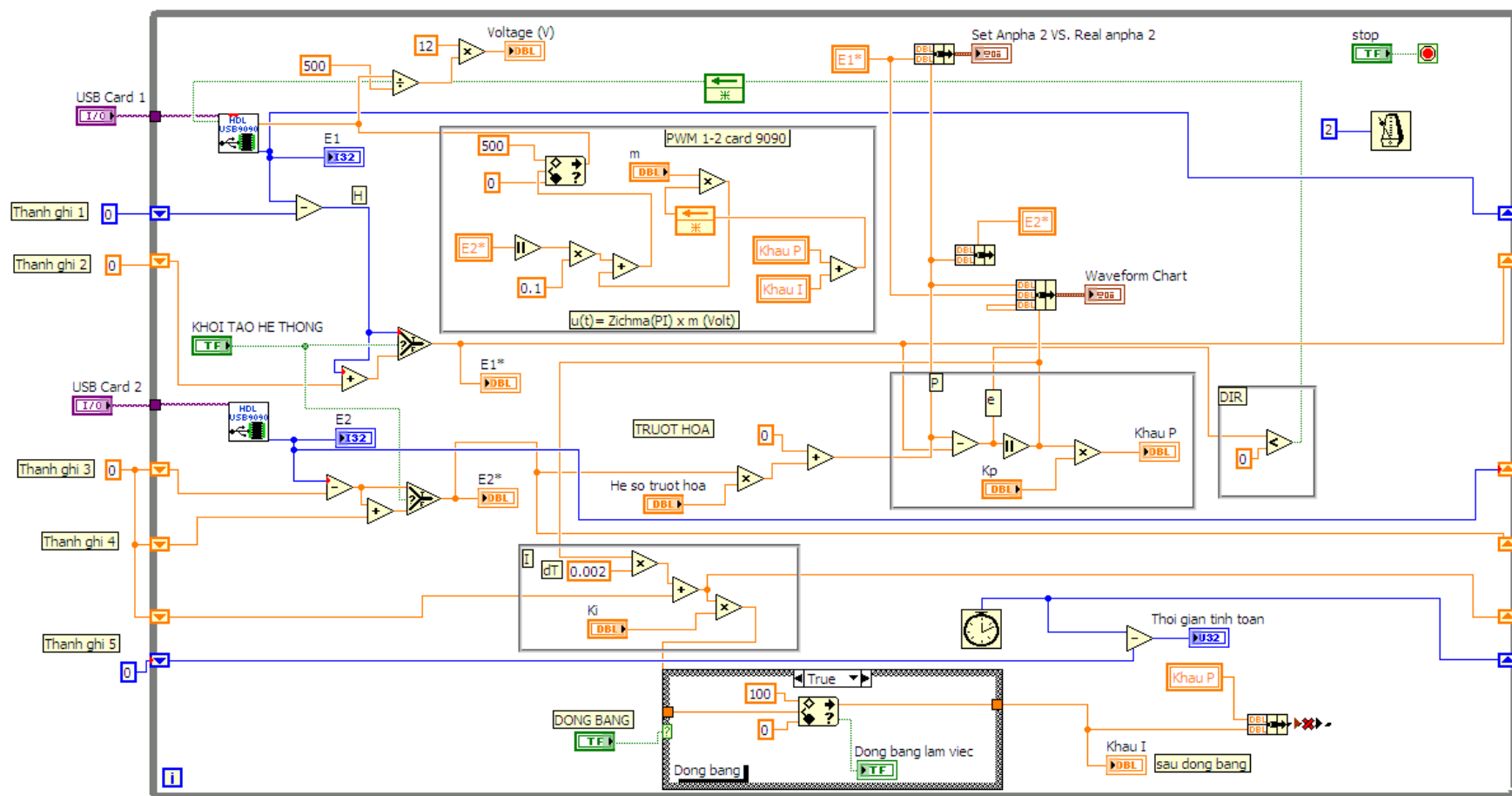
Hình 6.2 Hình sơ đồ giải thuật Khâu I



3

Hình 6.3 Sơ đồ giải thuật điều khiển hệ Pendubot bằng thuật toán PI

Dựa vào giải thuật điều khiển trên thiết lập chương trình điều khiển trong môi trường Labview. Trong thư viện của Labview lấy các khối để giao tiếp giữa hai card 9090 với máy tính và mô hình thật Pendubot, để xây dựng chương trình điều khiển như hình 5.7

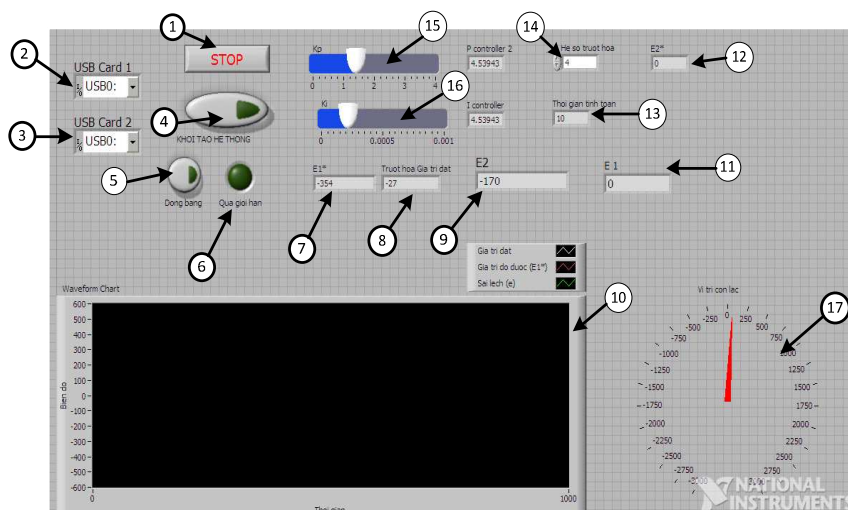


Hình 6.4 Chương trình điều khiển Hệ Pendubot trên nền Labview

### 6.1.2 Trang giao diện chính của chương trình.

Hình 5.8 là giao diện điều khiển hệ thống tin Pendubot trên nền Labview. Đây là trang điều khiển chính của phần điều khiển, hiển thị các thông tin giao tiếp từ máy tính đến mô hình hệ thống thực Pendubot, điều chỉnh hệ số  $K_p$ ,  $K_i$  bằng phương pháp thực nghiệm lấy mẫu nhiều lần với các bộ thông số  $K_p$ ,  $K_i$  khác nhau cuối cùng đã chọn được các thông số mong muốn thích hợp để điều khiển hệ Pendubot ổn định (Sẽ được trình bày ở phần tiếp theo), kết quả đáp ứng ngõ ra của bộ PI sau khi đã lấy giá trị dương cũng sẽ được hiện trên giao diện chính (File code lập trình cũng sẽ được trình bày trong phần phụ lục)

### 6.1.3 Chuẩn bị thực nghiệm và khảo sát chương trình nội dung thực



hiện

Hình 6.5 Giao diện điều khiển hệ Pendubot trên nền Labview

#### Giải thích giao diện

- |                         |                     |                      |                          |
|-------------------------|---------------------|----------------------|--------------------------|
| 1. Stop                 | 2. USB card 1       | 3. USB card 2        | 4. Khởi tạo hệ thống     |
| 5. Đóng băng            | 6. Quá giới hạn     | 7. $E18^*$           | 8. Trượt hóa giá trị đặt |
| 9. $E2$                 | 10. Đồ thị quá độ   | 11. $E1$             | 12. $E2$                 |
| 13. Thời gian tính toán | 14. Hệ số trượt hóa | 15. Điều chỉnh $K_p$ |                          |
| 16. Điều chỉnh $K_i$    | 17. Vị trí con lắc  |                      |                          |

Cấp nguồn điện 220V vào hệ Pendubot, thông qua các bộ nguồn xuất ra các điện áp 12V cho hai encoder và hai card 9090, Motor driver, đầu nối các cổng USB 1 và USB 2 với máy tính. Thiết lập các giải thuật trong labview. trình tự thực hiện như sau.

Bước 1: Đầu nối dây cáp cổng USB của card USB 9090 vào máy tính.

Bước 2: Trong giao diện window XP hoặc Win 7 , click phải chuột vào My computer → chọn Device manager/Manage → xem cổng giao tiếp đã được thiết lập hai card USB 9090.

Bước 3: Quay về giao diện chính của Labview, ngay vị trí 2 và 3 chọn cổng giao tiếp hai card USB 9090 đã chọn ở bước 2.

