

**ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN**

**ĐỒ ÁN TỐT NGHIỆP
CAPSTONE PROJECT**

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN VÀ TỰ ĐỘNG HÓA

ĐỀ TÀI:

**NGHIÊN CỨU THUẬT TOÁN ĐIỀU KHIỂN
TAY MÁY ROBOT 3 BẬC TỰ DO**

Người hướng dẫn: **TS. TRƯƠNG THỊ BÍCH THANH**

Sinh viên thực hiện:

1. LƯU QUANG THÁI – MSSV: 105200383

LỚP: 20TDHCLC1

2. NGUYỄN ĐẮC THẠNH – MSSV: 105200430

LỚP: 20TDHCLC2

Đà Nẵng, 6/2025

TÓM TẮT

Tên đề tài: “Nghiên cứu thuật toán điều khiển tay máy Robot 3 bậc tự do”

Sinh viên thực hiện 1: Lưu Quang Thái

Số thẻ sinh viên: 105200383

Lớp: 20TDHCLC1

Sinh viên thực hiện 2: Nguyễn Đắc Thanh

Số thẻ sinh viên: 150200430

Lớp: 20TDHCLC2

Phần mở đầu:

Giới thiệu bối cảnh đề tài, lí do chọn đề tài cũng như mục tiêu đề ra.

Chương 1: Tổng quan đề tài

Nội dung chương này giới thiệu sơ lược quá trình hình thành và phát triển của Robot công nghiệp, vai trò và ứng dụng của Robot công nghiệp trong các quy trình tự động trong sản xuất và nghiên cứu hiện nay. Ngoài ra, phần này còn giới thiệu các khái niệm, định nghĩa và cấu trúc cơ bản của tay máy Robot công nghiệp.

Chương 2: Xây dựng mô hình động lực học Robot 3DOF

Nội dung chương này trình bày cơ sở lý thuyết về tay máy (động học thuận, động học ngược, động lực học). Thiết kế mô hình tay máy trên phần mềm Solidworks và Matlab Simulink.

Chương 3: Các phương pháp điều khiển cánh tay Robot

Nội dung chương này trình bày phương pháp thiết kế các bộ điều khiển bao gồm bộ điều khiển PID và bộ điều khiển học lặp ILC để ứng dụng điều khiển tay máy hoạt động một cách chính xác theo quỹ đạo cho trước.

Chương 4: Mô phỏng cánh tay Robot và đánh giá kết quả

Nội dung chương này trình bày phần mô phỏng và xuất ra kết quả trên phần mềm Matlab Simulink.

Phần kết luận

Từ kết quả mô phỏng kết hợp với các chỉ tiêu chất lượng đưa ra kết quả phân tích so sánh giữa các phương pháp điều khiển. Đánh giá lại toàn bộ quá trình làm đồ án, đưa ra những kết quả đạt được cũng như hướng phát triển trong tương lai.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

TT	Họ và tên	MSSV	Lớp	Ngành
1	Lưu Quang Thái	105200383	20TDHCLC1	Kỹ thuật điều khiển và Tự động hóa
2	Nguyễn Đắc Thạnh	105200430	20TDHCLC2	Kỹ thuật điều khiển và Tự động hóa

1. Tên đề tài đồ án:

“Nghiên cứu thuật toán điều khiển tay máy Robot 3 bậc tự do”

2. Đề tài thuộc diện: Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện.

3. Các số liệu và dữ liệu ban đầu:

4. Nội dung các phần thuyết minh và tính toán:

a) Phần chung:

TT	Họ tên sinh viên	Nội dung
1	Lưu Quang Thái	- Viết báo cáo và thuyết minh. - Tìm hiểu các phương pháp điều khiển tay máy Robot 3 bậc tự do (PID và ILC).
2	Nguyễn Đắc Thạnh	- Đánh giá, nhận xét kết quả so sánh giữa hai phương pháp điều khiển PID và PID kết hợp ILC.

b) Phần riêng:

TT	Họ tên sinh viên	Nội dung
1	Lưu Quang Thái	- Phân tích động học thuận tay máy Robot 3 bậc tự do. - Phân tích động học nghịch tay máy Robot 3 bậc tự do. - Phân tích và xác định phương trình động lực học tay máy Robot 3 bậc tự do. - Tìm hiểu phương pháp điều khiển tay máy Robot 3 bậc tự do sử dụng bộ điều khiển PID. - Mô phỏng điều khiển tay máy Robot 3 bậc tự do sử dụng bộ điều khiển PID. - Mô phỏng điều khiển tay máy Robot 3 bậc tự do sử dụng bộ điều khiển PID kết hợp ILC.

2	Nguyễn Đắc Thanh	<ul style="list-style-type: none"> - Tìm hiểu tổng quan, cấu trúc về tay máy Robot 3 bậc tự do. - Lựa chọn thông số tay máy Robot 3 bậc tự do theo tay máy Robot có trong thực tế. - Nhận dạng mô hình động cơ. - Thiết kế tay máy Robot 3 bậc tự do trên SolidWork. - Tìm hiểu phương pháp điều khiển tay máy Robot 3 bậc tự do sử dụng bộ điều khiển ILC. - Lựa chọn các tiêu chí đánh giá chất lượng hệ thống.
---	------------------	---

5. Các bản vẽ, đồ thị (ghi rõ các loại và kích thước bản vẽ):

6. Họ tên hướng dẫn: TS. Trương Thị Bích Thanh

Nội dung hướng dẫn:

- Hướng dẫn tư vấn hướng đi cho đề tài.
- Đề xuất nhận dạng động cơ, phương pháp điều khiển ILC cho việc điều khiển tay máy Robot.
- Đánh giá việc mô phỏng của đề tài.
- Hướng dẫn làm báo cáo thuyết minh cho đề tài.
- Theo dõi tiến độ dự án hàng tuần.

7. Ngày giao nhiệm vụ đồ án: 03/03/2025

8. Ngày hoàn thành đồ án: 02/06/2025

Trưởng bộ môn

GVHD

TS. Giáp Quang Huy

TS. Trương Thị Bích Thanh

PHIẾU KIỂM SOÁT TIẾN ĐỘ LÀM ĐỒ ÁN TỐT NGHIỆP

(Phiếu dành cho người hướng dẫn/sinh viên)

Họ tên sinh viên 1: Lưu Quang Thái

Số thẻ SV : 105200383

Họ tên sinh viên 2: Nguyễn Đức Thạnh

Số thẻ SV : 105200430

Tên đề tài ĐATN: “Nghiên cứu thuật toán điều khiển tay máy Robot 3 bậc tự do”

Họ tên người HD: Trương Thị Bích Thanh

Đơn vị: Khoa Điện

Tuần	Ngày	Khối lượng		GVHD ký tên
		đã thực hiện (%)	tiếp tục thực hiện (%)	
1	03/03/2025	Nhận đề tài.	Trao đổi đề tài với người hướng dẫn.	
2	06/03/2025	Lựa chọn lĩnh vực loại tay máy Robot muốn thiết kế và điều khiển.	Thiết kế và lựa chọn phương pháp điều khiển cho tay máy Robot.	
3	10/03/2025	Thiết kế tay máy Robot 3DOF trên SolidWork và tìm hiểu các phương pháp điều khiển.	Phân tích động học thuận, nghịch cho tay máy Robot 3DOF.	
4	13/03/2025	Duyệt lần 1: Đánh giá khối lượng hoàn thành 15% : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
5	17/03/2025	Phân tích động thuận, động học nghịch cho tay máy Robot 3DOF.	Xây dựng phương trình động lực học cho tay máy Robot 3DOF.	
6	24/03/2025	Xây dựng phương trình động lực học cho tay máy Robot 3DOF.	Nhận dạng mô hình động cơ. Tìm hiểu bộ điều khiển PID và bộ điều khiển ILC.	
7	31/03/2025	Nhận dạng mô hình động cơ. Tìm hiểu các phương pháp tinh chỉnh bộ điều khiển PID.	Áp dụng được bộ điều khiển PID cho việc mô phỏng điều khiển tay máy Robot 3DOF.	
8	07/04/2025	Áp dụng được bộ điều khiển PID cho việc mô phỏng điều khiển tay máy Robot 3DOF.	Tìm hiểu bộ điều khiển ILC và áp dụng điều khiển tay máy Robot sử dụng bộ điều khiển PID kết hợp ILC.	
9	11/04/2025	Duyệt lần 2: Đánh giá khối lượng hoàn thành 65% : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
10	14/04/2025	Áp dụng được bộ điều khiển PID kết hợp ILC kiểu D cho việc mô phỏng điều khiển tay máy Robot 3DOF.	Áp dụng được bộ điều khiển PID kết hợp ILC kiểu P và kiểu PD cho việc mô phỏng điều khiển tay máy Robot 3DOF.	

11	21/04/2025	Áp dụng được bộ điều khiển PID kết hợp ILC kiểu P cho việc mô phỏng điều khiển tay máy Robot 3DOF.	Áp dụng được bộ điều khiển PID kết hợp ILC kiểu PD cho việc mô phỏng điều khiển tay máy Robot 3DOF.	
12	28/04/2025	Áp dụng được bộ điều khiển PID kết hợp ILC kiểu PD cho việc mô phỏng điều khiển tay máy Robot 3DOF.	Đưa ra các chỉ tiêu đánh giá chất lượng hệ thống để tiến hành so sánh các phương pháp điều khiển	
13	05/05/2025	Đưa ra 3 chỉ tiêu đánh giá chất lượng hệ thống.	So sánh phương pháp điều khiển sử dụng bộ PID với sử dụng bộ PID kết hợp ILC trong việc mô phỏng điều khiển tay máy Robot 3DOF.	
14	12/05/2025	Duyệt lần 3: Đánh giá khối lượng hoàn thành <u>95</u> % : Được tiếp tục làm ĐATN <input type="checkbox"/> Không tiếp tục thực hiện ĐATN <input type="checkbox"/>		
15	19/05/2025	Lập các bảng so sánh hai phương pháp điều khiển sử dụng bộ PID với sử dụng bộ PID kết hợp ILC và đánh giá kết quả.	Tổng hợp tài liệu, hoàn thiện viết báo cáo thuyết minh.	
16	26/05/2025	Viết báo cáo thuyết minh.		
17	02/06/2025	Chuẩn bị cho thuyết trình đồ án.		

LỜI NÓI ĐẦU VÀ CẢM ƠN

Trong những năm gần đây, sự phát triển mạnh mẽ của công nghệ tự động hóa và Robot đã tạo ra những bước đột phá đáng kể trong nhiều lĩnh vực, từ sản xuất công nghiệp đến dịch vụ y tế, giáo dục và cả trong đời sống hàng ngày. Tự động hóa, thông qua việc ứng dụng các hệ thống Robot, không chỉ giúp nâng cao hiệu quả công việc mà còn giảm chi phí, tăng năng suất lao động và đảm bảo chất lượng sản phẩm.

Với sự ra đời và ứng dụng của các loại Robot công nghiệp, các hệ thống tự động hóa đã và đang thay đổi cách thức hoạt động trong các ngành công nghiệp, đặc biệt là trong các công đoạn yêu cầu độ chính xác cao. Những Robot này không chỉ giúp giảm thiểu sự can thiệp của con người mà còn nâng cao năng suất lao động, góp phần vào sự phát triển bền vững của doanh nghiệp.

Qua thời gian học tập trên ghế nhà trường và tổng hợp từ những kiến thức học được liên quan đến lĩnh vực Robot thì chúng em đã chọn đề tài: “**Nghiên cứu thuật toán điều khiển tay máy Robot 3 bậc tự do**”. Nghiên cứu đề tài nhằm củng cố kiến thức và nhận ra sự liên hệ mật thiết giữa kiến thức đã được học ở trường và kiến thức thực tế bên ngoài, từ đó rút ra được những kinh nghiệm và kỹ năng cần thiết cho công việc sau này.

Nhóm chúng em xin cảm ơn đến GVHD TS.Trương Thị Bích Thanh đã tận tình hướng dẫn và giúp đỡ nhóm trong quá trình thực hiện đề án tốt nghiệp này. Đồng thời, nhóm cũng xin cảm ơn các thầy cô trong khoa Điện đã giảng dạy, cung cấp kiến thức cho chúng em trong những năm tháng học trên giảng đường.

Vì kiến thức còn hạn chế, nên không tránh khỏi thiếu sót trong quá trình thực hiện đề án. Nhóm chúng em rất mong nhận được sự đóng góp ý kiến của quý thầy, cô để báo cáo tốt nghiệp đạt được kết quả tốt nhất.

Chúng em xin chân thành cảm ơn.

LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT

Đồ án này là công trình nghiên cứu của nhóm dưới sự hỗ trợ từ giảng viên hướng dẫn TS. Trương Thị Bích Thanh. Tất cả các đoạn văn, ý kiến, quan điểm của người khác được sử dụng trong đồ án này đều được dẫn nguồn một cách cụ thể. Các thành viên trong nhóm đã đọc rõ quy định của nhà trường và hiểu rõ đạo văn là một lỗi nghiêm trọng trong học tập. Chúng tôi xin chịu hoàn toàn trách nhiệm nếu có bất kỳ sự sao chép, gian dối kết quả nào trong sản phẩm đồ án này.

Sinh viên thực hiện

Sinh viên thực hiện

Lưu Quang Thái

Nguyễn Đắc Thịnh

MỤC LỤC

LỜI NÓI ĐẦU VÀ CẢM ƠN	i
LỜI CAM ĐOAN LIÊM CHÍNH HỌC THUẬT	ii
MỤC LỤC	iii
DANH SÁCH CÁC BẢNG, HÌNH VẼ	vi
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	xiv
	Trang
MỞ ĐẦU	1
Chương 1: TỔNG QUAN ĐỀ TÀI.....	3
1.2. Vai trò và ứng dụng của Robot trong sản xuất	4
1.3. Các khái niệm và định nghĩa về robot công nghiệp.....	7
1.3.1. Định nghĩa về Robot công nghiệp	7
1.3.2. Các khái niệm	7
1.4. Cấu trúc cơ bản của Robot công nghiệp.....	10
1.4.1. Các thành phần chính của Robot công nghiệp	10
1.4.2. Cánh tay máy Robot	11
1.5. Phân loại Robot công nghiệp	12
Chương 2: XÂY DỰNG MÔ HÌNH ĐỘNG LỰC HỌC ROBOT 3DOF.....	14
2.1. Cấu trúc và thông số cánh tay robot 3 bậc tự do	14
2.1.1. Cấu trúc cánh tay robot 3 bậc tự do.....	14
2.1.2. Thông số cánh tay robot 3 bậc tự do	15
2.2. Động học thuận cánh tay Robot 3 bậc tự do	18
2.2.1. Gắn hệ tọa độ cho từng khâu	18
2.2.2. Lập bảng thông số D-H (Denavit-Hartenberg).....	19
2.2.3. Xác định các ma trận suy rộng A_i	19
2.2.4. Xác định ma trận vecto cuối T theo các ma trận A_i	22
2.3. Động học nghịch cánh tay Robot 3 bậc tự do.....	23
2.4. Xác định mô hình động lực học của cánh tay Robot 3 bậc tự do	26
2.5. Thiết kế mô hình Robot 3DOF trên Solidworks	33
2.6. Nhận dạng mô hình động cơ	36
Chương 3: CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN CÁN H TAY ROBOT	41
3.1. Tổng quan về điều khiển Robot.....	41
3.2. Phương pháp điều khiển cánh tay Robot sử dụng bộ điều khiển PID	42
3.2.1. Tổng quan về bộ điều khiển PID.....	42