Bước 4: Tại vị trí 4 chọn khởi tạo hệ thống

Bước 5: Chọn điều chỉnh thông số  $K_p$ ,  $K_i$

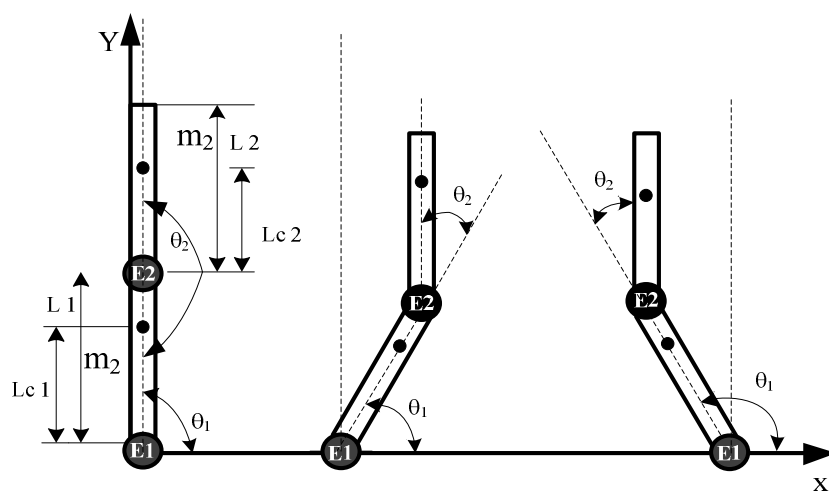
Bước 6: Xác định thông số trượt hóa, thời gian tính toán

Bước 7: Vị trí 5 chọn đóng băng

Bước 8: tương trình hoạt động

Bước 9: Dừng chương trình nhấn nút Stop vị trí 1

Để tính được các khâu P và khâu I, học viên đầu tiên chuẩn bị các thuật ngữ, định nghĩa các đại lượng thông số cho phương pháp tính toán xác định các góc  $\alpha_1$  và  $\alpha_2$  tương ứng theo hình vẽ 5.10 sao cho khi chạy chương trình và khởi tạo hệ thống , xác định các thông số  $K_p$   $K_i$  thích hợp bằng phương pháp thực nghiệm của Ziegler – Nichols. Để chương trình đưa ra đáp ứng tốt nhất khi tác động một lực nhỏ bất ở các hướng khác nhau trên Link 2 cân bằng thẳng đứng ( $\theta_2 = 0^\circ$ ) ở vị trí bất kỳ đối với Link 1 thay đổi vị trí bất kỳ ( $\theta_1 \neq 0^\circ$ ). Các góc  $\theta_2$   $\theta_1$  được xác định bởi số xung phát ra từ hai encoder 1 và encoder 2.

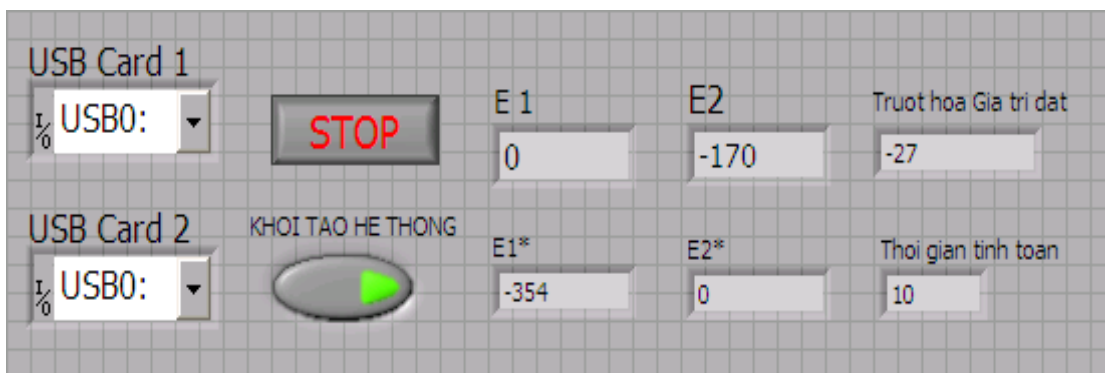


Hình 6.6 Biểu diễn Link 2 cân bằng thẳng đứng ( $\theta_2 = 0^\circ$ ) ở vị trí bất kỳ đối với Link 1 thay đổi vị trí bất kỳ ( $\theta_1 \neq 0^\circ$ ).

## 6.2 Nhận dạng và điều khiển hệ Pendubot trên nền Labview

### 6.2.1 Khởi tạo hệ thống

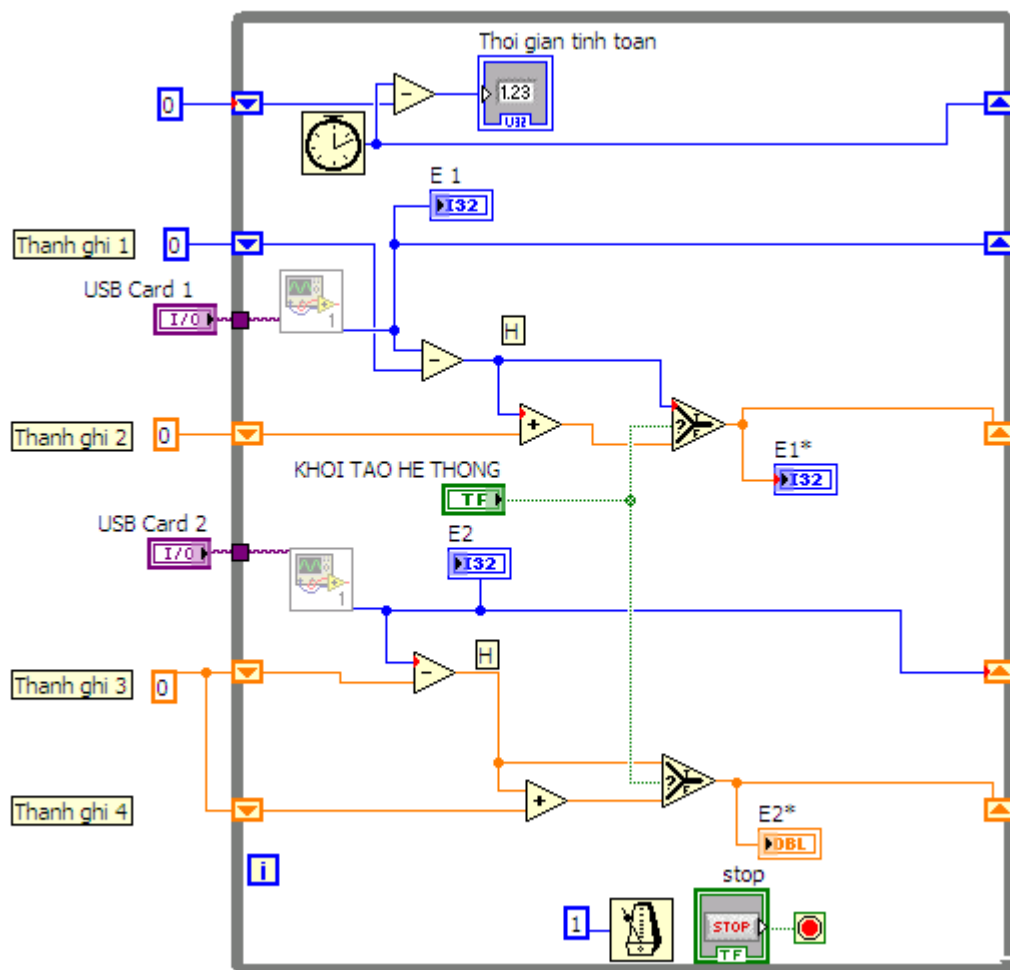
#### Giao diện khởi tạo hệ thống



Hình 6.7 Giao diện Khởi tạo hệ thống



## Thuật toán của chương trình labview thực hiện việc khởi tạo hệ thống



Hình 6.8 Thuật toán của chương trình labview thực hiện việc khởi tạo hệ thống

### Giải thích:

- **Thanh ghi:** Mặc định đây là một đường hàm truyền dữ liệu giá trị hiện tại vị trí 1 theo hình 5.12 cũng chính là giá trị chính nó tại vị trí 2, các giá trị tại hai vị trí này luôn cập nhật liên tục sau một thời vòng lặp .
- **Vòng lặp:** Là một khoảng thời gian chạy hết chương trình của một vòng lặp.

- **Khởi tạo hệ thống:** Bước này đầu tiên cho chạy chương trình, khởi tạo để xác định cho giá trị E1, E2

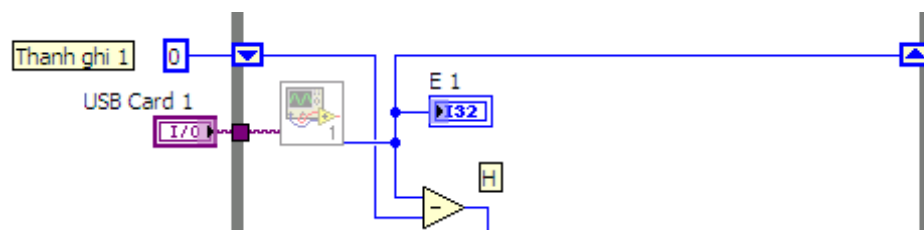
- **Đọc giá trị Encoder E1:** Đầu tiên cắm card USB 1 vào để máy nhận tín hiệu đầu vào của E1. Do E1 không phải lúc nào là một giá trị nhất định do phụ thuộc vào góc quay lúc ban đầu, do vậy đưa giá trị ban đầu E1 vào thanh ghi 1. Mục đích là biến giá trị bất kỳ của E1 thành một giá trị bằng 0°.

**Đặt:** Giá trị E1 hiện tại là E1’

Giá trị E1 sau khi hệ thống chạy chương trình có vóng lặp đầu là E1’’

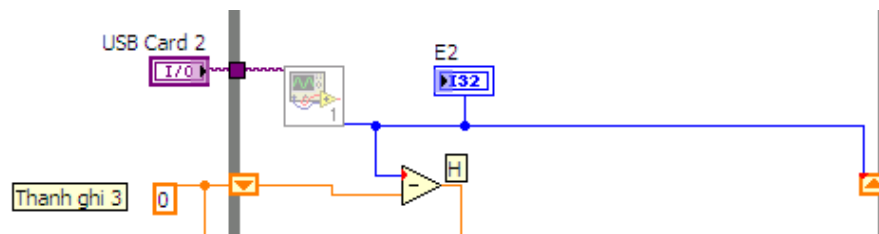
- **Thuật toán tính giá trị H**

$$H = E1' - E1''$$



Hình 6.9 Thuật toán tính giá trị H

- **Đọc giá trị Encoder E2:** Cắm card USB 2 vào để máy nhận tín hiệu đầu vào của E2. Do E2 không phải lúc nào là một giá trị nhất định do phụ thuộc vào góc quay lúc ban đầu, do vậy đưa giá trị ban đầu E2 vào thanh ghi 3. Mục đích là biến giá trị bất kỳ của E1 thành một giá trị bằng 0°.

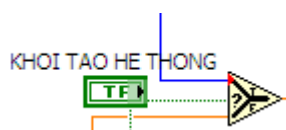


Hình 6.10 Đọc giá trị Encoder E2

- **Thuật toán khởi tạo hệ thống:** Để nhận biết được các góc  $\theta_2$   $\theta_1$  nguyên lý chung là dùng cảm biến encoder E1 và E2, dựa vào chu kỳ xung của E1 và E2 ta nhận biết được các góc  $\theta_2$   $\theta_1$ .

Encoder 1 có giá trị mới E1\* nhờ công tắc khởi tạo hệ thống, để khởi tạo ra giá trị E1\* và thanh ghi 2 vậy ta xác định được góc  $\theta_1$ .

+ Khởi công tắc khởi tạo hệ thống

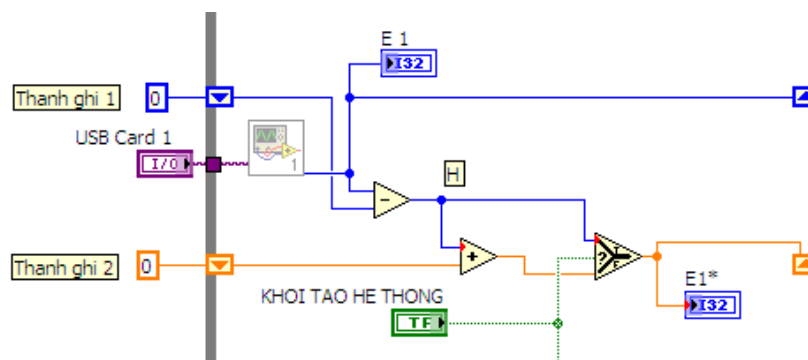


+ Hàm select



Nhờ khối công tắc khởi tạo hệ thống ta có công thức sau sau

$$E1^* = \begin{cases} E1' - E1'' = 0 \\ (E1' - E1'') + E1^* \end{cases} \quad \begin{matrix} S = true & \text{Với} \\ S = false & \text{Với} \end{matrix}$$



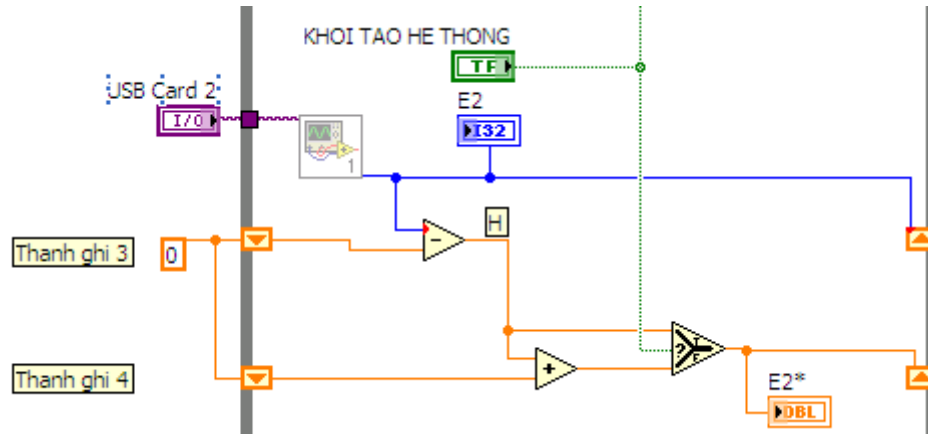
Hình 6.11 Giải thuật khởi tạo ra giá trị encoder 1 E1\*

Tương tự Encoder 2 có giá trị mới E2\* nhờ công tắc khởi tạo hệ thống, để khởi tạo ra giá trị E2\* vào thanh ghi 4 vậy ta xác định được góc  $\theta_2$ .

Với

Với

$$E2^* = \begin{cases} E2' - E2'' = 0 \\ (E2' - E2'') + E2^* \end{cases} \quad \begin{matrix} S = true \\ S = false \end{matrix}$$

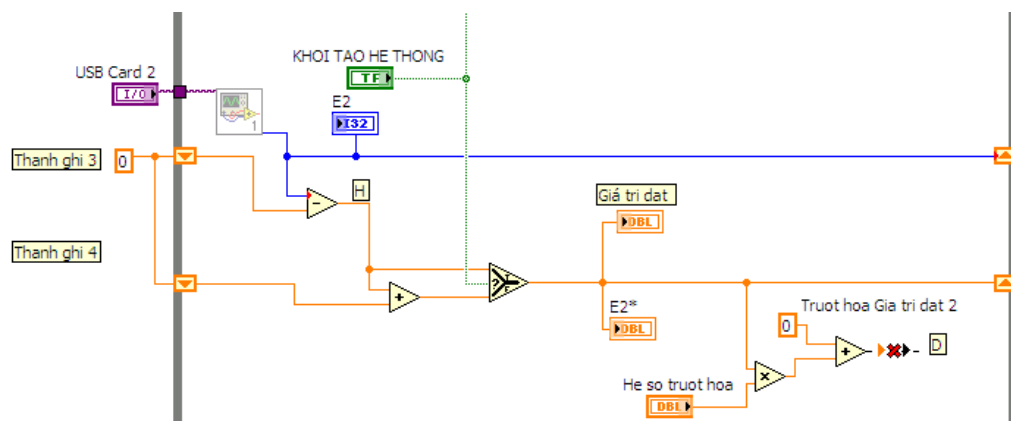


Hình 6.12 Giải thuật khởi tạo ra giá trị encoder 2 E2\*

### 6.2.2 Giá trị đặt, Trượt hóa giá trị đặt

Lúc khởi tạo giá trị đặt là bằng 0 nhưng khi chương trình chạy thì giá trị đặt luôn biến thiên, và góc đo  $\theta_2$  của Link 2 so với phương thẳng đứng Link 1 đo đó góc  $\theta_2$  luôn biến thiên khi Link 1 thay đổi một góc  $\theta_1$ .

#### Chương trình thuật toán labview thực hiện trượt hóa giá trị đặt

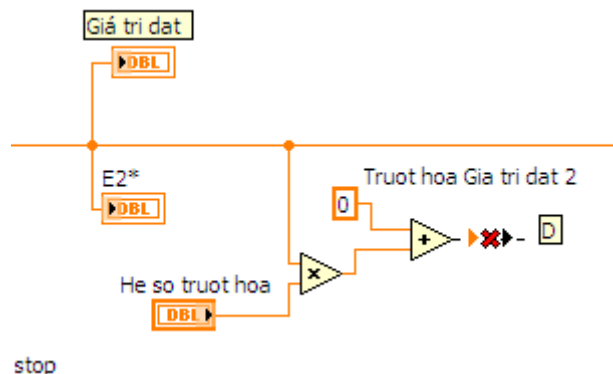


Hình 6.13 Chương trình thuật toán labview thực hiện trượt hóa giá trị đặt

**Giải thích:**

**Trượt hóa giá trị đặt:** Do giá trị đặt liên tục luôn bằng 0 nhưng do hệ thống làm việc liên tục do đó ta phải trượt hóa để giá trị đặt không thay đổi tùy vào góc quay  $\theta_1$  do đó giá trị đặt sẽ biến thiên.