3.2.2. Các phương pháp tính chọn tham số bộ điều khiển PID	43
3.2.3. Lựa chọn thông số bộ điều khiển PID cho cánh tay Robot 3 bậc tự do	46
3.3. Phương pháp điều khiển cánh tay Robot sử dụng bộ điều khiển ILC	48
3.3.1. Tổng quan về bộ điều khiển ILC	48
3.3.2. Quy trình học lặp của bộ điều khiển ILC	50
3.3.3. Thuật toán điều khiển học lặp ILC cho cánh tay Robot 3 bậc tự do	52
3.3.4. Thiết kế bộ điều khiển học lặp ILC cho cánh tay Robot 3 bậc tự do	54
Chương 4: MÔ PHỎNG ĐIỀU KHIỂN CÁNH TAY ROBOT VÀ ĐÁNH GIÁ KẾT QUẢ	62
4.1. Sơ đồ khối điều khiển và các chỉ tiêu đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do	62
4.1.1. Sơ đồ khối điều khiển mô hình SolidWork và mô hình Động lực học sử dụng bộ điều khiển PID	62
4.1.2. Sơ đồ khối điều khiển mô hình SolidWork và mô hình Động lực học sử dụng bộ điều khiển ILC	63
4.1.3. Các chỉ tiêu đánh giá chất lượng của hệ thống.....	64
4.2. Kết quả mô phỏng hệ thống điều khiển cánh tay robot 3 DOF sử dụng bộ điều khiển PID.....	66
4.2.2. Mô phỏng điều khiển mô hình SolidWork và mô hình Động lực học khi chưa sử dụng bộ điều khiển	67
4.2.3. Sơ đồ mô phỏng điều khiển mô hình SolidWork và mô hình Động lực học khi sử dụng chung 1 thông số bộ điều khiển PID	68
4.2.4. Kết quả mô phỏng điều khiển mô hình SolidWork và mô hình Động lực học khi sử dụng chung 1 thông số bộ điều khiển PID.....	68
4.2.5. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID	73
4.2.6. Kết quả mô phỏng điều khiển mô hình xây dựng trên SolidWork khi sử dụng thông số bộ điều khiển PID	80
4.3. Kết quả mô phỏng hệ thống điều khiển cánh tay robot 3 DOF sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC	86
4.3.1. Sơ đồ mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC.....	86
4.3.2. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D.....	87
4.3.3. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P	93

4.3.4. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD	99
4.3.5. Kết quả mô phỏng điều khiển mô hình SolidWork khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D	105
4.3.6. Kết quả mô phỏng điều khiển mô hình SolidWork khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P	111
4.3.7. Kết quả mô phỏng điều khiển mô hình SolidWork khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD	118
4.4. So sánh, nhận xét kết quả mô phỏng giữa 2 phương pháp điều khiển sử dụng bộ điều khiển PID với bộ điều khiển PID kết hợp ILC.....	124
4.4.1. So sánh, nhận xét kết quả mô phỏng mô hình động lực học giữa 2 phương pháp điều khiển	124
4.4.2. So sánh, nhận xét kết quả mô phỏng mô hình SolidWork giữa 2 phương pháp điều khiển	127
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	130
TÀI LIỆU THAM KHẢO.....	132
PHỤ LỤC	1

DANH SÁCH CÁC BẢNG, HÌNH VẼ

Bảng 2.1: Bảng thông số D-H cho cánh tay Robot 3 bậc tự do	19
Bảng 3.1: Bảng phương pháp Ziegler-Nichols.....	44
Bảng 4.1: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình động lực học khi in 3D	79
Bảng 4.2: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình động lực học khi vẽ 2D	79
Bảng 4.3: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình SolidWork khi in 3D.....	85
Bảng 4.4: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình SolidWork khi vẽ 2D.....	86
Bảng 4.5: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình động lực học khi in 3D	92
Bảng 4.6: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình động lực học khi vẽ 2D	92
Bảng 4.7: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình động lực học khi in 3D.....	98
Bảng 4.8: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình động lực học khi vẽ 2D	98
Bảng 4.9: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình động lực học khi in 3D.....	104
Bảng 4.10: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình động lực học khi vẽ 2D.....	104
Bảng 4.11: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình SolidWork khi in 3D.....	111
Bảng 4.12: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình SolidWork khi vẽ 2D.....	111
Bảng 4.13: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình SolidWork khi in 3D	117
Bảng 4.14: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình SolidWork khi vẽ 2D.....	117
Bảng 4.15: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình SolidWork khi in 3D	124

Bảng 4.16: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình SolidWork khi vẽ 2D.....	124
Bảng 4.17: Bảng so sánh kết quả mô phỏng mô hình động lực học khi in 3D giữa các phương pháp điều khiển khác nhau.....	125
Bảng 4.18: Bảng so sánh kết quả mô phỏng mô hình động lực học khi vẽ 2D giữa các phương pháp điều khiển khác nhau.....	125
Bảng 4.19: Bảng so sánh kết quả mô phỏng mô hình SolidWork khi in 3D giữa các phương pháp điều khiển khác nhau.....	127
Bảng 4.20: Bảng so sánh kết quả mô phỏng mô hình SolidWork khi vẽ 2D giữa các phương pháp điều khiển khác nhau.....	128
Hình 1.1: Robot trong vở kịch Rossum's Universal Robots	3
Hình 1.2: Robot trong dây chuyền sản xuất ô tô.....	5
Hình 1.3: Robot trong lĩnh vực nông nghiệp.....	6
Hình 1.4: Robot trong lĩnh vực thăm dò vũ trụ.....	6
Hình 1.5: Các tọa độ suy rộng của Robot.....	9
Hình 1.6: Quy tắc bàn tay phải.....	9
Hình 1.7: Trường công tác của Robot	10
Hình 1.8: Các thành phần chính của hệ thống Robot.....	10
Hình 2.1: Cơ cấu Robot 3 bậc tự do (RRR)	14
Hình 2.2: Cánh tay Robot 3DOF có tỷ lệ các khâu 1:1:1 dùng trong giáo dục và đào tạo	16
Hình 2.3: Cánh tay Robot 3DOF có tỷ lệ các khâu là 1:1.5:1 dùng trong ứng dụng gấp vật	16
Hình 2.4: Cánh tay Robot 3DOF có tỷ lệ các khâu là 1:1.5:2 dùng trong vẽ 2D.....	17
Hình 2.5: Mô hình cánh tay Robot 3 bậc tự do ZEKEEP mã số ZKBOT-3042B.....	17
Hình 2.6: Cánh tay Robot 3 bậc tự do trong hệ trục tọa độ.....	19
Hình 2.7: Cánh tay Robot 3 bậc tự do trên mặt phẳng 2D	27
Hình 2.8: Cấu trúc để cánh tay Robot 3DOF trên SolidWork.....	33
Hình 2.9: Cấu hình khâu 1 cánh tay Robot 3DOF trên SolidWork.....	34
Hình 2.10: Cấu trúc khâu 2 cánh tay Robot 3DOF trên SolidWork.....	34
Hình 2.11: Cấu trúc khâu 3 cánh tay Robot 3DOF trên SolidWork.....	35
Hình 2.12: Cấu trúc cánh tay Robot 3 bậc tự do trên SolidWork.....	35
Hình 2.13: Thông báo chuyển mô hình từ SolidWork sang Matlab thành công.....	36
Hình 2.14: Mô hình cánh tay Robot trên Solid khi được chuyển sang Matlab Simulink	36

Hình 2.15: Mô hình lấy dữ liệu đầu vào và đầu ra từ mô hình SolidWork	38
Hình 2.16: Giao diện nhập dữ liệu vào Matlab Identification Toolbox	38
Hình 2.17: Giao diện nhận dạng hàm truyền đạt Matlab Identification Toolbox	39
Hình 2.18: Giao diện kết quả hàm truyền đạt nhận dạng được từ Matlab Identification Toolbox	39
Hình 3.1: Sơ đồ khối bộ điều khiển PID	42
Hình 3.2: Sơ đồ khối phương pháp Relay - Feedback	45
Hình 3.3: Hướng dẫn sử dụng PID Tuner để lựa chọn thông số PID.....	46
Hình 3.4: Hướng dẫn sử dụng PID Tuner để lựa chọn thông số PID.....	47
Hình 3.5: Hướng dẫn sử dụng PID Tuner để lựa chọn thông số PID.....	47
Hình 3.6: Đồ thị kết quả sau khi sử dụng PID Tuner để chọn thông số PID	48
Hình 3.7: Sơ đồ hệ điều khiển học lặp với bộ điều khiển ILC	48
Hình 3.8: Sơ đồ khối các bước học và chỉnh định trong điều khiển học lặp.....	50
Hình 3.9: Sơ đồ mô tả một hệ thống kết hợp giữa bộ điều khiển truyền thống và bộ điều khiển học lặp ILC	53
Hình 4.1: Sơ đồ khối mô hình SolidWork sử dụng bộ điều khiển PID	62
Hình 4.2: Sơ đồ khối mô hình động lực học sử dụng bộ điều khiển PID	62
Hình 4.3: Sơ đồ khối mô hình SolidWork sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC	63
Hình 4.4: Sơ đồ khối mô hình Động lực học sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC	63
Hình 4.5: Đồ thị hiện tượng độ quá điều chỉnh trong đáp ứng quá độ của hệ thống	64
Hình 4.6: Đồ thị biểu diễn thời gian xác lập của hệ thống.....	65
Hình 4.7: Sơ đồ mô phỏng mô hình động lực học cánh tay Robot	66
Hình 4.8: Sơ đồ mô phỏng cánh tay Robot SolidWork trên Matlab Simulink.....	66
Hình 4.9: Sơ đồ mô phỏng điều khiển mô hình SolidWork và Động lực học khi chưa có bộ điều khiển.	67
Hình 4.10: Đồ thị tọa độ trục x y z của 2 mô hình SolidWork và Động lực học khi chưa có bộ điều khiển.....	67
Hình 4.11: Sơ đồ mô phỏng điều khiển mô hình SolidWork và Động lực học sử dụng bộ điều khiển PID	68
Hình 4.12: Hình in 3D mong muốn.....	69
Hình 4.13: Đồ thị tọa độ x y z đầu ra cho hình in 3D của 2 mô hình sử dụng chung 1 bộ điều khiển PID	69
Hình 4.14: Đồ thị sai số góc quay cho hình in 3D của 2 mô hình sử dụng chung 1 bộ điều khiển PID	70

Hình 4.15: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork.....	70
Hình 4.16: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học	71
Hình 4.17: Hình vẽ 2D mong muốn	71
Hình 4.18: Đồ thị tọa độ đầu ra cho hình vẽ 2D của 2 mô hình sử dụng chung 1 bộ điều khiển PID	71
Hình 4.19: Đồ thị sai số góc quay khi vẽ 2D của 2 mô hình sử dụng chung 1 bộ điều khiển PID	72
Hình 4.20: Đồ thị kết quả vẽ 2D của mô hình SolidWork	72
Hình 4.21: Đồ thị kết quả vẽ 2D của mô hình động lực học	73
Hình 4.22: Đồ thị tọa độ x y z đầu ra cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu	74
Hình 4.23: Đồ thị tọa độ x y z đầu ra cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu	74
Hình 4.24: Đồ thị sai lệch góc quay cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu	75
Hình 4.25: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học khi có nhiễu và khi không có nhiễu.....	76
Hình 4.26: Đồ thị tọa độ x y đầu ra cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu	77
Hình 4.27: Đồ thị tọa độ x y đầu ra cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu	77
Hình 4.28: Đồ thị sai số góc quay cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu	78
Hình 4.29: Đồ thị kết quả vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu.....	78
Hình 4.30: Đồ thị tọa độ x y z đầu ra cho hình in 3D của mô hình SolidWork sử dụng bộ điều khiển PID khi không có nhiễu	80
Hình 4.31: Đồ thị tọa độ x y z đầu ra cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu	81
Hình 4.32: Đồ thị sai lệch góc quay cho hình in 3D của mô hình SolidWork sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu	82
Hình 4.33: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork khi không có nhiễu và khi có nhiễu.....	83
Hình 4.34: Đồ thị tọa độ đầu ra cho hình vẽ 2D của mô hình SolidWork sử dụng bộ điều khiển PID khi không có nhiễu	83

Hình 4.35: Đồ thị tọa độ x y đầu ra cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu	84
Hình 4.36: Đồ thị sai số góc quay khi vẽ 2D của mô hình SolidWork sử dụng bộ điều khiển PID khi có nhiễu	84
Hình 4.37: Đồ thị kết quả vẽ 2D của mô hình SolidWork khi không có nhiễu và khi có nhiễu	85
Hình 4.38: Rời rạc hóa tín hiệu điều khiển và biến tín hiệu điều khiển đầu ra về miền liên tục cho bộ điều khiển ILC	87
Hình 4.39: Sơ đồ mô phỏng điều khiển mô hình Động lực học sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC	87
Hình 4.40: Sơ đồ mô phỏng bộ điều khiển ILC kiểu D	88
Hình 4.41: Đồ thị so sánh tọa độ x y z khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học	88
Hình 4.42: Đồ thị so sánh khả năng xử lý nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu D	89
Hình 4.43: Đồ thị so sánh sai số góc quay khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học	89
Hình 4.44: Đồ thị so sánh theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu	90
Hình 4.45: Đồ thị so sánh tọa độ x y z khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học	90
Hình 4.46: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu D	91
Hình 4.47: Đồ thị so sánh sai lệch góc quay khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học	91
Hình 4.48: Đồ thị so sánh kết quả vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học	92
Hình 4.49: Sơ đồ mô phỏng bộ điều khiển ILC kiểu P	93
Hình 4.50: Đồ thị so sánh tọa độ x y z khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học	94
Hình 4.51: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu P	94
Hình 4.52: Đồ thị so sánh sai số góc quay khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học	95
Hình 4.53: Đồ thị so sánh theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu	96

Hình 4.54: Đồ thị so sánh tọa độ x y z khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học.....	96
Hình 4.55: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu P.....	97
Hình 4.56: Đồ thị so sánh sai số góc quay khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học.....	97
Hình 4.57: Đồ thị so sánh kết quả vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học	98
Hình 4.58: Sơ đồ mô phỏng bộ điều khiển PD cho mô hình SolidWork	99
Hình 4.59: Đồ thị so sánh tọa độ x y z khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học.....	100
Hình 4.60: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu PD.....	100
Hình 4.61: Đồ thị so sánh sai lệch góc quay khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học.....	101
Hình 4.62: Đồ thị so sánh theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu	102
Hình 4.63: Đồ thị so sánh tọa độ x y z khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học.....	102
Hình 4.64: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu PD.....	103
Hình 4.65: Đồ thị so sánh sai số góc quay khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học.....	103
Hình 4.66: Đồ thị so sánh kết quả vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học	104
Hình 4.67: Sơ đồ mô phỏng bộ điều khiển ILC kiểu D cho mô hình SolidWork	105
Hình 4.68: Đồ thị so sánh tọa độ x y z đầu ra cho hình in 3D của mô hình SolidWork giữa PID với PID kết hợp ILC kiểu D.....	106
Hình 4.69: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D	107
Hình 4.70: Đồ thị so sánh sai lệch góc quay q_3 khi in 3D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D	107
Hình 4.71: Đồ thị so sánh đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu	108
Hình 4.72: Đồ thị tọa độ x y z đầu ra cho hình vẽ 2D của mô hình SolidWork sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D	109

Hình 4.73: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D.....	109
Hình 4.74: Đồ thị so sánh sai lệch góc quay q_3 khi vẽ 2D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D.....	110
Hình 4.75: Đồ thị so sánh hình vẽ 2D đầu ra của mô hình SolidWork giữa bộ điều khiển PID và bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D.....	110
Hình 4.76: Sơ đồ mô phỏng bộ điều khiển ILC kiểu P cho mô hình SolidWork.....	112
Hình 4.77: Đồ thị so sánh tọa độ x y z đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P.....	113
Hình 4.78: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P.....	113
Hình 4.79: Đồ thị so sánh sai lệch góc quay q_3 khi in 3D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P.....	114
Hình 4.80: Đồ thị so sánh đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu.....	114
Hình 4.81: Đồ thị so sánh tọa độ x y đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P.....	115
Hình 4.82: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P.....	116
Hình 4.83: Đồ thị so sánh sai lệch góc quay q_3 khi vẽ 2D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P.....	116
Hình 4.84: Đồ thị so sánh hình vẽ 2D đầu ra của mô hình SolidWork giữa bộ điều khiển PID và bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P.....	117
Hình 4.85: Sơ đồ mô phỏng bộ điều khiển ILC kiểu PD cho mô hình SolidWork.....	118
Hình 4.86: Đồ thị so sánh tọa độ x y z đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD.....	119
Hình 4.87: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD.....	120
Hình 4.88: Đồ thị so sánh sai lệch góc quay q_3 khi in 3D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD.....	120
Hình 4.89: Đồ thị so sánh đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu.....	121
Hình 4.90: Đồ thị so sánh tọa độ x y đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD.....	122

Hình 4.91: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD.....	122
Hình 4.92: Đồ thị so sánh sai lệch góc quay q_3 khi vẽ 2D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD.....	123
Hình 4.93: Đồ thị so sánh hình vẽ 2D đầu ra của mô hình SolidWork giữa bộ điều khiển PID và bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD	123

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

TT	Chữ viết tắt	Chữ viết đầy đủ
1	DOF	Degrees of Freedom (Bậc tự do)
2	PID	Proportional – Integral – Derivative (Tỷ lệ – Tích phân – Vi phân)
3	ILC	Iterative Learning Control (Điều khiển học lặp)
4	R-R-R	Revolute – Revolute – Revolute (Robot có 3 khớp quay)

MỞ ĐẦU

1. Giới thiệu bối cảnh đề tài:

Trong bối cảnh cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, Robot công nghiệp ngày càng đóng vai trò then chốt trong tự động hóa sản xuất, đặc biệt trong các lĩnh vực như lắp ráp, hàn, in 3D, sơn, vận chuyển linh kiện, và thao tác chính xác trong môi trường khắc nghiệt. Một trong những cấu trúc phổ biến và nền tảng nhất trong lĩnh vực Robot là cánh tay Robot 3 bậc tự do (3 DOF - Degrees of Freedom).