### Thuật toán trượt hóa giá trị đặt E2\*



Hình 6.14 Thuật toán trượt hóa giá trị đặt E2\*

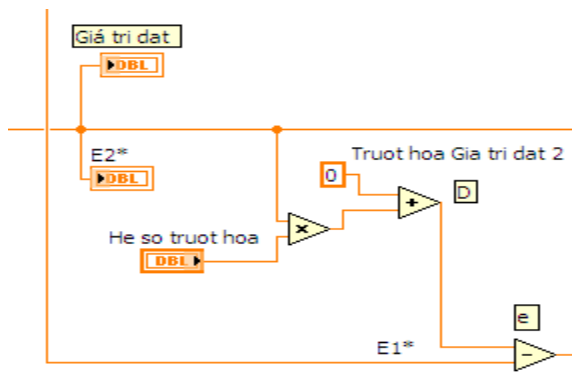
### Phương trình trượt hóa giá trị đặt

$$D = (0 + (\theta_2 * 4)) \text{ (Xung)}$$

Đối với điều khiển PID bình thường thì đặc tính giá trị đặt thường cố định  $D = \text{const}$ , với điều khiển PI hệ Pendubot thì giá trị đặt  $D$  thường thay đổi theo vị trí đứng của Link 2 so với Link 1 góc  $\theta_2$  luôn biến thiên. Do đó ta phải trượt hóa giá trị đặt thì hệ thống mới ổn định được và tính được sai lệch hệ thống  $e$ .

### 6.2.3 Tính sai lệch hệ thống và nhận giá trị $K_p$ :

Gọi  $e$  là sai lệch hệ thống, ta có phương trình thuật toán sau



Hình 6.15 Thuật toán tính sai lệch hệ thống  $e$

### Phương trình tính sai lệch hệ thống

$$e(t) = D - E1^*$$

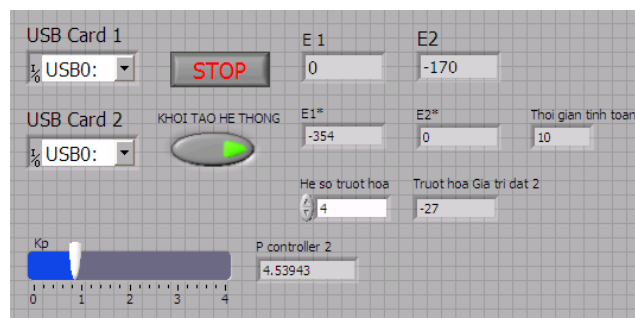
#### 6.2.4 Nhập giá trị $K_p$ :

Giá trị  $K_p$  hằng số tỉ lệ, học viên thực hiện bằng phương pháp phương pháp thứ hai của Ziegler – Nichols. Điều đặc biệt là phương pháp này không sử dụng mô hình toán học của đối tượng điều khiển, ngay cả mô hình xấp xỉ gần đúng, kết quả lựa chọn  $K_p$  trong khoảng 0.5 đến 0.8.

#### 6.2.5 Tính khâu P:

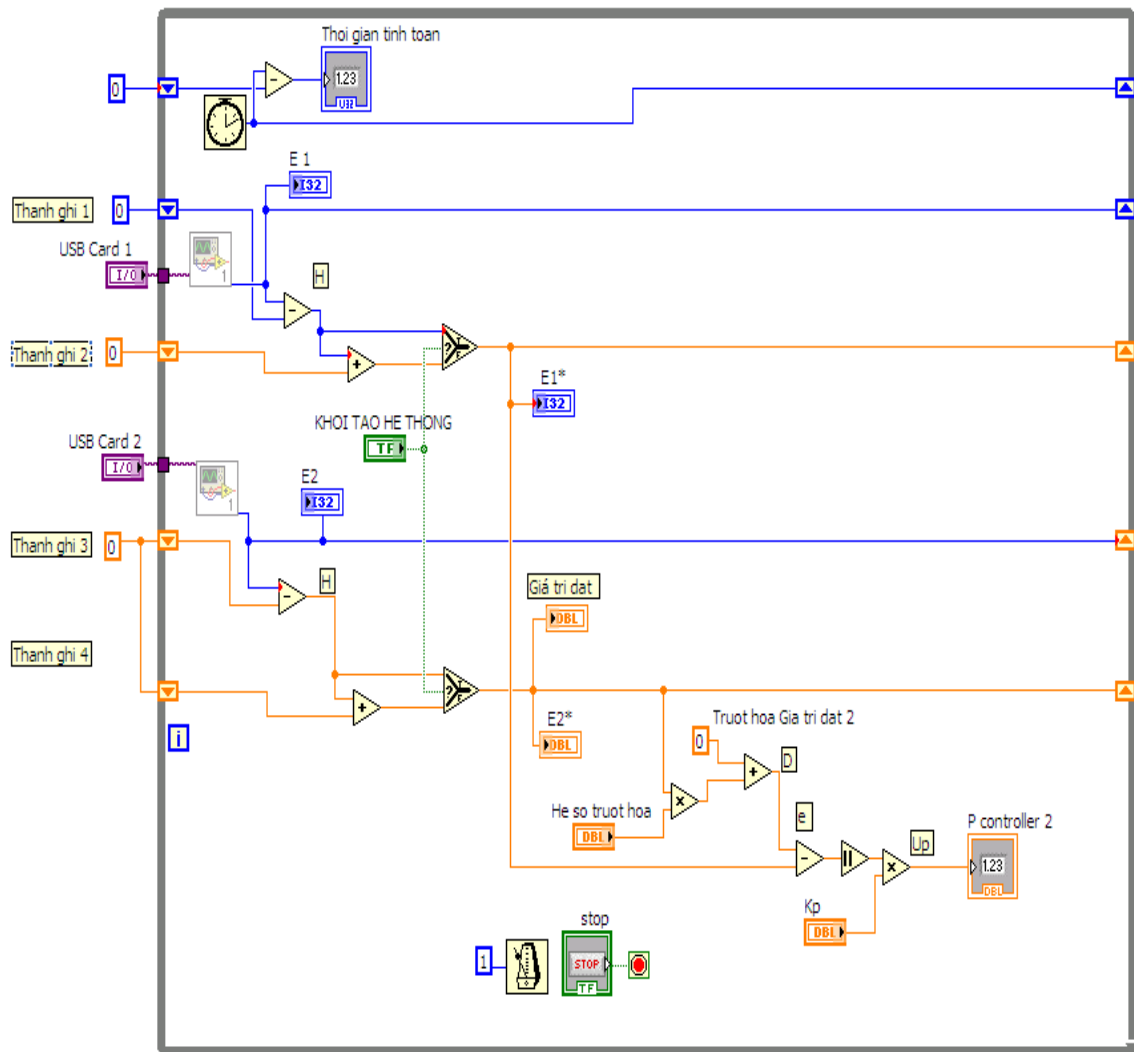
Khâu P tạo ra tín hiệu điều khiển tỉ lệ với giá trị của sai lệch. Việc này được thực hiện bằng cách nhân sai lệch  $e$  với hằng số  $K_p$  – gọi là hằng số tỉ lệ. Khâu P được tính dựa trên công thức:  $P = K_p * e(t)$

### Giao diện điều khiển khâu P



Hình 6.16 Giao diện chính điều khiển khâu P

## Chương trình thuật toán labview thực hiện tính khâu P



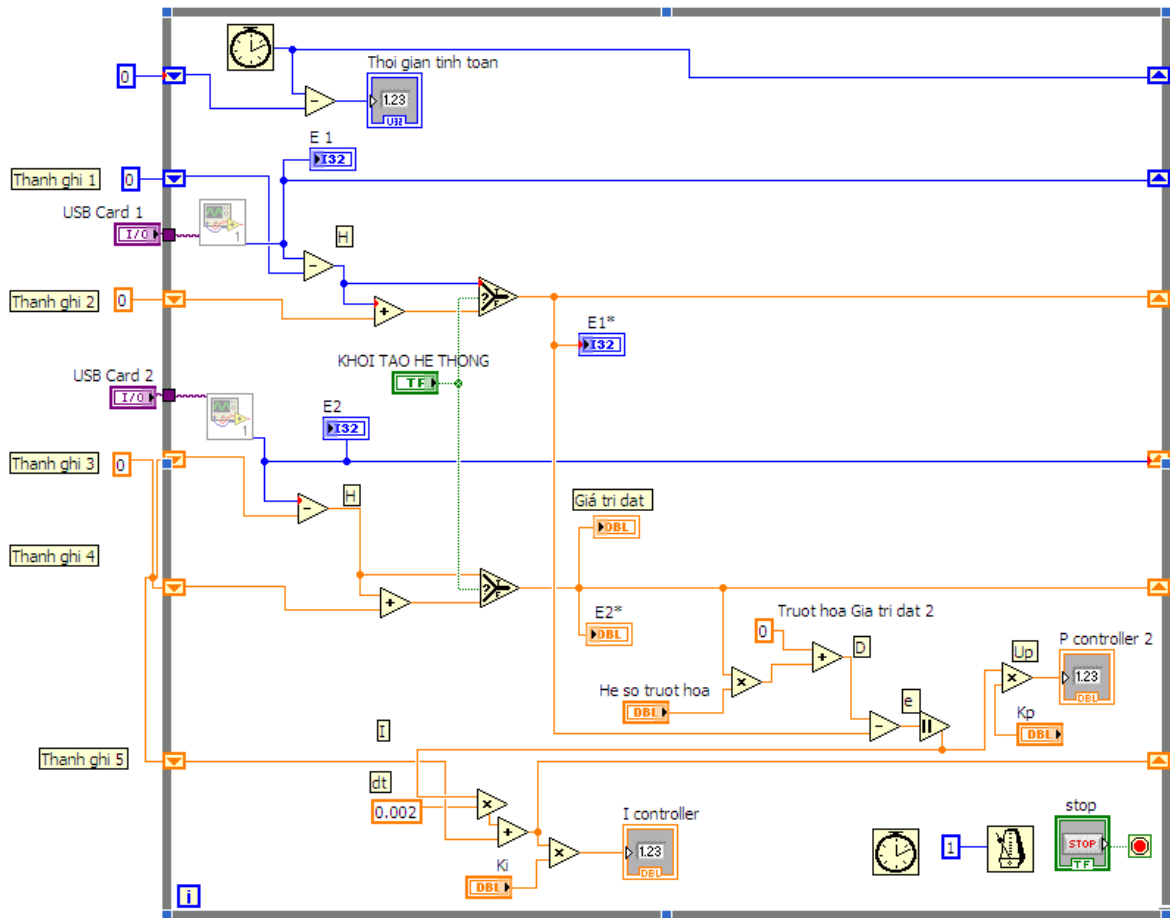
Hình 6.17 Chương trình thuật toán labview thực hiện tính Khâu P

**Kết Luận:** P controller có giá trị hữu dụng là 0-500 tương đương với 0% đến 100% duty circle của PWM (hoặc tương đương 0V đến 24V sau khi đi qua bộ công suất – tuyến tính).

### 6.2.5 Tính khâu I:

Khâu I cộng thêm tổng các sai số trước đó vào giá trị điều khiển. Việc tính tổng các sai số được thực hiện liên tục cho đến khi giá trị đạt được bằng với giá trị đặt, và kết quả là khi hệ cân bằng thì sai số bằng 0.

## Chương trình thuật toán labview thực hiện tính khâu I



Hình 6.18 Chương trình thuật toán labview thực hiện tính khâu I

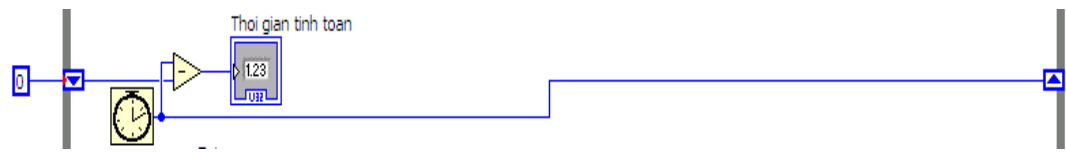
Giải thích: chương trình thuật toán tính khâu I

1. Nhận giá trị  $K_i$ :

2. Nhập giá trị sai lệch  $e(t)$  là giá trị đặt (giá trị kỳ vọng muốn góc  $\theta_2$  giữa Link 2 với phương thẳng đứng Link 1), trừ đi giá trị đo được E1

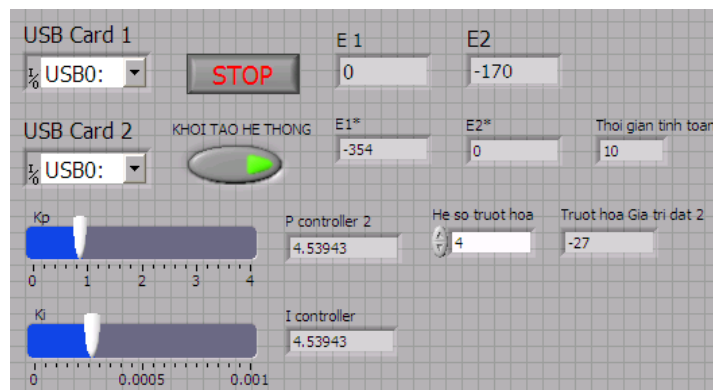
3. Thời gian lấy mẫu đặt  $dt = 0,002$  thời gian chạy hết chương trình vòng lặp được đo bằng lấy mẫu, do thời gian lấy mẫu đặt  $10ms = 0,0001$  quá lớn nên phải chọn  $K_i$  quá nhỏ do đó học viên chọn thời gian lấy mẫu  $dt = 0.002$  để chọn thông số để chỉnh định  $K_i$  lớn nhờ giải thuật toán sau.



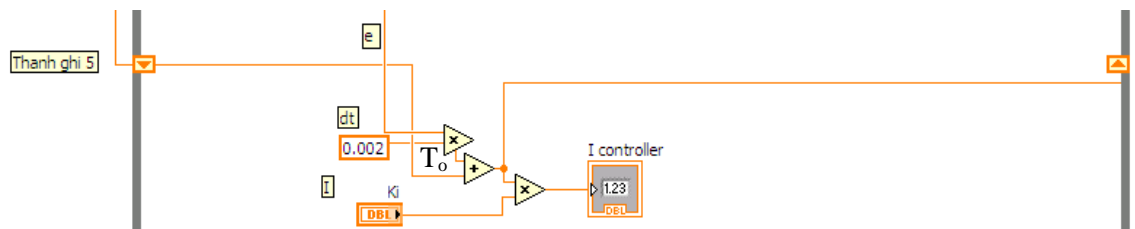


Hình 6.19 Giải thuật thời gian lấy mẫu

4. Khâu I được tính theo công thức: Giải thuật trong hình trình bày phương pháp tính toán khâu I, và giao diện nhập dữ liệu.



Hình 6.20 Giao diện nhập dữ liệu nhập giá trị Ki và tính Khâu I



Hình 6.21 Giải thuật trình bày phương pháp tính toán khâu I.

Tính : 
$$\int_0^t e_t . dt$$

Thông qua thanh ghi 5 tính tổng  $T_0$  :  $Tổng (T_o) = \sum (\Delta S)$

Tính  $\Delta S$ :  $\Delta S = e(t).dt$  (giá trị tức thời )

Tính khâu tích phân: 
$$I = K_i . = K_i . \int_0^t e_t . dt$$

**Kết luận:** I controller có giá trị hữu dụng là 0-500 tương đương với 0% đến 100% duty circle của PWM (hoặc tương đương 0V đến 12V sau khi đi qua bộ công suất – tuyến tính).

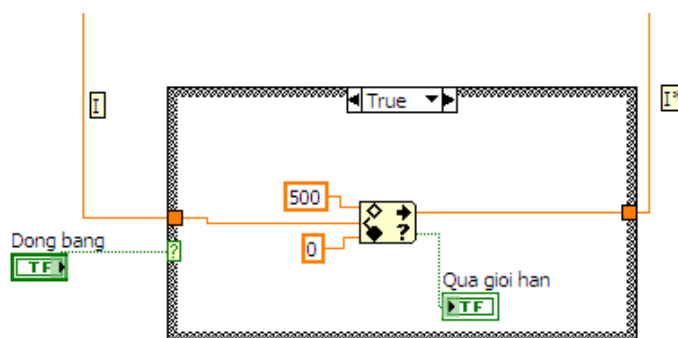
### 6.3.6 Đóng băng “Chức năng tích phân”

Ta xét lúc đầu chạy chương trình:

+ 0 là thời gian lúc bắt đầu chạy.

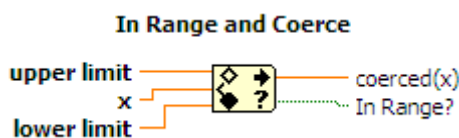
+ t là lúc thời gian tắt chương trình.

Do đó khi chương trình tính lâu thì tạo ra trị e rất lớn với thời gian kéo dài sẽ làm cho giá trị  $u(t)$  của khâu PI rất lớn và điện áp rất lớn dẫn tới mất ổn định động cơ DC và hệ thống bị hư hại. Do đó phải dùng chức năng đóng băng tích phân để chống chèn giá trị e.



Hình 6.22 Khối Giải thuật trình bày phương pháp đóng băng tích phân.