Đề tài “**Nghiên cứu thuật toán điều khiển tay máy Robot 3 bậc tự do**” được xây dựng nhằm nghiên cứu về các thuật toán điều khiển áp dụng cho cánh tay Robot có cấu trúc 3 bậc tự do, một dạng Robot đơn giản nhưng được sử dụng nhiều cho các ứng dụng thực tế.

Trong cấu hình 3 bậc tự do, Robot có thể thực hiện các chuyển động trong không gian 2D hoặc 3D tùy theo thiết kế khớp quay (revolute) hoặc khớp tịnh tiến (prismatic). Việc nghiên cứu điều khiển loại Robot này đóng vai trò quan trọng trong việc xây dựng nền tảng cho các hệ thống Robot phức tạp hơn.

2. Lý do chọn đề tài:

Trong số các cấu hình Robot, cánh tay Robot 3 bậc tự do (3-DOF) là dạng nền tảng, thường được sử dụng để nghiên cứu cơ bản về động học, động lực học và các phương pháp điều khiển. Với cấu trúc đơn giản nhưng đủ phức tạp để thể hiện bản chất của nhiều vấn đề kỹ thuật, Robot 3-DOF là đối tượng lý tưởng để thử nghiệm và phát triển các thuật toán điều khiển hiện đại.

Việc nghiên cứu các phương pháp điều khiển khác nhau như PID, điều khiển học lặp (ILC), sẽ giúp đánh giá được ưu nhược điểm của từng phương pháp, từ đó lựa chọn giải pháp phù hợp cho từng yêu cầu ứng dụng cụ thể. Điều này không chỉ có giá trị về mặt học thuật mà còn mang tính ứng dụng thực tiễn cao trong việc phát triển các hệ thống Robot thông minh, linh hoạt hơn.

3. Mục tiêu đề tài:

- Tìm hiểu cấu trúc, động học và động lực học của cánh tay Robot 3 bậc tự do.
- Xây dựng và mô phỏng mô hình cánh tay Robot bằng phần mềm Solidworks và Matlab/Simulink.
- Nghiên cứu và so sánh hiệu quả của phương pháp điều khiển PID và điều khiển học lặp (ILC).
- Phân tích và đánh giá kết quả đạt được thông qua kết quả mô phỏng thực nghiệm.

4. Đối tượng và phạm vi nghiên cứu:

a. Đối tượng nghiên cứu:

- Mô hình cánh tay Robot 3 bậc tự do (3-DOF).
- Phương pháp điều khiển PID.
- Phương pháp điều khiển học lặp (ILC).

b. Phạm vi nghiên cứu:

- Đề tài chỉ nghiên cứu cánh tay Robot với 3 bậc tự do (3-DOF), gồm 3 khớp quay (cấu trúc R-R-R).
- Tập trung vào việc thiết kế, mô phỏng, so sánh phương pháp điều khiển PID và điều khiển học lặp (ILC).
- Mô phỏng tập trung nhiệm làm sao cho cánh tay Robot hoạt động bám theo một quỹ đạo cho trước.

5. Phương pháp nghiên cứu:

- Tìm hiểu cơ sở lý thuyết về mô hình cánh tay Robot 3 bậc tự do (3-DOF) thông qua các bài báo khoa học, internet, sách...
- Thiết lập mô hình động học và động lực học cho cánh tay Robot 3 bậc tự do (3-DOF) bằng các cách sau:
 - Phép biến đổi Denavit-Hartenberg (D-H) để mô tả cấu trúc hình học.
 - Phương trình Lagrange để xây dựng mô hình động lực học.
- Thiết kế và mô phỏng tay máy bằng phần mềm Matlab Simulink và Solidworks.
- Tạo ra một quỹ đạo không gian cho trước là đầu vào của các bộ điều khiển trên phần mềm Matlab Simulink.
- Tìm hiểu lý thuyết về các phương pháp điều khiển cho tay máy Robot 3 bậc tự do hoạt động bám theo quỹ đạo cho trước.

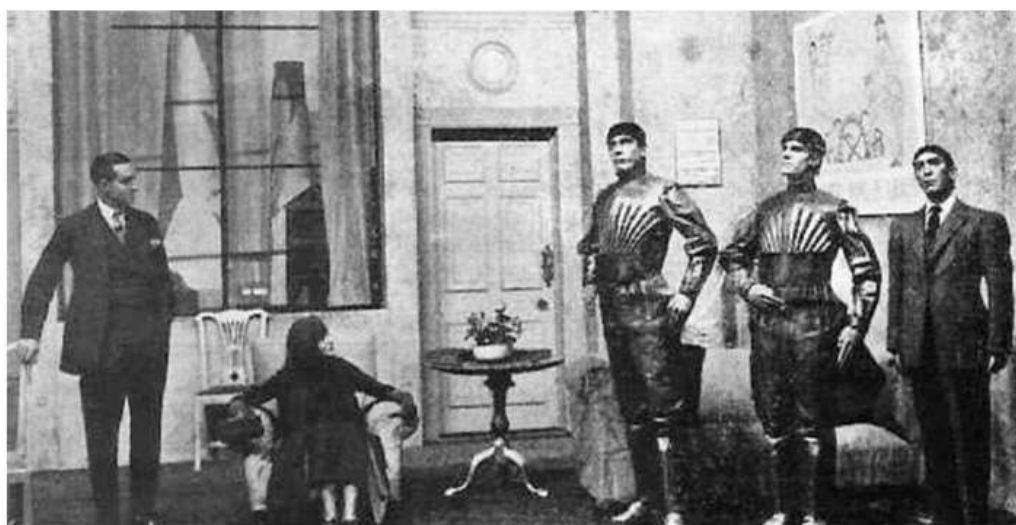
6. Cấu trúc của đồ án tốt nghiệp:

- Mục lục
- Danh sách các bảng, hình vẽ
- Danh sách các ký hiệu, chữ viết tắt
- Mở đầu
- Chương 1: Tổng quan đề tài
- Chương 2: Xây dựng mô hình động lực học robot 3dof
- Chương 3: Các phương pháp điều khiển cánh tay robot
- Chương 4: Mô phỏng điều khiển cánh tay robot và đánh giá kết quả
- Kết luận và hướng phát triển
- Tài liệu tham khảo
- Phụ lục

Chương 1: TỔNG QUAN ĐỀ TÀI

1.1. Sơ lược quá trình phát triển của Robot

Thuật ngữ “ Robot” xuất phát từ tiếng Sec (Czech) “Robota” có nghĩa là công việc tạp dịch trong vở kịch Rossum’s Universal Robots của Karel Capek, vào năm 1921. Trong vở kịch này, Rossum và con trai của ông ta đã chế tạo ra những chiếc máy gần giống với con người để phục vụ con người. Có lẽ đó là một gợi ý ban đầu cho các nhà sáng chế kỹ thuật về những cơ cấu, máy móc bắt chước các hoạt động cơ bắp của con người.



Hình 1.1: Robot trong vở kịch Rossum’s Universal Robots

Đầu thập kỷ 60, công ty Mỹ AMF (American Machine and Foundry Company) quảng cáo một loại máy tự động vận năng và gọi là “Người máy công nghiệp” (Industrial Robot). Ngày nay người ta đặt tên người máy công nghiệp (hay Robot công nghiệp) cho những loại thiết bị có dáng dấp và một vài chức năng như tay người được điều khiển tự động để thực hiện một số thao tác sản xuất.

Về mặt kỹ thuật, những robot công nghiệp ngày nay, có nguồn gốc từ hai lĩnh vực kỹ thuật ra đời sớm hơn đó là các cơ cấu điều khiển từ xa (Teleoperators) và các máy công cụ điều khiển số (NC - Numerically Controlled machine tool).

Các cơ cấu điều khiển từ xa (hay các thiết bị kiểu chủ-tớ) đã phát triển mạnh trong chiến tranh thế giới lần thứ hai nhằm nghiên cứu các vật liệu phóng xạ. Người thao tác được tách biệt khỏi khu vực phóng xạ bởi một bức tường có một hoặc vài cửa quan sát để có thể nhìn thấy được công việc bên trong. Các cơ cấu điều khiển từ xa thay thế cho cánh tay của người thao tác; nó gồm có một bộ kẹp ở bên trong (tớ) và hai tay cầm ở bên ngoài (chủ). Cả hai, tay cầm và bộ kẹp, được nối với nhau bằng một cơ cấu sáu bậc

tự do để tạo ra các vị trí và hướng tùy ý của tay cầm và bộ kẹp. Cơ cấu dùng để điều khiển bộ kẹp theo chuyển động của tay cầm.

Vào khoảng năm 1949, các máy công cụ điều khiển số ra đời, nhằm đáp ứng yêu cầu gia công các chi tiết trong ngành chế tạo máy bay. Những Robot đầu tiên thực chất là sự nối kết giữa các khâu cơ khí của cơ cấu điều khiển từ xa với khả năng lập trình của máy công cụ điều khiển số.

Một trong những Robot Công nghiệp đầu tiên được chế tạo là Robot Versatran của công ty AMF, Mỹ. Cũng vào khoảng thời gian này ở Mỹ xuất hiện loại Robot Unimate (1900) được dùng đầu tiên trong kỹ nghệ ô tô.

Tiếp theo Mỹ, các nước khác bắt đầu sản xuất Robot công nghiệp: Anh (1967), Thụy Điển và Nhật (1968) theo bản quyền của Mỹ, CHLB Đức (1971), Pháp (1972), Ý (1973)...

Tính năng làm việc của Robot ngày càng được nâng cao, nhất là khả năng nhận biết và xử lý. Năm 1967 ở trường Đại học tổng hợp Stanford (Mỹ) đã chế tạo ra mẫu Robot hoạt động theo mô hình “mắt-tay”, có khả năng nhận biết và định hướng bàn kẹp theo vị trí vật kẹp nhờ các cảm biến. Năm 1974 Công ty Mỹ Cincinnati đưa ra loại robot được điều khiển bằng máy vi tính, gọi là robot T3 (The Tomorrow Tool: Công cụ của tương lai). Robot này có thể nâng được vật có khối lượng đến 40 Kg.

Có thể nói, Robot là sự tổ hợp khả năng hoạt động linh hoạt của các cơ cấu điều khiển từ xa với mức độ “tri thức” ngày càng phong phú của hệ thống điều khiển theo chương trình số cũng như kỹ thuật chế tạo các bộ cảm biến, công nghệ lập trình và các phát triển của trí khôn nhân tạo, hệ chuyên gia...

Trong những năm sau này, việc nâng cao tính năng hoạt động của Robot không ngừng phát triển. Các Robot được trang bị thêm các loại cảm biến khác nhau để nhận biết môi trường xung quanh, cùng với những thành tựu to lớn trong lĩnh vực Tin học - Điện tử đã tạo ra các thế hệ robot với nhiều tính năng đặc biệt, Số lượng Robot ngày càng gia tăng, giá thành ngày càng giảm. Nhờ vậy, Robot công nghiệp đã có vị trí quan trọng trong các dây chuyền sản xuất hiện đại.

1.2. Vai trò và ứng dụng của Robot trong sản xuất

Theo ước tính của Liên đoàn Robot quốc tế (IRF), hiện nay trên thế giới có khoảng 50% số lượng Robot được sử dụng tại châu Á (trong đó Nhật Bản chiếm 30%), 32% ở Châu Âu, 16% ở Bắc Mỹ, 1% ở Australia và 1% ở châu Phi.

Trong đó, Robot được sử dụng trong các ngành chế tạo ô tô chiếm 33,2%, ngành unsecified chiếm 25%, ngành điện- điện tử 9,9%, ngành hóa chất + cao su + nhựa

chiếm 9,4%, ngành chế tạo máy 4,3%, ngành điện tử viễn thông chiếm 2,5%, sản xuất metal chiếm 3,7%, ngành sản xuất gỗ 2,5% và các ngành khác là 10,3%.

Từ khi mới ra đời Robot công nghiệp được áp dụng trong nhiều lĩnh vực dưới góc độ thay thế sức người. Nhờ vậy các dây chuyền sản xuất được tổ chức lại, năng suất và hiệu quả sản xuất tăng lên rõ rệt.

Mục tiêu ứng dụng Robot công nghiệp nhằm góp phần nâng cao năng suất dây chuyền công nghệ, giảm giá thành, nâng cao chất lượng và khả năng cạnh tranh của sản phẩm đồng thời cải thiện điều kiện lao động. Đạt được các mục tiêu trên là nhờ vào những khả năng to lớn của Robot như : làm việc không biết mệt mỏi, chịu được phóng xạ và các môi trường làm việc độc hại, nhiệt độ cao, “cảm thấy” được cả từ trường và “nghe” được cả siêu âm... Robot được dùng thay thế con người trong các trường hợp trên hoặc thực hiện các công việc tuy không nặng nhọc nhưng đơn điệu, dễ gây mệt mỏi, nhàm lẫn.

Trong ngành cơ khí, Robot được sử dụng nhiều trong công nghệ đúc, công nghệ hàn (hình 1.2), cắt kim loại, sơn, phun phủ kim loại, tháo lắp vận chuyển phôi, lắp ráp sản phẩm... Ngày nay đã xuất hiện nhiều dây chuyền sản xuất tự động gồm các máy CNC với Robot công nghiệp, các dây chuyền đó đạt mức tự động hoá cao, mức độ linh hoạt cao... Ở đây các máy và Robot được điều khiển bằng cùng một hệ thống chương trình.