- Khối In range coerce (trong khoảng). Dựa vào mô hình thực học viên thực nghiệm giới hạn chèn I từ 0-500.

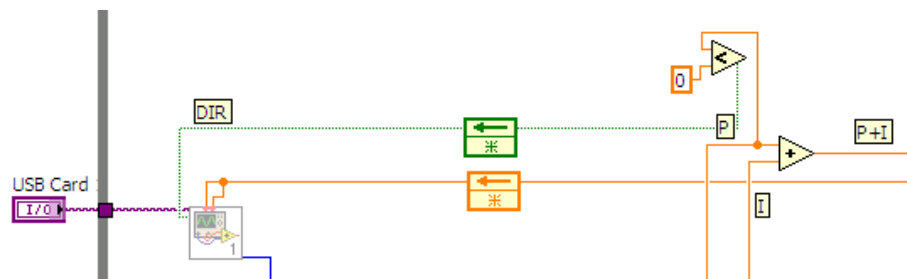


Hình 6.23 Hàm In range coerce (trong khoảng)

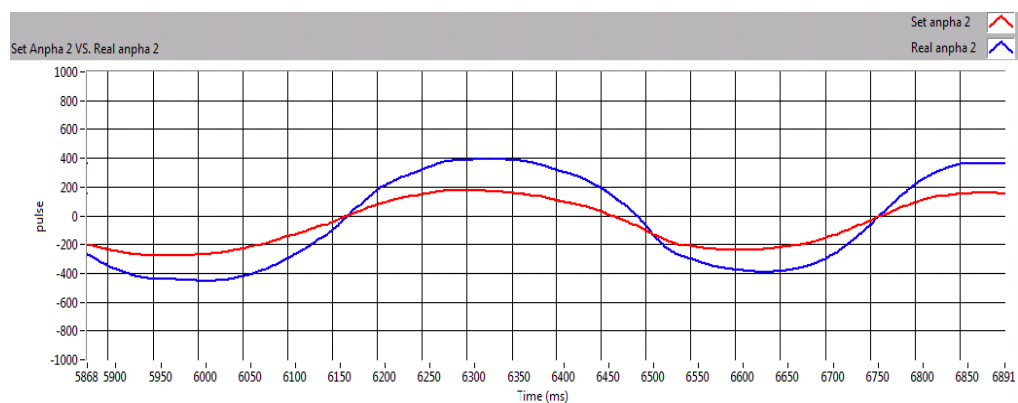
### 6.3.7 Tính khâu PI

Sau khi tính được khâu P và khâu I, theo phương trình tính Khâu PI được tính theo công thức:  $u(t) = Ki.e + \int_0^t e(t).d(t)$

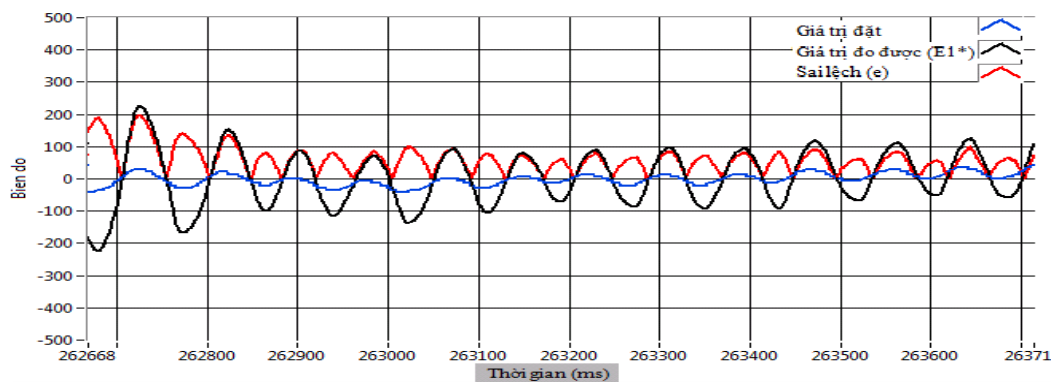
Giải thuật trong hình 6.24 trình bày phương pháp tính toán khâu PI.



Hình 6.24 Khối Giải thuật trình bày tính khâu PI.



Hình 6.25 Kết quả Đáp ứng giá trị đặt khi chưa bù mômen xoay.



Hình 6.26 Kết quả đáp ứng khi chưa có Bù Momen xoay, Ki hiệu chỉnh lớn

### **Kết luận:**

Từ những kết quả đáp ứng khâu PI trên hình 5.27 5.28, nhận thấy hệ pendubot không ổn định do moment xoay gọi là  $M$  của động cơ DC chứa giá trị  $\cos(\theta_1)$  tạo nên thành phần moment phi tuyến.

Moment này có xu hướng kéo hệ vật Pendubot (encoder 1,2, link 1, 2) đi xuống theo phương thẳng đứng với mọi góc  $\theta_2$  giữa Link 1 và Link 2. (với độ trùng khác nhau ở những góc  $\theta_{1,2}$  khác nhau).

Phần tiếp theo học viên bằng phương pháp thực nghiệm trên mô hình thực bằng thuật toán bù mô-men xoay để hệ Pendubot cân bằng ổn định nhất.

## **6.3 Nhận dạng và điều khiển Hệ pendubot bằng phương pháp bù mômen xoay phi tuyến**

### **6.3.1 Cơ sở lý thuyết:**

Ta có công thức  $M = F.d = \cos(\alpha).L_1.F$   
 $= \cos(\alpha).L_1.m.g$

Trong đó:

$M$  = khối lượng cơ cấu tạo mô men xoay.

$g = 9.8m/s^2$

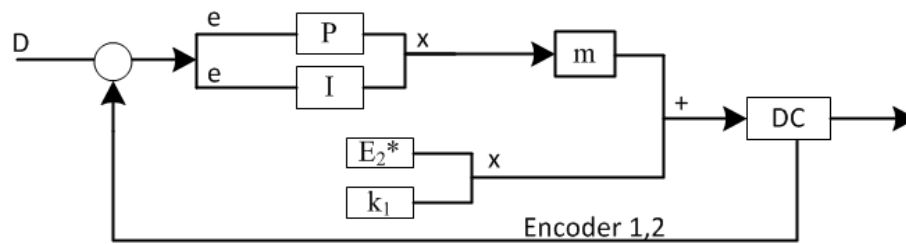
$L_1$  = Chiều dài Link 1

$\alpha$ : (90 - Góc xoay Link1) độ.

Vì Moment xoay  $M$  chứa giá trị  $\cos(\alpha)$  tạo nên thành phần moment phi tuyến.

Moment này có xu hướng kéo hệ vật (encoder, link 1, 2) đi xuống theo phương thẳng đứng. (với độ trùng khác nhau ở những góc anpha khác nhau).

Để giải quyết bài toán này chúng ta đề xuất bù thêm thành phần m như hình sau



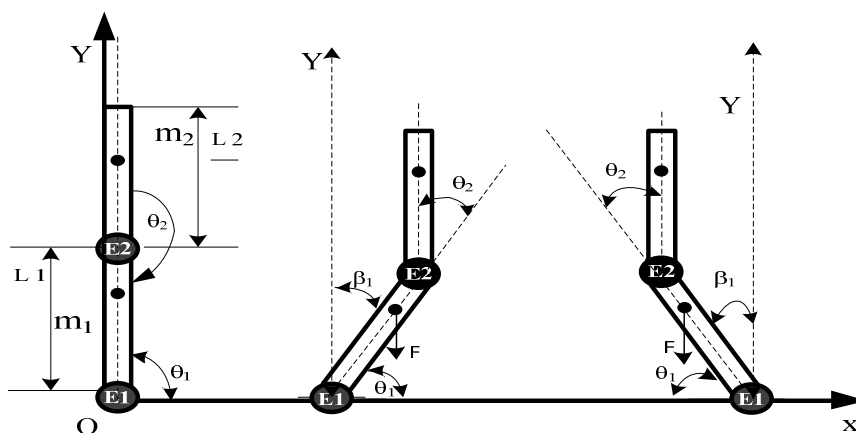
Hình 6.27 Sơ đồ điều khiển hệ Pendubot bằng phương pháp bù mômen xoay

**Giải thích:**

- $k_1$  : Hệ số bù phi tuyến
- $m$  : Khối lượng cơ cấu tạo mômen xoay.
- $P$  : Khâu P
- $I^*$  : Khâu I sau khi đóng băng tích phân
- $E_2^*$  : Giá trị Encoder 2 sau khi trượt hóa giá trị đặt
- $D$  : Giá trị đặt
- DC: Động cơ DC

### 6.3.2 Nhận dạng và điều khiển Hệ pendubot bằng phương pháp bù mômen xoay phi tuyến

Từ mô hình toán học của hệ Pendubot chương 3, nếu xét hệ vật Pendubot đơn giản như hình sau :



Hình 6.28 biểu diễn góc lệch của Link 1 so với phương OY là  $\beta$ .

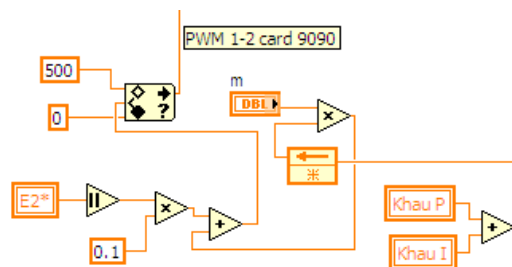
Ta gọi :

- Góc lệch của Link 1 so với phương OY là  $\beta$ .
- Mô-men xoay :  $M_x = F.d$  (N/m).
- F là trọng lực của hệ vật tạo ra mô-men xoay. Với  $F=m.g$
- d là cánh tay đòn . Với  $d = L_1 \cdot \cos \theta_1 = L_1 \cdot \cos(90^\circ - \beta)$

Bài toán bù mô-men xoay ở đây không tìm ra giá trị Mô-men xoay cụ thể của hệ vật mà là đi tính tỷ lệ để xác định hệ số bù, mà hệ số bù là một đại lượng phi tuyến do  $\cos\theta_1$  là một đại lượng phi tuyến.

Vậy để giải quyết bài toán này học viên đề xuất bù thêm thành phần m là khối lượng cơ cấu tạo mô-men xoay. Ta có sơ đồ thiết lập bù thành phần m như hình 6.27.

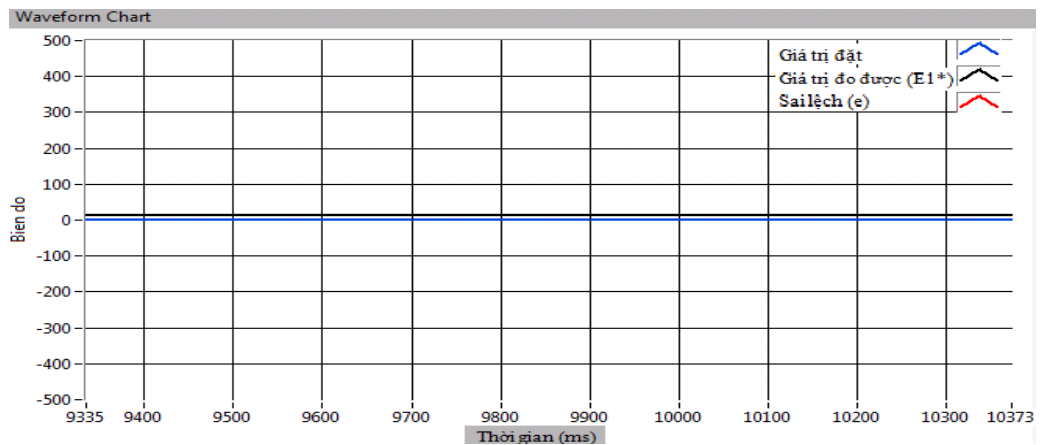
### Khối Giải thuật trình bày phương pháp điều khiển Hệ pendubot bằng phương pháp bù mômen xoay phi tuyến



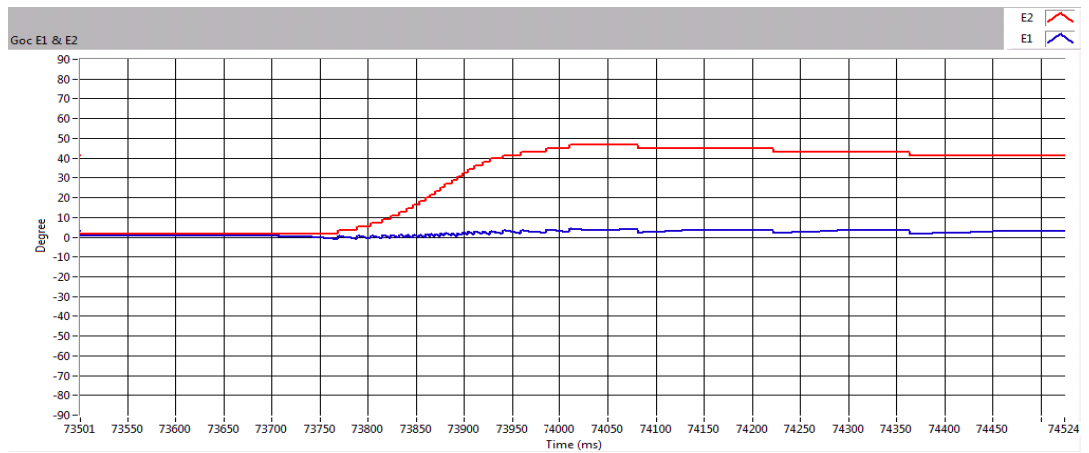
Hình 6.29 Giải thuật trình bày phương pháp bù mômen xoay phi tuyến

**Phương trình ta có như sau:**  $U(t) = ((P + I^*)Xm) + k_1.E_2^*$  (Vol)

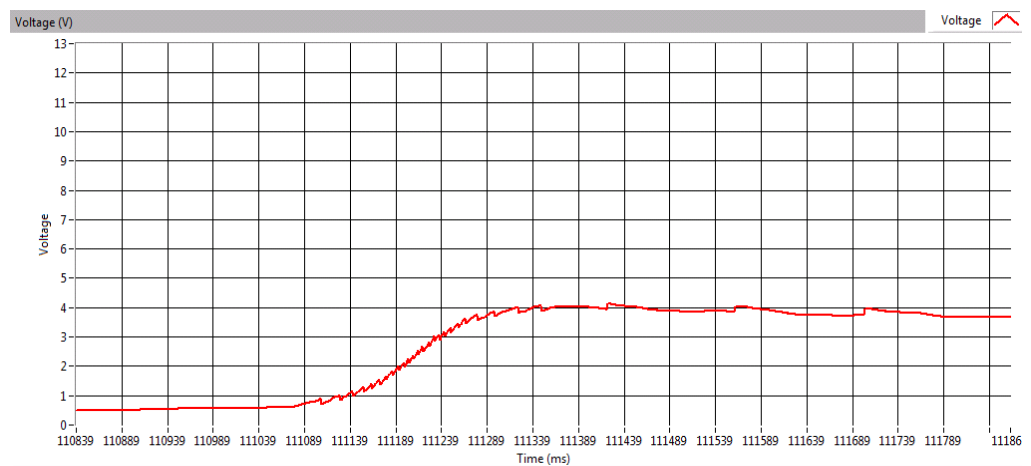
**Các đáp ứng ngõ ra :**



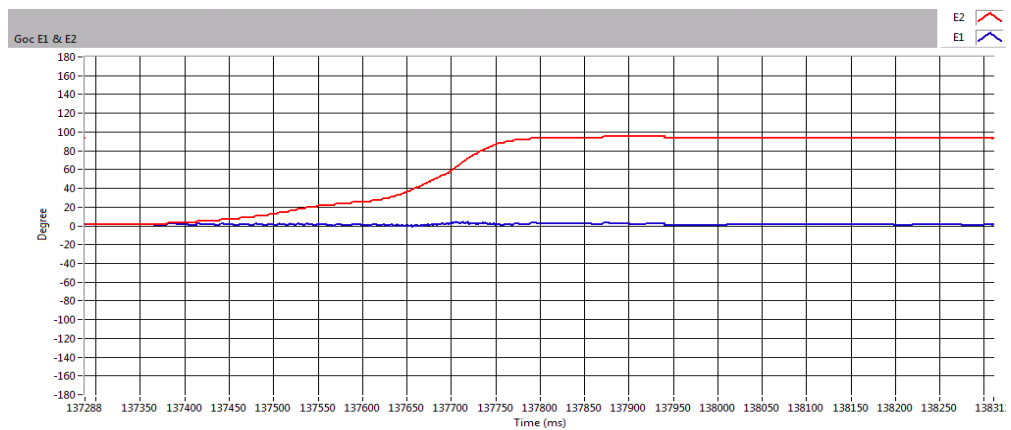
Hình 6.30 Đáp ứng có Bù Momen xoay,  $K_i$  hiệu chỉnh nhỏ



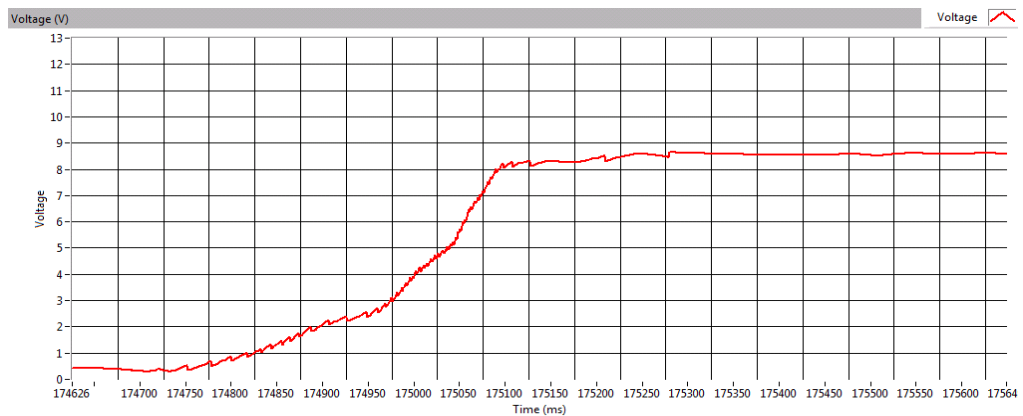
Hình 6.31 đáp ứng góc  $\theta_1 \cong 45^\circ$ ,  $\theta_2 \cong 0^\circ$