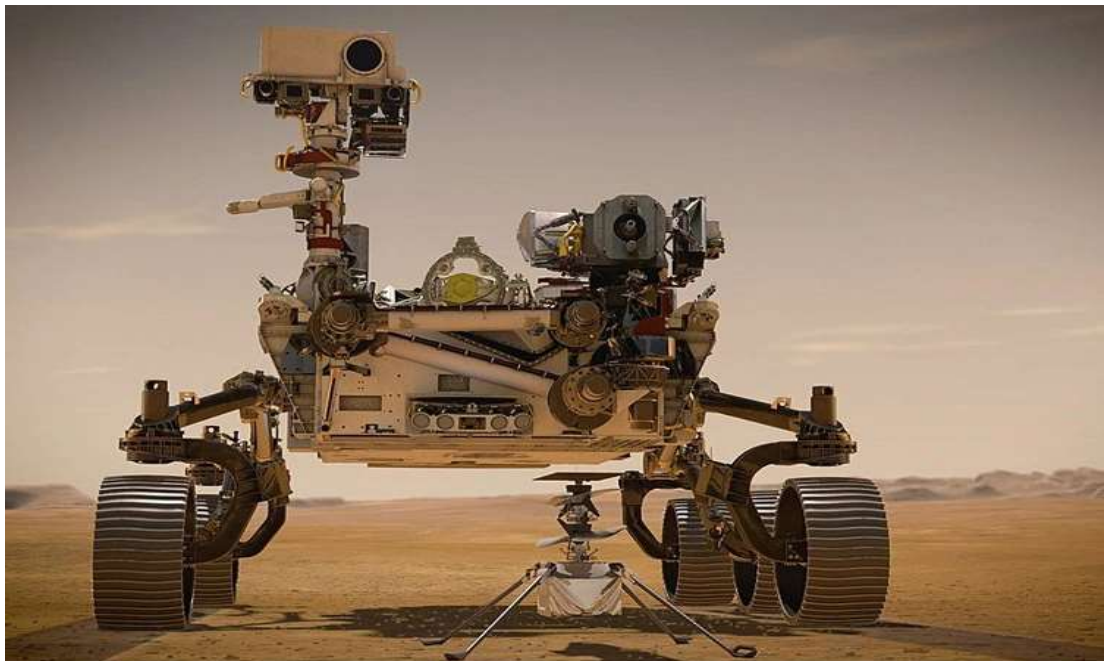
Hình 1.2: Robot trong dây chuyền sản xuất ô tô



Hình 1.3: Robot trong lĩnh vực nông nghiệp

Ngày nay đã xuất hiện nhiều dây chuyền sản xuất tự động gồm các máy CNC với Robot công nghiệp, các dây chuyền đó đạt mức tự động hoá cao, mức độ linh hoạt cao... Ở đây các máy và robot được điều khiển bằng cùng một hệ thống chương trình.

Ngoài các phân xưởng, nhà máy, kỹ thuật robot cũng được sử dụng trong việc khai thác thềm lục địa và đại dương, trong y học, sử dụng trong quốc phòng, trong chinh phục vũ trụ (hình 1.4), trong công nghiệp nguyên tử, trong các lĩnh vực xã hội...



Hình 1.4: Robot trong lĩnh vực thăm dò vũ trụ

Rõ ràng là khả năng làm việc của Robot trong một số điều kiện vượt hơn khả năng của con người, do đó nó là phương tiện hữu hiệu để tự động hoá, nâng cao năng suất lao động, giảm nhẹ cho con người những công việc nặng nhọc và độc hại. Nhược điểm lớn nhất của Robot là chưa linh hoạt như con người, trong dây chuyền tự động, nếu có một Robot bị hỏng có thể làm ngừng hoạt động của cả dây chuyền, cho nên Robot vẫn luôn hoạt động dưới sự giám sát của con người.

1.3. Các khái niệm và định nghĩa về robot công nghiệp

1.3.1. Định nghĩa về Robot công nghiệp

Hiện nay có nhiều định nghĩa về Robot, có thể điểm qua một số định nghĩa như sau:

Định nghĩa theo ISO 8373 (Tổ chức Tiêu chuẩn hóa Quốc tế): Robot công nghiệp là một thiết bị cơ khí tự động, có thể lập trình được theo nhiều trục, có khả năng di chuyển và thao tác các vật thể, công cụ, hoặc thiết bị chuyên dụng thông qua các chương trình định sẵn để thực hiện các nhiệm vụ tự động trong công nghiệp.

Định nghĩa theo viện Robot Hoa Kỳ (RIA - Robotic Industries Association): Robot công nghiệp là một thiết bị điều khiển tự động, có thể lập trình lại nhiều lần, có tính linh hoạt cao, hoạt động trong môi trường ba chiều để thực hiện một chuỗi các nhiệm vụ như lắp ráp, hàn, sơn, xử lý vật liệu... trong công nghiệp.

Định nghĩa theo GOCT 25686-85 (Nga): Robot công nghiệp là một máy tự động, được đặt cố định hoặc di động được, liên kết giữa một tay máy và một hệ thống điều khiển chương trình, có thể lập trình lại để hoàn thành các chức năng vận động và điều khiển trong quá trình sản xuất.

Có thể nói Robot công nghiệp là một máy tự động linh hoạt thay thế từng phần hoặc toàn bộ các hoạt động cơ bắp và hoạt động trí tuệ của con người trong nhiều khả năng thích nghi khác nhau.

Robot công nghiệp có khả năng chương trình hóa linh hoạt trên nhiều trục chuyển động, biểu thị cho số bậc tự do của chúng. Robot công nghiệp được trang bị những bàn tay máy hoặc các cơ cấu chấp hành, giải quyết những nhiệm vụ được xác định trong các quá trình công nghệ hoặc trực tiếp tham gia thực hiện các công đoạn (sơn, hàn, phun phủ, rót kim loại vào khuôn đúc, lắp ráp máy...) hoặc phục vụ các quá trình công nghệ (tháo lắp chi tiết gia công, dao cụ, đồ đá...) với những thao tác cầm nắm, vận chuyển và trao đổi các đối tượng với các trạm công nghệ, trong một hệ thống máy tự động linh hoạt, được gọi là “Hệ thống tự động linh hoạt Robot hóa” cho phép thích ứng nhanh và thao tác đơn giản khi nhiệm vụ sản xuất thay đổi.

1.3.2. Các khái niệm

- *Bậc tự do của Robot:*

Bậc tự do là số khả năng chuyển động của một cơ cấu (chuyển động quay hoặc tịnh tiến). Để dịch chuyển được một vật thể trong không gian, cơ cấu chấp hành của Robot phải đạt được một số bậc tự do. Nói chung cơ hệ của Robot là một cơ cấu hở, do đó bậc tự do của nó có thể tính theo công thức:

$$w = 6n - \sum_{i=1}^5 ip_i$$

Ở đây: n - số khâu động;

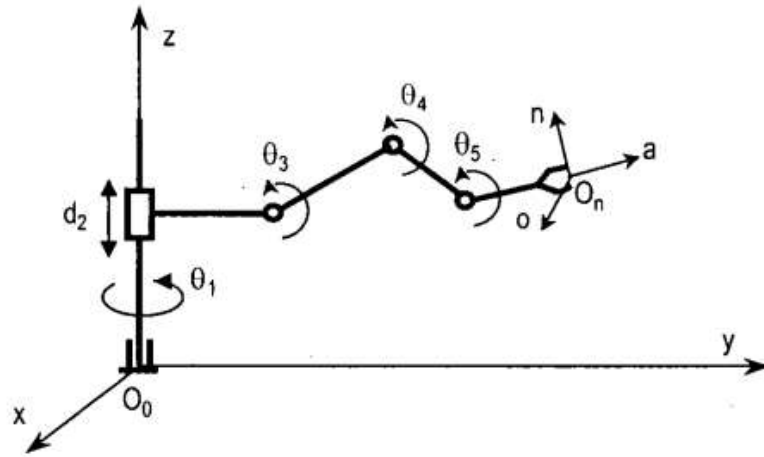
p_i - số khớp các loại i ($i=1,2,\dots,5$: số bậc tự do bị hạn chế).

Đối với các cơ cấu có các khâu được nối với nhau bằng khớp quay hoặc tịnh tiến (khớp động loại 5) thì số bậc tự do bằng với số khâu động. Đối với cơ cấu hở, số bậc tự do bằng tổng số bậc tự do của khớp động.

Để định vị và định hướng khâu chấp hành cuối một cách tùy ý trong không gian 3 chiều Robot cần có 6 bậc tự do, trong đó 3 bậc tự do đầu để định vị và 3 bậc tự do còn lại dùng để định hướng. Một số công việc đơn giản như nâng hạ, sắp xếp... có thể yêu cầu số bậc tự do ít hơn. Các Robot hàn, sơn... thường yêu cầu 5 hoặc 6 bậc tự do. Trong một số trường hợp cần sự khéo léo, linh hoạt hoặc khi yêu cầu tối ưu quỹ đạo,... người ta dùng Robot với số bậc tự do lớn hơn 6. Khi số bậc tự do lớn thì kết cấu của Robot và việc lập trình điều khiển sẽ phức tạp, giá thành Robot cao hơn.

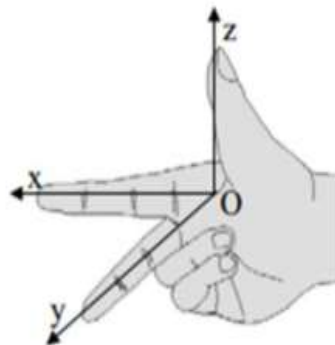
- *Hệ tọa độ:*

Mỗi Robot thường bao gồm nhiều khâu (links) liên kết với nhau qua các khớp (joints), tạo thành một xích động học xuất phát từ một khâu cơ bản (base) đứng yên. Hệ tọa độ gắn với khâu cơ bản gọi là hệ tọa độ cơ bản (hay hệ tọa độ gốc). Các hệ tọa độ trung gian khác gắn với các khâu động gọi là hệ tọa độ suy rộng. Trong từng thời điểm hoạt động, các tọa độ suy rộng xác định cấu hình của Robot bằng các chuyển dịch dài hoặc các chuyển dịch góc của các khớp tịnh tiến hoặc khớp quay. Các tọa độ suy rộng được gọi là biến khớp.



Hình 1.5: Các tọa độ suy rộng của Robot

Các hệ tọa độ gắn trên các khâu của Robot phải tuân theo quy tắc bàn tay phải: Dùng tay phải, nắm hai ngón tay út và áp út vào lòng bàn tay, xòe 3 ngón: cái, trỏ và giữa theo 3 phương vuông góc nhau, nếu chọn ngón cái là phương và chiều của trục z, thì ngón trỏ chỉ phương, chiều của trục x và ngón giữa sẽ biểu thị phương và chiều của trục y.



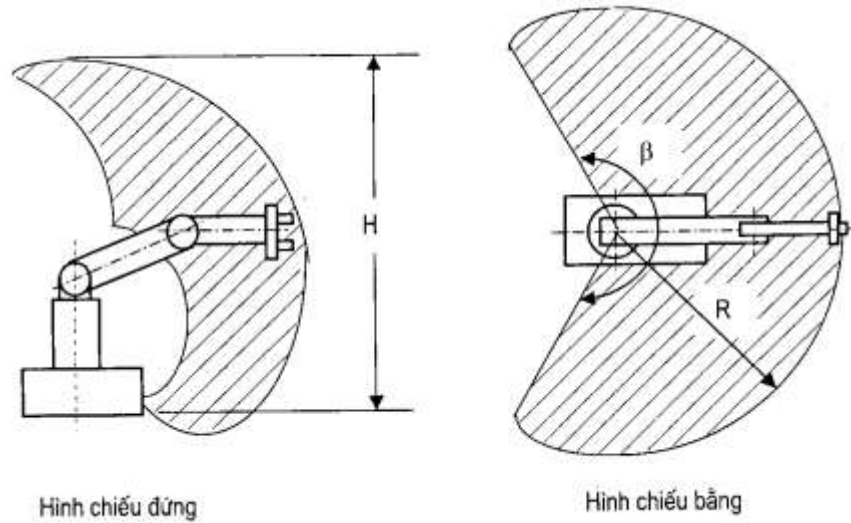
Hình 1.6: Quy tắc bàn tay phải

Trong Robot ta thường dùng chữ O và chỉ số n để chỉ hệ tọa độ gắn trên khâu thứ n . Như vậy hệ tọa độ cơ bản (hệ tọa độ gắn với các khâu cố định) sẽ được ký hiệu là O_o , hệ tọa độ gắn trên các khâu trung gian tương ứng sẽ là O_1, O_2, \dots, O_{n-1} , hệ tọa độ gắn trên khâu chấp hành cuối ký hiệu là O_n .

- *Trường công tác của Robot:*

Trường công tác (hay vùng làm việc, không gian công tác) của Robot là toàn bộ thể tích được quét bởi khâu chấp hành cuối khi Robot thực hiện tất cả các chuyển động có thể. Trường công tác bị ràng buộc bởi các thông số hình học của Robot cũng như các ràng buộc cơ học của khớp, ví dụ như một khớp quay có chuyển động

trong phạm vi nhỏ hơn 360^0 . Người ta thường dùng hai hình chiếu để mô tả trường công tác của một Robot.

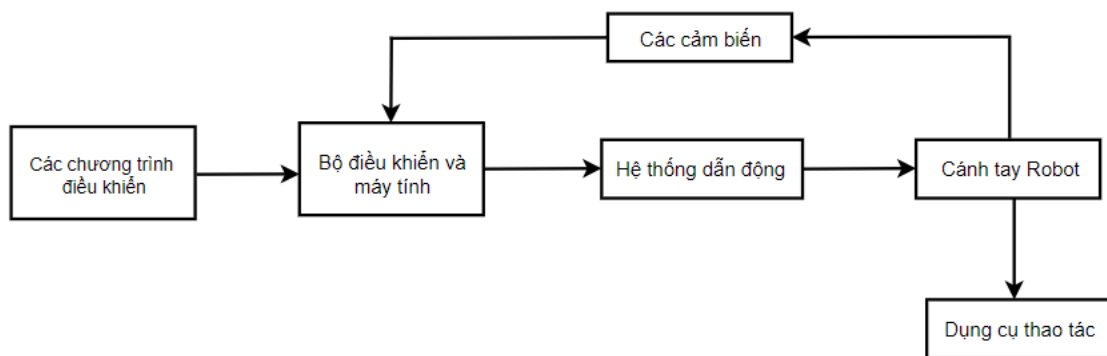


Hình 1.7: Trường công tác của Robot

1.4. Cấu trúc cơ bản của Robot công nghiệp

1.4.1. Các thành phần chính của Robot công nghiệp

Một Robot công nghiệp thường bao gồm các thành phần chính như: cánh tay Robot, hệ thống dẫn động, dụng cụ thao tác, các cảm biến, bộ điều khiển, máy tính... các chương trình điều khiển cũng được coi là một thành phần của hệ thống Robot.



Hình 1.8: Các thành phần chính của hệ thống Robot

- Cánh tay Robot (tay máy) là kết cấu cơ khí gồm các khâu liên kết với nhau bằng các khớp động để tạo nên những chuyển động cơ bản của Robot.
- Hệ thống dẫn động thường dùng trong Robot là các động cơ điện (một chiều hoặc động cơ bước), các hệ thống xi lanh khí nén, thủy lực để tạo động lực cho tay máy hoạt động.

- Bộ điều khiển và máy tính là trung tâm xử lý thông tin, đóng vai trò thu thập dữ liệu đầu vào, xử lý lệnh điều khiển và gửi tín hiệu đến hệ thống dẫn động. Ngoài ra, bộ điều khiển còn nhận lệnh từ phần mềm điều khiển, thực hiện các tính toán, lập kế hoạch quỹ đạo, đồng thời đọc dữ liệu từ các cảm biến phản hồi.
- Dụng cụ thao tác được gắn trên khâu cuối của Robot, dụng cụ của Robot có thể có nhiều kiểu khác nhau như: dạng bàn tay để bắt đối tượng hoặc các công cụ làm việc như mỏ hàn, đầu phun sơn, đầu laser...
- Các phần mềm để lập trình và các chương trình điều khiển Robot được cài đặt trên máy tính, dùng để điều khiển Robot thông qua bộ điều khiển (Controller). Chương trình điều khiển là các mã lệnh, thuật toán hoặc các phần mềm do con người lập trình, nhằm quy định chuyển động, quỹ đạo, lực và các phản hồi của Robot.
- Các cảm biến được sử dụng phổ biến trên Robot để nhận biết trạng thái của bản thân hoặc biết đối tượng làm việc và môi trường chung quanh, tạo ra các tín hiệu phản hồi về bộ điều khiển để điều chỉnh các hoạt động của Robot.

1.4.2. Cánh tay máy Robot

Tay máy là thành phần quan trọng, nó quyết định khả năng làm việc của một hệ thống Robot. Kết cấu của tay máy là một cơ cấu cơ khí có cấu trúc được mô phỏng theo cấu tạo và chức năng của cánh tay người, bao gồm nhiều khớp nối và đoạn liên kết, được thiết kế nhằm thực hiện các thao tác tự động như gấp, hàn, lắp ráp hoặc dịch chuyển vật thể. Về cơ bản, tay máy bao gồm các thành phần chính sau:

Phần đầu tiên là đế, đóng vai trò làm nền tảng vững chắc cho toàn bộ Robot. Đế thường được cố định xuống sàn hoặc bệ máy, đồng thời chứa khớp quay đầu tiên cho phép toàn bộ tay máy xoay quanh trục thẳng đứng, tạo nên chuyển động quay cơ bản.

Tiếp theo là các đoạn tay được kết nối với nhau thông qua các khớp nối. Các khớp này có thể là khớp quay (cho phép xoay quanh trục) hoặc khớp tịnh tiến (cho phép trượt theo trục thẳng). Mỗi khớp tương ứng với một bậc tự do (DOF - Degree of Freedom). Một tay máy Robot thông thường có từ 3 đến 6 bậc tự do, cho phép định vị chính xác vị trí và định hướng của đầu chấp hành trong không gian ba chiều.

Phần cuối của tay máy là cơ cấu chấp hành, là bộ phận tiếp xúc trực tiếp với đối tượng cần thao tác. Tùy vào ứng dụng, cơ cấu này có thể là kẹp, súng hàn, bút vẽ, đầu hút chân không hoặc các dụng cụ đặc biệt khác.

Toàn bộ hệ thống được điều khiển thông qua bộ điều khiển trung tâm, kết hợp với các cơ cấu truyền động như động cơ servo, động cơ bước, khí nén hoặc thủy lực, giúp tạo ra chuyển động chính xác tại từng khớp nối. Ngoài ra, các cảm biến vị trí, lực hoặc góc quay có thể được tích hợp để nâng cao khả năng điều khiển và phản hồi của hệ thống.

1.5. Phân loại Robot công nghiệp

Robot công nghiệp rất phong phú đa dạng, có thể được phân loại theo các cách sau:

- *Phân loại theo kết cấu:*

Theo kết cấu của tay máy người ta phân thành Robot kiểu tọa độ Đề-Các, kiểu tọa độ trụ, kiểu tọa độ cầu, kiểu tọa độ góc, Robot SCARA.

- *Phân loại theo hệ thống truyền động:*

Có các dạng truyền động phổ biến là:

Hệ truyền động điện: Thường dùng các động cơ điện 1 chiều (DC: Direct Current) hoặc các động cơ bước (step motor). Loại truyền động này dễ điều khiển, kết cấu gọn.

Hệ truyền động thủy lực: Có thể đạt được công suất cao, đáp ứng những điều kiện làm việc nặng. Tuy nhiên hệ thống thủy lực thường có kết cấu cồng kềnh, tồn tại độ phi tuyến lớn khó xử lý khi điều khiển.

Hệ truyền động khí nén: Có kết cấu gọn nhẹ hơn do không cần cầu dẫn ngược những lại phải gắn liền với trung tâm tạo ra khí nén. Hệ này làm việc với công suất trung bình và nhỏ, kém chính xác, thường chỉ thích hợp với các Robot hoạt động theo chương trình và nhỏ, kém chính xác, thường chỉ thích hợp với các Robot hoạt động theo chương trình định sẵn với các thao tác đơn giản “nhấc lên - hạ xuống”

- *Phân loại theo ứng dụng:*

Dựa vào ứng dụng của Robot trong sản xuất có Robot sơn, Robot hàn, Robot lắp ráp, Robot chuyển phôi...

- *Phân loại theo cách thức và đặc trưng của phương pháp điều khiển:*

Có Robot điều khiển hở (mạch điều khiển không có các quan hệ phản hồi), Robot điều khiển kín (hay điều khiển servo): sử dụng cảm biến, mạch phản hồi để tăng độ chính xác với mức độ linh hoạt khi điều khiển.

Ngoài ra còn có thể có các cách phân loại khác tùy theo quan điểm và mục đích nghiên cứu.

Kết luận: Trong chương này, chúng ta đã điếm qua tổng quan về lịch sử hình thành và sơ lược quá trình phát triển của Robot, ngoài ra còn tìm hiểu cơ bản về cấu trúc, các khái niệm và định nghĩa về Robot công nghiệp. Sự ra đời của Robot công nghiệp đã làm thay đổi căn bản cách thức vận hành của dây chuyền sản xuất, nâng cao năng suất, chất lượng sản phẩm mà còn đảm bảo độ chính xác. Từ vai trò thay thế sức lao động thủ công Robot đang dần trở thành người cộng tác thông minh của con người trong nhiều lĩnh vực. Hiện nay, Robot 3 bậc tự do (3DOF) xuất hiện phổ biến trong các ứng dụng như dây chuyền sản xuất tự động, phòng thí nghiệm nghiên cứu, giáo dục STEM và cả trong

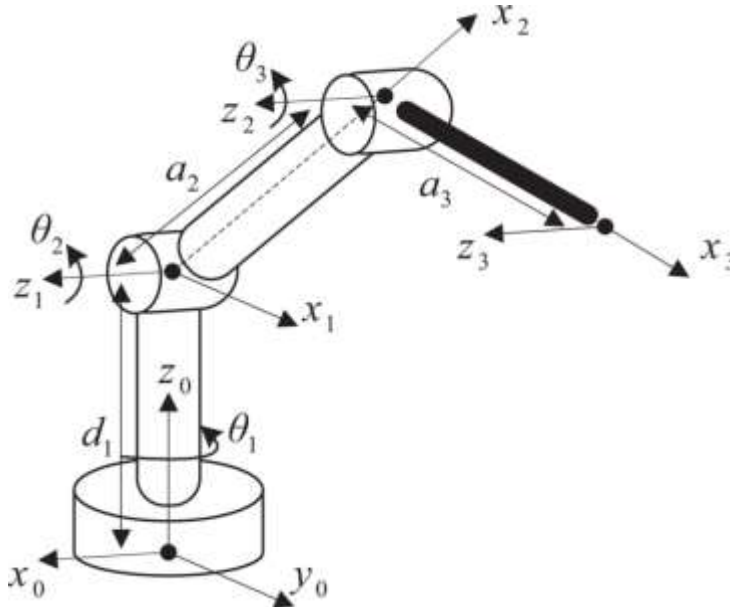
lĩnh vực mô phỏng phần thuật y tế. Với cấu trúc cơ khí đơn giản nhưng vẫn đảm bảo được sự linh hoạt nhất định, Robot 3 DOF dễ dàng triển khai và vận hành, chi phí chế tạo thấp hơn so với các Robot có nhiều bậc tự do, đồng thời dễ kiểm soát và lập trình. Điều này đặt ra yêu cầu cấp thiết về việc nghiên cứu, phát triển và ứng dụng của các phương pháp điều khiển, mô hình Robot phù hợp, cả về cấu trúc, tính năng và chi phí để phục vụ nhu cầu thực tiễn. Đây cũng chính là lý do nhóm chúng em chọn đề tài **“Nghiên cứu thuật toán điều khiển tay máy Robot 3 bậc tự do”** cho đề án tốt nghiệp lần này.

Chương 2: XÂY DỰNG MÔ HÌNH ĐỘNG LỰC HỌC ROBOT 3DOF

2.1. Cấu trúc và thông số cánh tay robot 3 bậc tự do

2.1.1. Cấu trúc cánh tay robot 3 bậc tự do

Cánh tay Robot 3 bậc tự do có cấu trúc chung như hình 2.1 sau:



Hình 2.1: Cơ cấu Robot 3 bậc tự do (RRR)

Một cánh tay Robot bao gồm các khâu được nối với nhau bằng các khớp. Các khâu thường là các vật cứng tạo nên cấu trúc của một cánh tay, trong khi đó các khớp có thể là khớp quay hoặc khớp tịnh tiến cho phép chuyển động tương đối giữa các khâu. Góc khớp kí hiệu là theta đối với khớp quay, điều khiển vị trí và hướng của từng khâu trong cánh tay robot.

Cánh tay Robot 3 bậc tự do sẽ gồm có 1 đế và 3 khâu chính được nối với nhau bởi 3 khớp (quay hoặc tịnh tiến) tạo thành chuỗi động học của cánh tay Robot.

Theo phân loại các dạng chuyển động của các khớp cánh tay robot thì ta có 3 dạng chuyển động cơ bản:

- Khớp quay (Revolute – R): quay quanh trục
- Khớp tịnh tiến (Translation – T): tịnh tiến theo trục
- Khớp cầu (Spherical – S): quay 3 chiều (nhưng ít được sử dụng)

Với cánh tay robot 3 bậc tự do, ta sẽ có các cấu hình điển hình sau [5]:

- RRR (3 khớp quay): phổ biến, thường giống cánh tay người, thường dùng trong loại cánh tay robot PUMA 3 bậc tự do.
- RRT (2 khớp quay, 1 khớp tịnh tiến): thường dùng cho loại cánh tay robot SCARA 3 bậc tự do.

- TRR (1 khớp tịnh tiến, 2 khớp quay): thường được dùng nhiều trong các dây chuyền sản xuất linh kiện.
- TTT (3 khớp tịnh tiến): thường được dùng trong ứng dụng in 3D, máy CNC, khắc laser,...

Trong đề tài này, loại cánh tay robot 3 bậc tự do tập trung nghiên cứu là loại có 3 khớp quay RRR. Đây là loại Robot rất phổ biến trong công nghiệp nhờ vào những ưu điểm: ít tốn không gian, dễ dàng bố trí trong không gian làm việc, độ khéo léo, linh hoạt cao. Tuy nhiên nó có nhược điểm là độ cứng vững theo phương đứng thấp, độ chính xác giảm theo tầm hoạt động.

Các thành phần chính của cánh tay Robot 3 bậc tự do (RRR) gồm: Đế, khớp 1, khớp 2, khớp 3, các cánh tay và bàn tay Robot.

Chức năng của thành phần chính:

- Đế: Dùng để gắn khớp 1 và cố định cánh tay Robot.
- Khớp 1: Dùng để xoay toàn bộ cánh tay Robot theo trục đứng.
- Khớp 2: Dùng để nâng và hạ cánh tay Robot.
- Khớp 3: Dùng để điều chỉnh vị trí và góc của công cụ thao tác cuối.
- Các cánh tay: Là thanh nối giữa các khớp dùng để hỗ trợ truyền lực và tạo chiều dài và hỗ trợ không gian làm việc.
- Bàn tay Robot: Dùng để thực hiện các thao tác cuối cùng, tùy vào từng ứng dụng mà cấu trúc cánh tay có thể thay đổi

2.1.2. Thông số cánh tay robot 3 bậc tự do

Một cánh tay Robot muốn hoạt động một cách hiệu quả với độ chính xác cao thì phải có tỷ lệ chiều dài giữa các khâu hợp lý. Tỷ lệ hợp lý đó sẽ giúp cho cánh tay robot đạt được sự cân giữa khả năng vươn xa và khả năng điều khiển chính xác. Nếu khâu cuối quá dài, robot có thể mở rộng được vùng làm việc nhưng sẽ gặp khó khăn trong việc điều khiển. Ngược lại, nếu khâu đầu quá ngắn thì cánh tay robot có thể mất tính ổn định và khó tạo ra momen đủ lớn để nâng vật nặng. Vì vậy, việc thiết kế chiều dài cho các khâu cần dựa trên mục đích sử dụng cụ thể, khả năng tải trọng, phạm vi hoạt động mong muốn.

Dựa vào thông số chiều dài các khâu cánh tay robot 3 bậc tự do thực tế thì tỷ lệ các khâu của cánh tay robot thường dao động trong khoảng tỷ lệ phổ biến là (1:1:1 ~ 1:1.5:2).

Tùy vào ứng dụng cụ thể của mỗi loại cánh tay robot thì tỷ lệ chiều dài các khâu sẽ khác nhau, ví dụ như:

- Ứng dụng trong giáo dục – đào tạo cơ bản: tỷ lệ chiều dài các khâu thường là 1:1:1, vì nó có thiết kế đối xứng đơn giản cho việc mô hình hóa, điều khiển và mô phỏng.



Hình 2.2: Cánh tay Robot 3DOF có tỷ lệ các khâu 1:1:1 dùng trong giáo dục và đào tạo

- Ứng dụng gấp các vật nhỏ trong bán kính cố định: tỷ lệ chiều dài các khâu thường là 1:1.5:1, vì nó sẽ tăng tầm với ở khâu giữa để có thể gấp được nhiều vị trí mà vẫn giữ cho khâu đầu được ổn định.



Hình 2.3: Cánh tay Robot 3DOF có tỷ lệ các khâu là 1:1.5:1 dùng trong ứng dụng gấp vật

- Ứng dụng vẽ trên mặt phẳng 2D: tỷ lệ chiều dài các khâu thường là 1:1.5:2, vì cần vùng làm việc theo phương ngang, khâu cuối dài giúp vẽ được biên lớn hơn.



Hình 2.4: Cánh tay Robot 3DOF có tỷ lệ các khâu là 1:1.5:2 dùng trong vẽ 2D

Trong phạm vi đồ án lần này, nhóm chúng em lựa chọn kích thước cụ thể theo mô hình cánh tay robot công nghiệp nhỏ gọn của ZEKEEP mã số ZKBOT-3042B như hình 2.5 với tỷ lệ các khâu là 1:1.5:1.7, tải trọng 1kg, tầm với tối đa 400mm, trọng lượng 7 Kg [10]. Đây là loại robot 3DOF phổ biến, thường sử dụng trong các lĩnh vực như in 3D, vẽ tranh 2D, khắc logo hoặc hình bằng laser,...



Hình 2.5: Mô hình cánh tay Robot 3 bậc tự do ZEKEEP mã số ZKBOT-3042B

Dưới đây là các thông số của cánh tay robot:

Thông số	Ký hiệu	Giá trị	Đơn vị
Bán kính link 1	R	92.5	mm
Chiều dài link 1 (không tính đế)	l_1	132	mm
Chiều dài link 2	l_2	200	mm
Chiều dài link 3 (tính cả mũi vẽ, in)	l_3	230	mm
Trọng lượng link 1	m_1	3	Kg
Trọng lượng link 2	m_2	2	Kg
Trọng lượng link 3	m_3	2	Kg
Gia tốc trọng trường	g	9.8	m/s^2

Bảng 3.1 Thông số cánh tay robot 3 bậc tự do

2.2. Động học thuận cánh tay Robot 3 bậc tự do

Sau khi đã xác định được cấu trúc và thông số cánh tay robot 3 bậc tự do mà chúng ta muốn thiết kế, thì bước kế tiếp trong quá trình xây dựng mô hình động lực học cánh tay robot là tiến hành phân tích động học thuận để xác định được vị trí và hướng của đầu cuối khi ta biết các góc khớp.

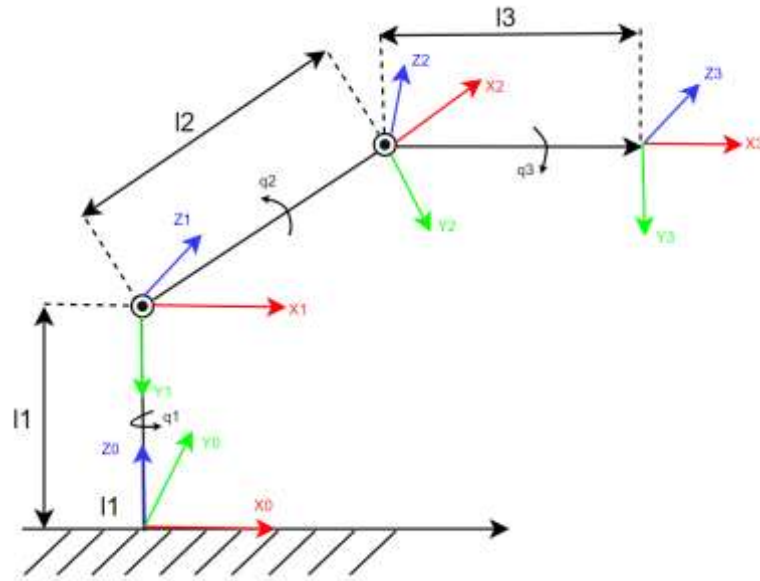
2.2.1. Gắn hệ tọa độ cho từng khâu

Để thực hiện phân tích động học thuận cánh tay robot, chúng ta cần gắn hệ tọa độ cho từng khâu của robot nhằm mô tả chính xác chuyển động và vị trí tương đối giữa các khâu. Việc gắn hệ tọa độ tọa độ tuân theo quy tắc Denavit - Hartenberg (D-H) [1], cho phép mô hình hóa hình học của robot một cách có hệ thống và dễ dàng xây dựng các ma trận chuyển đổi giữa các khâu. Đây là bước nền tảng quan trọng giúp biểu diễn các quan hệ không gian giữa các khâu trong mô hình toán học, từ đó phục vụ cho việc tính toán vị trí đầu cuối và xây dựng bộ điều khiển trong các chương tiếp theo.