Hình 6.32 Kết quả đáp ứng điện áp ngõ ra góc  $\theta_1 \cong 45^\circ$



Hình 6.33 Đáp ứng góc  $\theta_1 \cong 90^\circ$ ,  $\theta_2 \cong 0^\circ$



Hình 6.34 Kết quả đáp ứng điện áp ngõ ra góc  $\theta_1 \cong 90^\circ$

## 6.4 Kết luận

Qua phần thực nghiệm trên hệ thống thực hệ Pendubot học viên nhận thấy :

- Bộ điều khiển PI điều khiển hệ Pendubot ổn định với góc lệch của Link 1 so với trục Oy tương đối nhỏ trong thời gian ngắn. Nhưng do hệ thống chỉ điều khiển hồi tiếp góc  $\theta_1$ ,  $\theta_2$  theo giá trị đặt Link 2 luôn thẳng đứng theo trục OX nên khi Link 2 được thẳng đứng ổn định thì sau một thời gian Link 2 có xu hướng đổ xuống do đó dẫn đến hệ vật mất ổn định.

- Để khắc phục hệ quả trên tác giả đã Nhận dạng và điều khiển Hệ pendubot bằng phương pháp bù mômen xoay phi tuyến, khi tác động một lực nhất định lên



Link 2 theo một phương nhất định , Link 1 sẽ tự điều chỉnh theo một góc nhất định với mục đích giữ cho Link 2 luôn thẳng đứng ổn định theo trục OX cho kết quả tốt sai số nhỏ. Tín hiệu ngõ ra của PI luôn theo tín hiệu góc ngõ ra của hệ Pedubot theo thời gian thực.

## CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 7.1 Kết quả đạt được

Trong luận văn này học viên đã hoàn thành những công việc sau:

- Nghiên cứu xây dựng mô hình toán hệ thống Pendubot.
- Hiểu được nguyên lý cân bằng của hệ thống
- Module mạch giao tiếp và thu thập dữ liệu với máy tính PC.
- Mạch công suất điều khiển motor.
- Nghiên cứu phương pháp điều khiển toàn phương tuyến tính LQR (Linear Quadratic Regulation), lập trình bằng ngôn ngữ Matlab để mô phỏng và điều khiển hệ thống Pendubot.
- Nghiên cứu phương pháp điều khiển PID lập trình bằng ngôn ngữ Matlab để mô phỏng và điều khiển hệ thống Pendubot.
- Lập trình ngôn ngữ LabVIEW để xử lý tín hiệu Output, Input, PWM, Encoder. Điều khiển hệ thống từ vị trí cân bằng thẳng đứng sang vị trí cân bằng mới với Link 2 luôn đứng thẳng
- Nghiên cứu thiết kế và thi công phần cơ khí, và phần điện tử cho mô hình Pendubot

### 7.2 Hạn chế

Trong luận văn này học viên đã thực hiện được nhiệm vụ đặt ra, tuy nhiên hệ thống chưa ổn định với thiết kế bộ điều khiển LQR và PID vì thời gian có hạn.

### 7.3 Hướng phát triển của đề tài

- Tiếp tục xây dựng bộ swing\_up cho hệ thống .
- Xây dựng bộ điều khiển swing\_up và cân bằng ở vị trí Top.
- Điều khiển swing\_up và giữ cân bằng cho n bậc tự do.
- Có thể dùng nhiều giải thuật khác nhau để điều khiển hệ thống như điều khiển mờ, fuzzy logic, Neural, ...

#### 7.4 Kết luận:

Mục đích nghiên cứu điều khiển hệ Pendubot trên nền labview, để nhận dạng và điều khiển hệ thống con lắc ngược. Để nhận dạng hệ Pendubot, trước tiên phải thiết lập mô hình của hệ Pendubot dựa theo phương trình toán học. Do hệ Pendubot là hệ thống vòng hở không ổn định nên không thể sử dụng phương pháp nhận dạng vòng hở, vì vậy ta phải sử dụng phương pháp nhận dạng vòng kín mô hình hệ thống với bộ điều khiển PID và thuật toán bù mô-men xoay phi tuyến để ổn định cân bằng Link 2 thẳng đứng theo trục OX. Dựa vào các dữ liệu vào - ra của mô hình, ta tiến hành nhận dạng hệ thống tuyến tính và phi tuyến. Trong phương pháp nhận dạng tuyến tính PI trên nền Labview nhận thấy kết quả đáp ứng tín hiệu ngõ ra hệ thống và ngõ ra của các mô hình có sai số không đáng kể.

Tương tự, phương pháp Bù mô-men xoay phi tuyến, tác giả cũng sử dụng các dữ liệu vào-ra từ mô hình hệ thống với bộ điều khiển PID, khi hệ Pendubot hoạt động tác động một lực nhất định lên Link 2 theo một phương nhất định, Link 2 luôn bám theo phương thẳng đứng ổn định theo trục OX khi Link 1 điều chỉnh theo một góc nhất định ( $-90^\circ$ ,  $+90^\circ$ ).

Từ các kết quả thực nghiệm và khảo sát trên mô hình thật hệ Pendubot, kết quả nhận được điều khiển hệ Pendubot trên nền labview bằng các phương pháp trượt hóa giá trị đặt, và đóng băng tích phân khâu PI, cho thấy tín hiệu sai lệch tương đối nhỏ, nhưng hệ thống không ổn định trong thời gian dài. Nhưng trong quá trình thực nghiệm bù mô-men xoay phi tuyến, cho thấy tín hiệu sai lệch tương đối nhỏ, hệ thống ổn định trong thời gian dài khi tác động một ngoại lực nhỏ bất kỳ lên Link 2 làm thay đổi góc quay  $\theta_1$  thì Link 2 vẫn bám theo trục OY và giữ ổn định hệ thống trong thời gian dài.

## TÀI LIỆU THAM KHẢO

- [1] Bennett, Stuart (1993). *A history of control engineering, 1930-1955*. IET. tr. p. 48. ISBN 9-780863412998
- [2] Araki, M.. “*PID Control*”.
- [3] Ang, K.H., Chong, G.C.Y., and Li, Y. (2005). PID control system analysis, design, and technology, IEEE Trans Control Systems Tech, 13(4), pp.559-576. <http://eprints.gla.ac.uk/3817/>
- [4] Tyaki, M.A.a.B., *Design of Fuzzy logic controller for nonlinear model of inverted pendulum - cart system*. December 17-19,2008.
- [5] Khanesar, M.A., *Sliding Mode Control of Rotary Inverted Pendulum*. Tehran, Iran, July 27-29,2007.
- [6] Johansson, R., *System modelling and identification*. Prentice Hall.
- [7] M.Spong and D.Block, “*The pendubot: A mechatronic system for control research and education*.” Proceedings of the 34th IEEE Conference on Decision and Control, pp. 1 2, 1995.
- [8] D.J. Block. “*Mechanical design and control of the pendubot*”. Degree of Master Sciences in General Engineering Thesis. University of Illinois. 1991.
- [9] M.W. Spong. “*The control of underactuated mechanical systems*”. Plenary lecture at the First International Conference on Mechatronics, Mexico City, January 26-29, 1994.
- [10] Li, J. H., Wu, M. F., and Lin, W. C., “*Linear Quadratic Regulation of the Dsp-Based Pendubot*,” Proceedings of the 2007 Intelligent Systems Conference on Engineering Applications, Tainan, Taiwan (2007).
- [11] [www.ni.com](http://www.ni.com) tìm hiểu về lập trình và lịch sử Labview.
- [12] [www.labview.hocdelam.org](http://www.labview.hocdelam.org)
- [13] Nguyễn Đức Thành.(2005) .Matlab và ứng dụng trong điều khiển. Nhà xuất bản Đại học quốc gia TP.HCM.

[14] Nguyễn Thị Phương Hà.(2008).”*Lý thuyết điều khiển hiện đại*”,Tài liệu lưu hành nội bộ trường Đại học Bách Khoa TP.HCM.

[15] Huỳnh Thái Hoàng “*Lý Thuyết Điều Khiển Thông Minh*”, Nhà Xuất Bản Đại Học Quốc Gia 2006.