Về hệ tọa độ, ta xem mỗi Robot như một xích động học gồm các khâu liên kết bởi các khớp quay và tịnh tiến. Để nghiên cứu động học ta sẽ gắn hệ tọa độ lên mỗi khâu. Việc gắn hệ tọa độ sẽ được thực hiện từ gốc đến khâu chấp hành, thường thì trục z sẽ được đặt trùng với trục chuyển động của khớp, trục x thường được đặt vuông góc chung với trục z của 2 khâu liên kế nhau, còn trục y được đặt sao cho ba trục x y z tạo thành hệ tọa độ vuông góc chuẩn quy tắc tay phải.

Ở đây, số bậc tự do của cánh tay Robot ta nghiên cứu là 3 bậc, ta sẽ chọn hệ tọa độ gốc O_0 của Robot như hình , các trục z sẽ được đặt cùng phương với các trục khớp.

Tiếp theo, ta gắn hệ tọa độ như hình 2.6 sau:



Hình 2.6: Cánh tay Robot 3 bậc tự do trong hệ trục tọa độ

2.2.2. Lập bảng thông số D-H (Denavit-Hartenberg)

Sau khi gán hệ tọa độ cho mỗi khâu của cánh tay Robot thì mỗi hệ tọa độ đó sẽ mô tả vị trí và hướng của khâu đó so với các khâu trước thông qua ma trận biến đổi. Phương pháp Denavit-Hartenberg (D-H) giúp ta chuẩn hóa cách xác định các hệ tọa độ này dựa trên 4 tham số đặc trưng [1].

Với mỗi khâu thứ i , ta cần xác định 4 tham số D-H:

- a_i – Độ dài pháp tuyến chung: Khoảng cách giữa 2 trục z .
- α_i – Góc giữa các trục khớp: Góc hợp giữa 2 trục z .
- d_i – Khoảng cách dọc trục: Khoảng cách giữa 2 trục x .
- θ_i – Góc xoay khớp: Góc hợp giữa 2 trục x .

Ta có bảng thông số D-H cho cánh tay Robot 3 bậc tự do như sau:

Khâu	a_i	α_i	d_i	θ_i
1	0	-90°	l_1	q_1
2	l_2	0	0	q_2
3	l_3	0	0	q_3

Bảng 2.1: Bảng thông số D-H cho cánh tay Robot 3 bậc tự do

2.2.3. Xác định các ma trận suy rộng A_i

Vị trí và hướng giữa các hệ tọa độ được biểu diễn bằng các phép biến đổi thuận nhất. Các phép biến đổi này sẽ được biểu diễn bằng các ma trận suy rộng A_i , thể hiện phép quay và phép tịnh tiến giữa hai khâu liên tiếp nhau [1].

Các ma trận thể hiện phép quay và phép tịnh tiến có dạng như sau:

- Quay quanh trục x 1 góc α :

$$Rot_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Quay quanh trục y 1 góc α :

$$Rot_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & \sin \alpha & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \alpha & 0 & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Quay quanh trục z 1 góc α :

$$Rot_z(\alpha) = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & 0 \\ \sin \alpha & \cos \alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Tịnh tiến theo x 1 khoảng d :

$$Trans_x(d) = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Tịnh tiến theo y 1 khoảng d :

$$Trans_y(d) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Tịnh tiến theo z 1 khoảng d :

$$Trans_z(d) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Để mô tả chuyển động tương đối giữa hai khâu liên tiếp trong robot thì chúng ta cần xây dựng một ma trận biến đổi biểu diễn sự chuyển đổi từ hệ tọa độ khâu trước sang khâu sau được gọi là ma trận suy rộng A . Ma trận suy rộng A này bao gồm cả phép quay và tịnh tiến và được biểu diễn dưới dạng:

$$A = \begin{bmatrix} r & p \\ 0 & 1 \end{bmatrix}$$

Trong đó:

- r : ma trận quay 3×3 thể hiện hướng của khâu sau so với khâu trước.
- p : vecto tịnh tiến 3×1 thể hiện vị trí của gốc hệ tọa độ sau so với hệ tọa độ trước.

Phương pháp D-H [1] chuẩn hóa phép biến đổi giữa 2 khâu bằng cách chia nó thành 4 bước cơ bản, kết hợp các phép quay và tịnh tiến lần lượt theo các trục. Để biến đổi từ hệ tọa độ khâu trước $i-1$ sang khâu sau i thì ta thực hiện:

- Quay quanh trục z_{i-1} một góc $\theta_i \rightarrow Rot_z(\theta_i)$
- Tịnh tiến dọc trục z_{i-1} một đoạn $d_i \rightarrow Trans_z(d_i)$
- Tịnh tiến dọc trục x_i một đoạn $a_i \rightarrow Trans_x(a_i)$
- Quay quanh trục x_i một góc $\alpha_i \rightarrow Rot_x(\alpha_i)$

Ma trận suy rộng chuyển đổi A_i từ khâu trước $i-1$ sang khâu sau i là tích của các phép quay và tịnh tiến được viết dưới dạng sau:

$$A_i = Rot_z(\theta_i) \times Trans_z(d_i) \times Trans_x(a_i) \times Rot_x(\alpha_i)$$

Vậy dạng tổng quát của ma trận suy rộng giữa hai khâu $i-1$ và i là ma trận ${}^{i-1}_i A$ có dạng:

$${}^{i-1}_i A = \begin{bmatrix} \cos q & -\sin q \cos \alpha & \sin q \sin \alpha & \alpha \cos q \\ \sin q & \cos q \cos \alpha & -\cos q \sin \alpha & \alpha \sin q \\ 0 & \sin \alpha & \cos \alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Dựa vào bảng Denavit-Hartenberg (D-H) đã lập thì ta sẽ thay 4 thông số vào công thức ma trận suy rộng ${}^{i-1}_i A$ (2.1) để xác định các ma trận suy rộng của từng khâu, cụ thể như sau:

- Ma trận suy rộng khâu 1:

$${}^0_1 A = \begin{bmatrix} \cos q_1 & 0 & -\sin q_1 & 0 \\ \sin q_1 & 0 & \cos q_1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

- Ma trận suy rộng khâu 2:

$${}^1_2A = \begin{bmatrix} \cos q_2 & -\sin q_2 & 0 & l_2 \cos q_2 \\ \sin q_2 & \cos q_2 & 0 & l_2 \sin q_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

- Ma trận suy rộng khâu 3:

$${}^2_3A = \begin{bmatrix} \cos q_3 & -\sin q_3 & 0 & l_3 \cos q_3 \\ \sin q_3 & \cos q_3 & 0 & l_3 \sin q_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

Vậy, ta đã xác định được các ma trận suy rộng A của từng khâu thể hiện hướng và vị trí giữa các khâu với nhau của cánh tay robot 3 bậc tự do.

2.2.4. Xác định ma trận vecto cuối T theo các ma trận A_i

Sau khi đã xác định được các ma trận suy rộng biến đổi giữa các hệ tọa độ kế tiếp nhau theo phương pháp D-H [1] thì tiếp theo là xác định ma trận chuyển đổi tổng thể từ hệ tọa độ gốc đến hệ tọa độ của khâu chấp hành gọi là ma trận T'_d , là tích các ma trận suy rộng (2.2) (2.3) (2.4). Ma trận chuyển đổi tổng thể T'_d mô tả hướng và vị trí của hệ tọa độ gắn với khâu chấp hành so với hệ tọa độ gốc. Với cánh tay robot 3 bậc tự do thì ma trận T'_d sẽ có dạng như sau:

$${}^0_3T = {}^0_1A \times {}^1_2A \times {}^2_3A = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$= \begin{bmatrix} \cos q_1 \cos(q_2 + q_3) & -\cos q_1 \sin(q_2 + q_3) & -\sin q_1 & \cos q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \\ \sin q_1 \cos(q_2 + q_3) & -\sin q_1 \sin(q_2 + q_3) & \cos q_1 & \sin q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \\ -\sin(q_2 + q_3) & -\cos(q_2 + q_3) & 0 & l_1 - l_2 \sin q_2 - l_3 \sin(q_2 + q_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cuối cùng ta sẽ xác định được vị trí điểm cuối từ ma trận (2.5) như sau:

$$p_x = \cos q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \quad (2.6)$$

$$p_y = \sin q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \quad (2.7)$$

$$p_z = l_1 - l_2 \sin q_2 - l_3 \sin(q_2 + q_3) \quad (2.8)$$

Sau khi chúng ta đã xây dựng được ma trận 3_0T thể hiện hướng và vị trí của hệ tọa độ khâu chấp hành so với hệ tọa độ gốc từ các ma trận suy rộng A thì ta sẽ có được vị trí điểm cuối của khâu chấp hành so với hệ tọa độ gốc, từ đó giúp ta có dữ liệu đầu vào để tiếp tục tính toán động học nghịch cho cánh tay Robot.

2.3. Động học nghịch cánh tay Robot 3 bậc tự do

Tiếp theo, chúng ta sẽ nghiên cứu bài toán động học nghịch cho cánh tay robot 3 bậc tự do sau khi đã xác định vị trí và hướng của khâu chấp hành. Động học nghịch là quá trình ngược lại với động học thuận, tìm ra các tham số điều khiển của robot (như góc quay) khi đã biết trước vị trí và hướng mong muốn của khâu chấp hành. Bài toán động học nghịch có thể sẽ phức tạp hơn so với động học thuận do có thể tồn tại nhiều nghiệm, vô nghiệm hoặc các nghiệm kỳ dị. Đối với cánh tay Robot 3 bậc tự do ta có thể sử dụng phương pháp đại số dựa vào cách giải “Bài toán động học ngược của Robot Elbow” để tính toán động học nghịch một cách đơn giản [1].

Theo phương pháp đại số, sau khi tính toán động học thuận và xác định được ma trận T'_d mô tả hướng và vị trí của hệ tọa độ gắn khâu chấp hành so với tọa độ gốc thì ta có được ma trận dạng 3×1 xác định vị trí của khâu chấp hành:

$$p = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} \cos q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \\ \sin q_1 (l_3 \cos(q_2 + q_3) + l_2 \cos q_2) \\ l_1 - l_2 \sin q_2 - l_3 \sin(q_2 + q_3) \end{bmatrix}$$

- Tính toán góc quay q_1 :

- Từ phương trình (2.6) và (2.7) ta có:

$$\frac{p_y}{p_x} = \frac{\sin q_1}{\cos q_1} \Rightarrow q_1 = \arctan 2(p_y, p_x)$$

- Tính toán góc quay q_3 :

- Nhân $\cos q_1$ cho (2.6) và nhân $\sin q_1$ cho (2.7) ta được:

$$p_x \cos q_1 = \cos^2 q_1 (l_2 \cos q_2 + l_3 \cos(q_2 + q_3))$$

$$p_y \sin q_1 = \sin^2 q_1 (l_2 \cos q_2 + l_3 \cos(q_2 + q_3))$$

- Cộng vế theo vế ta được:

$$p_x \cos q_1 + p_y \sin q_1 = l_2 \cos q_2 + l_3 \cos(q_2 + q_3)$$

- Từ phương trình (2.8) ta có:

$$l_1 - p_z = l_2 \sin q_2 + l_3 \sin(q_2 + q_3)$$

- Đặt $E = p_x \cos q_1 + p_y \sin q_1$, ta được:

$$E = l_2 \cos q_2 + l_3 \cos(q_2 + q_3) \quad (2.9)$$

$$l_1 - p_z = l_2 \sin q_2 + l_3 \sin(q_2 + q_3) \quad (2.10)$$

- Bình phương hai vế của (2.9) và (2.10), ta được:

$$E^2 = l_2^2 \cos^2 q_2 + 2l_2l_3 \cos q_2 \cos(q_2 + q_3) + l_3^2 \cos^2(q_2 + q_3)$$

$$l_1^2 - 2l_1p_z + p_z^2 = l_2^2 \sin^2 q_2 + 2l_2l_3 \sin q_2 \sin(q_2 + q_3) + l_3^2 \sin^2(q_2 + q_3)$$

- Cộng 2 vế của 2 phương trình, ta được:

$$E^2 + l_1^2 - 2l_1p_z + p_z^2 = l_2^2 + 2l_2l_3 (\cos q_2 \cos(q_2 + q_3) + \sin q_2 \sin(q_2 + q_3)) + l_3^2$$

$$\Leftrightarrow E^2 + l_1^2 - 2l_1z_d + z_d^2 - l_2^2 - l_3^2 = 2l_2l_3 \cos q_3$$

- Ta đặt: $F = E^2 + l_1^2 - 2l_1z_d + z_d^2 - l_2^2 - l_3^2$

$$\Rightarrow F = 2l_2l_3 \cos q_3$$

$$\Rightarrow \cos q_3 = \frac{F}{2l_2l_3} \Rightarrow \sin q_3 = \pm \sqrt{1 - \cos^2 q_3}$$

$$\text{Vậy } q_3 = \begin{cases} \arctan 2(\sin q_3, \cos q_3) \\ \arctan 2(-\sin q_3, \cos q_3) \end{cases}$$

- Tính toán góc quay q_2 :

- Theo công thức lượng giác, ta có:

$$\cos(q_2 + q_3) = \cos q_2 \cos q_3 - \sin q_2 \sin q_3$$

- Thay công thức lượng giác vào phương trình (2.9), ta có:

$$E = l_2 \cos q_2 + l_3 \cos q_2 \cos q_3 - l_3 \sin q_2 \sin q_3$$

$$\Leftrightarrow E = \cos q_2 (l_2 + l_3 \cos q_3) - l_3 \sin q_2 \sin q_3 \quad (2.11)$$

- Tương tự với công thức lượng giác:

$$\sin(q_2 + q_3) = \sin q_2 \times \cos q_3 + \cos q_2 \times \sin q_3$$

- Thay vào phương trình (2.10), ta có:

$$l_1 - p_z = l_2 \sin q_2 + l_3 \sin q_2 \cos q_3 + l_3 \cos q_2 \sin q_3$$

$$\Leftrightarrow l_1 - p_z = \sin q_2 (l_2 + l_3 \cos q_3) + l_3 \cos q_2 \sin q_3 \quad (2.12)$$

- Rút $\cos q_2$ từ phương trình (2.11), ta được:

$$\cos q_2 = \frac{E + l_3 \sin q_2 \sin q_3}{l_2 + l_3 \cos q_3} \quad (2.13)$$

- Rút $\sin q_2$ từ phương trình (2.12), ta được;

$$\sin q_2 = \frac{l_1 - p_z - l_3 \cos q_2 \sin q_3}{l_2 + l_3 \cos q_3} \quad (2.14)$$

- Thay $\sin q_2$ ở phương trình (2.14) vào phương trình (2.13), ta được:

$$\begin{aligned} \cos q_2 &= \frac{E + l_3 \sin q_3 \frac{l_1 - p_z - l_3 \cos q_2 \sin q_3}{l_2 + l_3 \cos q_3}}{l_2 + l_3 \cos q_3} \\ \Leftrightarrow \cos q_2 &= \frac{E(l_2 + l_3 \cos q_3) + l_3 \sin q_3 (l_1 - p_z - l_3 \cos q_2 \sin q_3)}{(l_2 + l_3 \cos q_3)^2} \\ \Leftrightarrow \cos q_2 &= \frac{E(l_2 + l_3 \cos q_3) + l_3 \sin q_3 (l_1 - p_z)}{(l_2 + l_3 \cos q_3)^2 + l_3^2 \sin^2 q_3} \end{aligned}$$

- Thay $\cos q_2$ ở phương trình (2.13) vào phương trình (2.14), ta được:

$$\begin{aligned} \sin q_2 &= \frac{l_1 - p_z - l_3 \sin q_3 \frac{E + l_3 \sin q_2 \sin q_3}{l_2 + l_3 \cos q_3}}{l_2 + l_3 \cos q_3} \\ \Leftrightarrow \sin q_2 &= \frac{(l_1 - p_z)(l_2 + l_3 \cos q_3) - l_3 \sin q_3 (E + l_3 \sin q_2 \sin q_3)}{(l_2 + l_3 \cos q_3)^2} \\ \Leftrightarrow \sin q_2 &= \frac{(l_1 - p_z)(l_2 + l_3 \cos q_3) - l_3 \sin q_3 E}{(l_2 + l_3 \cos q_3)^2 + l_3^2 \sin^2 q_3} \end{aligned}$$

- Vậy: $\Rightarrow q_2 = \arctan 2(\sin q_2, \cos q_2)$

Sau khi tính toán động học nghịch bằng phương pháp đại số dựa vào cách giải “Bài toán động học ngược của Robot Elbow” thì chúng ta thu được các góc quay khi biết trước vị trí của khâu chấp hành so với góc tọa độ, như sau:

$$\begin{aligned} q_1 &= \arctan 2(p_y, p_x) \\ q_2 &= \arctan 2(\sin q_2, \cos q_2) \\ q_3 &= \begin{cases} \arctan 2(\sin q_3, \cos q_3) \\ \arctan 2(-\sin q_3, \cos q_3) \end{cases} \end{aligned}$$

Kết quả ở đây ta thấy có 2 giá trị của góc quay q_3 , việc lựa chọn góc quay q_3 sẽ phụ thuộc vào việc chúng ta thực nghiệm mô phỏng động học nghịch trên phần mềm Matlab Simulink. Bằng cách đưa các giá trị đặt góc quay vào khối động học thuận từ đó

chúng ta sẽ được các giá trị tọa độ x y z của khâu chấp hành, tiếp theo ta đưa các giá trị hệ tọa độ đó vào cho khối động học nghịch thì kết quả chúng ta sẽ nhận được nhiều tập nghiệm góc quay khác nhau và ta sẽ lựa chọn được góc quay q_3 nào có giá trị trùng với góc quay q_3 đặt ở đầu vào của khối động học thuận thì ta sẽ chọn công thức góc quay q_3 đó để đưa vào mô phỏng cánh tay Robot 3 bậc tự do.

Sau khi đã xác định được vị trí khâu chấp hành từ động học thuận dựa trên các góc quay từ cánh tay Robot và xác định được góc quay của các khớp từ động học nghịch dựa trên vị trí điểm đặt mong muốn của khâu chấp hành để mô phỏng quỹ đạo cánh tay Robot, thì tiếp theo chúng ta cần xác định mô hình động lực học cho cánh tay Robot 3 bậc tự do.

2.4. Xác định mô hình động lực học của cánh tay Robot 3 bậc tự do

Sau khi đã tính toán động học thuận và động học nghịch, bước tiếp theo trong việc mô hình hóa cánh tay robot là xây dựng động lực học. Trong khi động học mô tả mối quan hệ giữa chuyển động đầu ra và biến trạng thái (vị trí, vận tốc, gia tốc) của robot thì động lực học lại mô tả mối quan hệ giữa momen tác động (T) và các biến chuyển động, mô tả chuyển động thực của Robot từ đó đánh giá độ sai lệch trong quá trình điều khiển.

Mục tiêu của việc xây dựng mô hình động lực học là xác định mối quan hệ giữa các momen điều khiển (T) tác động vào các khớp và các chuyển động tạo ra (góc, vận tốc, gia tốc) của các khâu.

Để xây dựng mô hình động lực học cho cánh tay Robot 3 bậc tự do, ta thường sử dụng một trong hai phương pháp phổ biến:

- Phương pháp Lagrange – Euler.
- Phương pháp Newton – Euler.

Trong hai phương pháp phổ biến, phương pháp Lagrange – Euler thường được ưu tiên sử dụng để xây dựng mô hình động lực học cho cánh tay robot vì tính hệ thống và dễ dàng mở rộng cho các cơ cấu phức tạp. Phương pháp này dựa trên năng lượng động học và thế năng, giúp loại bỏ trực tiếp các lực ràng buộc và không phụ thuộc vào hệ tọa độ tham chiếu, từ đó giảm thiểu sai sót trong quá trình mô hình hóa. Ngược lại, phương pháp Newton – Euler mặc dù hiệu quả trong tính toán động lực học ngược và phù hợp với các hệ thống có cấu trúc đơn giản, nhưng thường phức tạp hơn khi áp dụng cho các hệ nhiều bậc tự do do phải xử lý lực và mô men tác động chi tiết từng khâu một cách tuần tự. Vì vậy, Lagrange – Euler là lựa chọn phổ biến hơn trong nghiên cứu và thiết kế điều khiển robot đa bậc tự do [4]. Trong đề án này, nhóm chúng em sử dụng phương pháp Lagrange – Euler [1] (gọi tắt là phương pháp Lagrange) để xây dựng mô hình động lực học cánh tay Robot 3 bậc tự do. Phương pháp Lagrange bắt đầu triển khai bằng cách

xác định năng lượng của hệ thống bằng hàm Lagrange, được định nghĩa như là hiệu của tổng động năng và tổng thế năng của hệ:

$$L = K - P \quad (2.15)$$

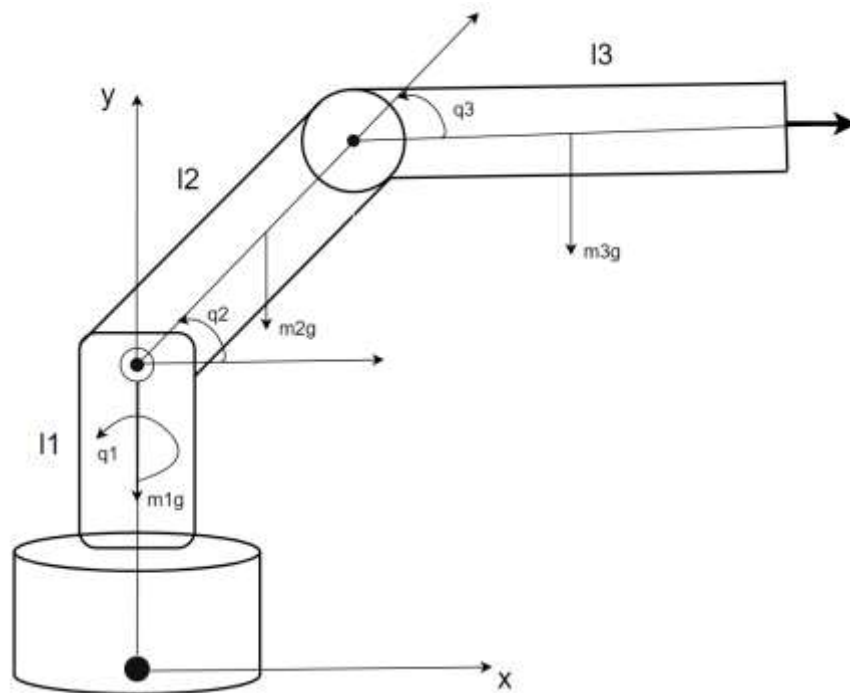
Trong đó:

K - Tổng động năng của hệ do chuyển động quay và tịnh tiến của các khâu.

P - Tổng thế năng của hệ do ảnh hưởng của trọng lực lên các khâu.

Để xác định được tổng động năng K và tổng thế năng P của hệ, ta tiến hành phân tích động năng và thế năng của từng khâu.

Ta đặt cánh tay robot vào mặt phẳng và góc tọa độ được đặt ở đế cánh tay robot, như hình 2.7.



Hình 2.7: Cánh tay Robot 3 bậc tự do trên mặt phẳng 2D

Các bước xây dựng mô hình động lực học cánh tay Robot 3 bậc tự do dựa trên việc xây dựng hàm Lagrange như sau.

- Bước 1: Xác định vị trí trọng tâm của các khâu

- Khâu 1: $\begin{cases} x_1 = 0 \\ y_1 = \frac{l_1}{2} \end{cases}$
- Khâu 2: $\begin{cases} x_2 = \frac{l_2}{2} \cos q_2 \\ y_2 = l_1 + \frac{l_2}{2} \sin q_2 \end{cases}$

- Khâu 3:
$$\begin{cases} x_3 = l_2 \cos q_2 + \frac{l_3}{2} \cos(q_2 + q_3) \\ y_3 = l_1 + l_2 \sin q_2 + \frac{l_3}{2} \sin(q_2 + q_3) \end{cases}$$

- Bước 2: Xác định vận tốc của trọng tâm từng khâu bằng cách lấy đạo hàm theo thời gian của các tọa độ ở bước 1 thì ta sẽ thu được các thành phần vận tốc

- Khâu 1:
$$\begin{cases} V_{x_1} = \dot{x}_1 = 0 \\ V_{y_1} = \dot{y}_1 = 0 \end{cases}$$

- Khâu 2:
$$\begin{cases} V_{x_2} = \dot{x}_2 = -\frac{l_2}{2} \dot{q}_2 \sin q_2 \\ V_{y_2} = \dot{y}_2 = \frac{l_2}{2} \dot{q}_2 \cos q_2 \end{cases}$$

- Khâu 3:
$$\begin{cases} V_{x_3} = \dot{x}_3 = -l_2 \dot{q}_2 \sin q_2 - \frac{l_3}{2} (\dot{q}_2 + \dot{q}_3) \sin(q_2 + q_3) \\ V_{y_3} = \dot{y}_3 = l_2 \dot{q}_2 \cos q_2 + \frac{l_3}{2} (\dot{q}_2 + \dot{q}_3) \cos(q_2 + q_3) \end{cases}$$

- Bước 3: Tính động năng của cánh tay Robot 3 bậc tự do

- Khâu 1 chỉ có động năng quay quanh trục với momen quán tính $I_1 = \frac{1}{2} m_1 R^2$ [2]

còn động năng tịnh tiến bằng 0 do khâu 1 không có vận tốc dọc trục. Vậy tổng động năng khâu 1 là:

$$\begin{aligned} K_1 &= \frac{1}{2} I_1 (\dot{q}_1)^2 + \frac{1}{2} m_1 (V_{x_1} + V_{y_1})^2 \\ &= \frac{1}{2} \left(\frac{1}{2} m_1 R^2 \right) (\dot{q}_1)^2 + 0 = \frac{1}{4} m_1 R^2 (\dot{q}_1)^2 \end{aligned}$$

- Khâu 2 có trọng tâm nằm ở giữa thanh nên có sẽ có động năng quay quanh trục z với momen quán tính $I_2 = \frac{1}{12} m_2 l_2^2$ [2] và động năng tịnh tiến dọc trục x có

công thức $K = \frac{1}{2} m_2 V_2^2$. Vậy tổng động năng khâu 2 là:

$$\begin{aligned}
 K_2 &= \frac{1}{2} I_2 \left(\dot{q}_2 \right)^2 + \frac{1}{2} m_2 V_2^2 \\
 &= \frac{1}{2} \left(\frac{1}{12} m_2 l_2^2 \right) \left(\dot{q}_2 \right)^2 + \frac{1}{2} m_2 \left(V_{x_2}^2 + V_{y_2}^2 \right) \\
 &= \frac{1}{2} \frac{1}{12} m_2 l_2^2 \left(\dot{q}_2 \right)^2 + \frac{1}{2} m_2 \frac{l_2^2}{4} \left(\dot{q}_2 \right)^2 \\
 &= \frac{1}{24} m_2 l_2^2 \left(\dot{q}_2 \right)^2 + \frac{1}{8} m_2 l_2^2 \left(\dot{q}_2 \right)^2 \\
 &= \frac{1}{6} m_2 l_2^2 \left(\dot{q}_2 \right)^2
 \end{aligned}$$

- Khâu 3 có trọng tâm nằm ở giữa thanh nên có sẽ có động năng quay quanh trục z với momen quán tính $I_3 = \frac{1}{12} m_3 l_3^2$ [2] và động năng tịnh tiến dọc trục x có công thức $K = \frac{1}{2} m_3 V_3^2$. Vậy tổng động năng khâu 3 là:

$$\begin{aligned}
 K_3 &= \frac{1}{2} I_3 \left(\dot{q}_2 + \dot{q}_3 \right)^2 + \frac{1}{2} m_3 \left(V_{x_3}^2 + V_{y_3}^2 \right) \\
 &= \frac{1}{2} \frac{1}{12} m_3 l_3^2 \left(\dot{q}_2 + \dot{q}_3 \right)^2 + \frac{1}{2} m_3 \left[\left(-l_2 \left(\dot{q}_2 \right) \sin q_2 - \frac{l_3}{2} \left(\dot{q}_2 + \dot{q}_3 \right) \sin \left(q_2 + q_3 \right) \right)^2 \right. \\
 &\quad \left. + \left(l_2 \left(\dot{q}_2 \right) \cos q_2 + \frac{l_3}{2} \left(\dot{q}_2 + \dot{q}_3 \right) \cos \left(q_2 + q_3 \right) \right)^2 \right] \\
 &= \frac{1}{24} m_3 l_3^2 \left(\dot{q}_2 + \dot{q}_3 \right)^2 + \frac{1}{2} m_3 l_2^2 \left(\dot{q}_2 \right)^2 + \frac{1}{8} m_3 l_3^2 \left(\dot{q}_2 + \dot{q}_3 \right)^2 \\
 &\quad + \frac{1}{2} m_3 l_2 l_3 \left(\dot{q}_2 \right) \left(\dot{q}_2 + \dot{q}_3 \right) \cos q_3 \\
 &= \frac{1}{6} m_3 l_3^2 \left(\dot{q}_2 + \dot{q}_3 \right)^2 + \frac{1}{2} m_3 l_3^2 \left(\dot{q}_2 \right)^2 + \frac{1}{2} m_3 l_2 l_3 \left(\dot{q}_2 \right) \left(\dot{q}_2 + \dot{q}_3 \right) \cos q_3
 \end{aligned}$$

- Bước 4: Tính thế năng của cánh tay Robot 3 bậc tự do, thế năng do trọng lực tác động lên từng khâu, giả sử thế năng tại mặt phẳng đáy bằng 0 khi $y = 0$
 - Thế năng khâu 1 do tác động của trọng lực khi ở độ cao $h = y_1$:

$$p_1 = m_1 g y_1 = m_1 g \frac{l_1}{2}$$

- Thế năng khâu 2 do tác động của trọng lực khi ở độ cao $h = y_2$:

$$\begin{aligned} p_2 &= m_2 g y_2 = m_2 g \left(l_1 + \frac{l_2}{2} \sin q_2 \right) \\ &= m_2 g l_1 + \frac{1}{2} m_2 g l_2 \sin q_2 \end{aligned}$$

- Thế năng khâu 3 do tác động của trọng lực khi ở độ cao $h = y_3$:

$$\begin{aligned} p_3 &= m_3 g y_3 = m_3 g \left(l_1 + l_2 \sin q_2 + \frac{l_3}{2} \sin(q_2 + q_3) \right) \\ &= m_3 g l_1 + m_3 g l_2 \sin q_2 + \frac{1}{2} m_3 g l_3 \sin(q_2 + q_3) \end{aligned}$$

- Bước 5: Sau khi tính được tổng động năng và tổng thế năng của Robot thì ta sẽ có được hàm Lagrange (2.15) để tiếp tục xây dựng mô hình động lực học cho Robot:

$$\begin{aligned} L &= \sum K - \sum P \\ &= K_1 + K_2 + K_3 - P_1 - P_2 - P_3 \\ \Rightarrow L &= \frac{1}{4} m_1 R^2 \left(\dot{q}_1 \right)^2 + \left(\frac{m_2}{6} + \frac{m_3}{3} \right) l_2^2 \left(\dot{q}_2 \right)^2 + \frac{1}{6} m_3 l_3^2 \left(\dot{q}_2 + \dot{q}_3 \right)^2 \\ &\quad + \frac{1}{2} m_3 l_2 l_3 \dot{q}_2 \left(\dot{q}_2 + \dot{q}_3 \right) \cos q_3 - \left(\frac{m_1}{2} + m_2 \right) g l_1 - m_3 g l_2 \\ &\quad - \left(\frac{m_2}{2} + m_3 \right) g l_2 \sin q_2 - \frac{1}{2} m_3 g l_3 \sin(q_2 + q_3) \end{aligned}$$

Để tiếp tục xây dựng mô hình động lực học cho cánh tay Robot 3 bậc tự do bằng phương pháp Lagrange [1] thì tiếp theo chúng ta cần xây dựng phương trình Lagrange để biểu diễn mối quan hệ giữa các biến động học (góc quay, vận tốc, gia tốc), năng lượng của hệ với momen điều khiển (T_i) tác động lên cánh tay Robot, được viết dưới dạng sau:

$$T_i = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} \quad (2.16)$$

- Bước 6: Tính đạo hàm hàm Lagrange L theo \dot{q}_i

$$\frac{\partial L}{\partial \dot{q}_1} = \frac{1}{2} m_1 R^2 \dot{q}_1$$

$$\frac{\partial L}{\partial q_2} = \left(\frac{m_2}{3} + m_3 \right) l_2^2 \dot{q}_2 + \frac{1}{3} m_3 l_3^2 (\dot{q}_2 + \dot{q}_3) + \frac{1}{2} m_3 l_2 l_3 (2 \dot{q}_2 + \dot{q}_3) \cos q_3$$

$$\frac{\partial L}{\partial q_3} = \frac{1}{3} m_3 l_3^2 (\dot{q}_2 + \dot{q}_3) + \frac{1}{2} m_3 l_2 l_3 \dot{q}_2 \cos q_3$$

- Bước 7: Tính đạo hàm $\frac{d}{dt} \left(\frac{\partial L}{\partial q_i} \right)$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial q_1} \right) = \frac{1}{2} m_1 R^2 \ddot{q}_1$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial q_2} \right) &= \left(\frac{m_2}{3} + m_3 \right) l_2^2 \ddot{q}_2 + \frac{1}{3} m_3 l_3^2 (\ddot{q}_2 + \ddot{q}_3) + \frac{1}{2} m_3 l_2 l_3 (2 \ddot{q}_2 + \ddot{q}_3) \cos q_3 \\ &\quad - \frac{1}{2} m_3 l_2 l_3 \left(2 \dot{q}_2 \dot{q}_3 + (\dot{q}_3)^2 \right) \sin q_3 \end{aligned}$$

$$\frac{d}{dt} \left(\frac{\partial L}{\partial q_3} \right) = \frac{1}{3} m_3 l_3^2 (\ddot{q}_2 + \ddot{q}_3) + \frac{1}{2} m_3 l_2 l_3 \dot{q}_2 \cos q_3 - \frac{1}{2} m_3 l_2 l_3 \dot{q}_2 \dot{q}_3 \sin q_3$$

- Bước 8: Tính đạo hàm hàm Lagrange L theo q_i

$$\frac{\partial L}{\partial q_1} = 0$$

$$\frac{\partial L}{\partial q_2} = - \left(\frac{m_2}{2} + m_3 \right) g l_2 \cos q_2 - \frac{1}{2} m_3 g l_3 \cos(q_2 + q_3)$$

$$\frac{\partial L}{\partial q_3} = - \frac{1}{2} m_3 l_2 l_3 \dot{q}_2 (\dot{q}_2 + \dot{q}_3) \sin q_3 - \frac{1}{2} m_3 g l_3 \cos(q_2 + q_3)$$

- Bước 9: Sau khi đã tính các đạo hàm có trong phương trình Lagrange thì ta có phương trình Lagrange của các khâu như sau:

- Phương trình Lagrange khâu 1, từ phương trình (2.16) ta có:

$$T_1 = \frac{1}{2} m_1 R^2 \dot{q}_1^2$$

- Phương trình Lagrange khâu 2, từ phương trình (2.16) ta có:

$$T_2 = \left(\frac{m_2}{3} + m_3 \right) l_2^2 \ddot{q}_2 + \frac{1}{3} m_3 l_3^2 \ddot{q}_2 + m_3 l_2 l_3 \ddot{q}_2 \cos q_3 + \frac{1}{3} m_3 l_3^2 \ddot{q}_3 + \frac{1}{2} m_3 l_2 l_3 \ddot{q}_3 \cos q_3 - m_3 l_2 l_3 \dot{q}_2 \dot{q}_3 \sin q_3 - \frac{1}{2} m_3 l_2 l_3 \left(\dot{q}_3 \right)^2 \sin q_3 + \left(\frac{m_2}{3} + m_3 \right) g l_2 \cos q_2 + \frac{1}{2} m_3 g l_3 \cos(q_2 + q_3)$$

- Phương trình Lagrange khâu 3, từ phương trình (2.16) ta có:

$$T_3 = \frac{1}{3} m_3 l_3^2 \ddot{q}_2 + \frac{1}{2} m_3 l_2 l_3 \ddot{q}_2 \cos q_3 + \frac{1}{3} m_3 l_3^2 \ddot{q}_3 + \frac{1}{2} m_3 l_2 l_3 \left(\dot{q}_2 \right)^2 \sin q_3 + \frac{1}{2} m_3 g l_3 \cos(q_2 + q_3)$$

Sau khi đã xác định được phương trình Lagrange biểu diễn mối quan hệ giữa các biến động học (góc quay, vận tốc, gia tốc), năng lượng của hệ với momen điều khiển (T_i) tác động lên cánh tay Robot, thì bước cuối cùng trong việc xây dựng mô hình động lực học cánh tay Robot 3 bậc tự do là tách phương trình Lagrange ra các ma trận riêng biệt: ma trận quán tính $M(q)$, lực Coriolis và ly tâm $C(q, \dot{q})$ và trọng lực $G(q)$ để tăng khả năng phân tích, mô phỏng, điều khiển và tối ưu hệ thống. Phương trình Lagrange biểu diễn động lực học cánh tay Robot sau khi tách riêng biệt ra các ma trận sẽ có dạng như sau:

$$T = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) \quad (2.17)$$

$$M = \begin{bmatrix} \frac{1}{2} m_1 R^2 & 0 & 0 \\ 0 & \left(\frac{m_2}{3} + m_3 \right) l_2^2 + \frac{1}{3} m_3 l_3^2 + m_3 l_2 l_3 \cos q_3 & \frac{1}{3} m_3 l_3^2 + \frac{1}{2} m_3 l_2 l_3 \cos q_3 \\ 0 & \frac{1}{3} m_3 l_3^2 + \frac{1}{2} m_3 l_2 l_3 \cos q_3 & \frac{1}{3} m_3 l_3^2 \end{bmatrix}$$

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -m_3 l_2 l_3 \dot{q}_3 \sin q_3 & -m_3 l_2 l_3 \dot{q}_2 \sin q_3 - \frac{1}{2} m_3 l_2 l_3 \dot{q}_3 \sin q_3 \\ 0 & \frac{1}{2} m_3 l_2 l_3 \dot{q}_2 \sin q_3 & 0 \end{bmatrix}$$

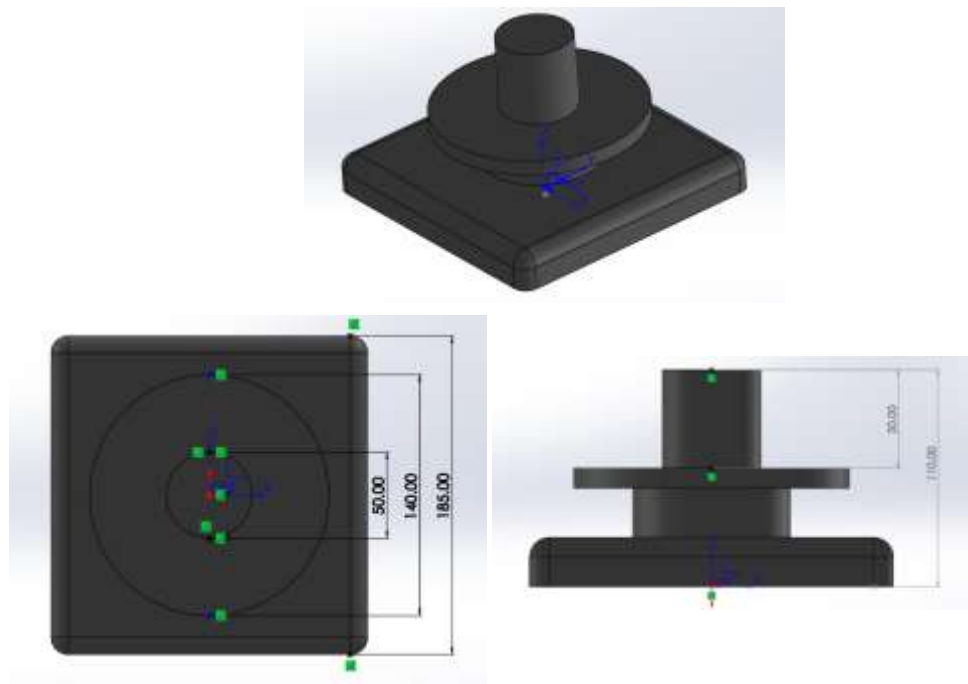
$$G = \begin{bmatrix} 0 \\ \left(\frac{m_2}{2} + m_3\right)gl_2 \cos q_2 + \frac{1}{2}m_3gl_3 \cos(q_2 + q_3) \\ \frac{1}{2}m_3gl_3 \cos(q_2 + q_3) \end{bmatrix}$$

Kết luận: Sau khi xây dựng phương trình Lagrange cho cánh tay Robot 3 bậc tự do thì ta thu được hệ phương trình động lực học phi tuyến (2.17), mô tả mối quan hệ giữa momen tác động (T) và chuyển động của các khớp. Dựa vào phương trình đó, chúng ta sẽ có thể mô phỏng chuyển động và quỹ đạo khi có momen đầu vào của mô hình động lực học cho cánh tay Robot 3 bậc tự do bằng phần mềm Matlab Simulink. Tiếp theo, ta sử dụng các phương pháp điều khiển như PID, ILC,... để giúp điều khiển chuyển động cho cánh tay Robot.

2.5. Thiết kế mô hình Robot 3DOF trên Solidworks

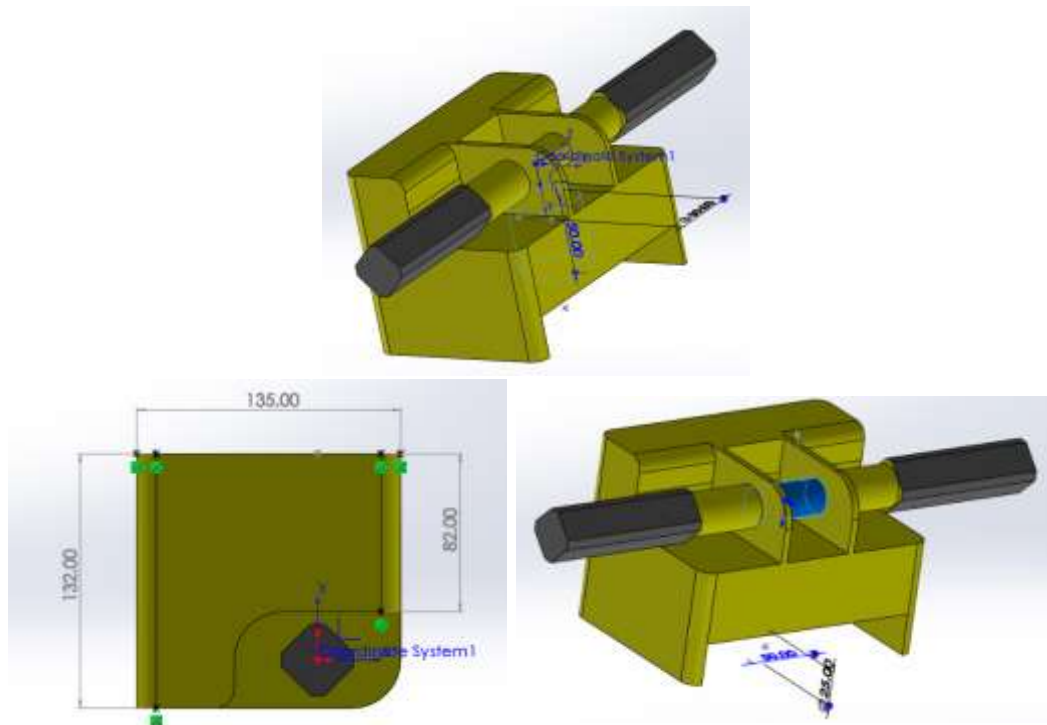
Sau khi đã xác định được mô hình động lực học cho cánh tay Robot 3 bậc tự do, chúng ta sẽ tiến hành thiết kế mô hình cánh tay Robot trên SolidWork để có thể so sánh với mô hình động lực học đã xác định. Mục đích thiết kế cánh tay Robot trên SolidWork là để đánh giá tính chính xác của mô hình động lực học vì mô hình trên SolidWork là mô hình gần với thực tế nhất.

Đầu tiên là để cánh tay robot được mô phỏng trên SolidWork với cấu trúc gần giống mô hình thực tế, như hình 2.8:



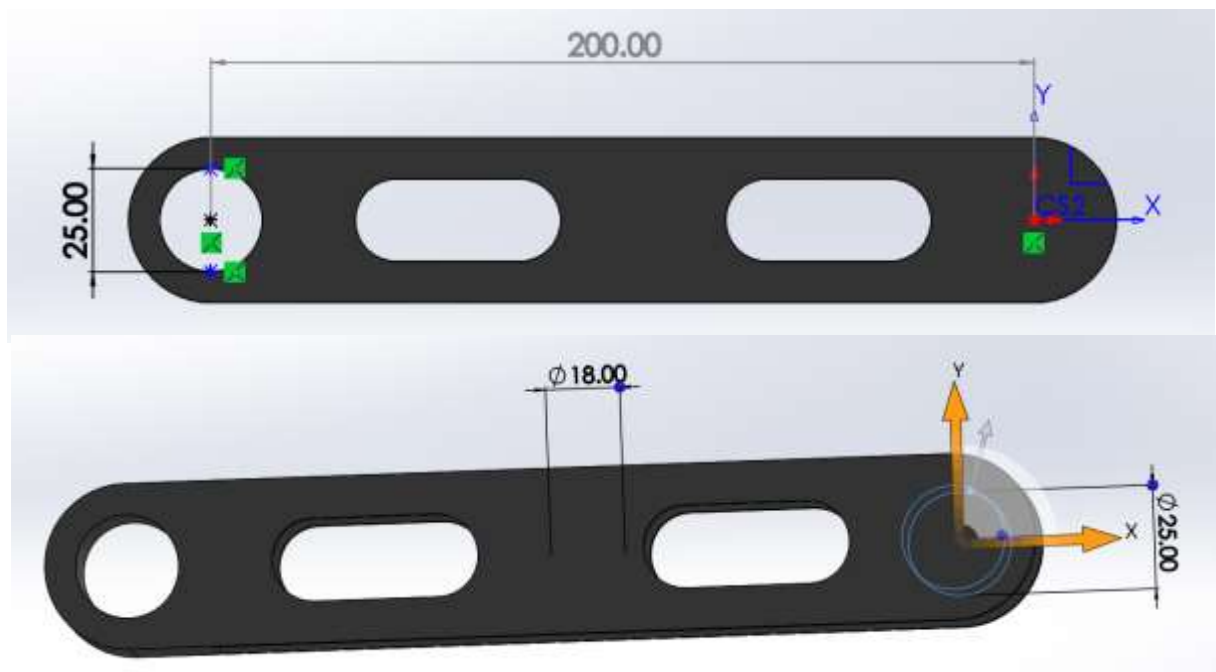
Hình 2.8: Cấu trúc đế cánh tay Robot 3DOF trên SolidWork

Tiếp theo là cấu trúc khâu 1 của cánh tay Robot gần giống mô hình thực tế được mô phỏng trên phần mềm SolidWork như hình 2.9:



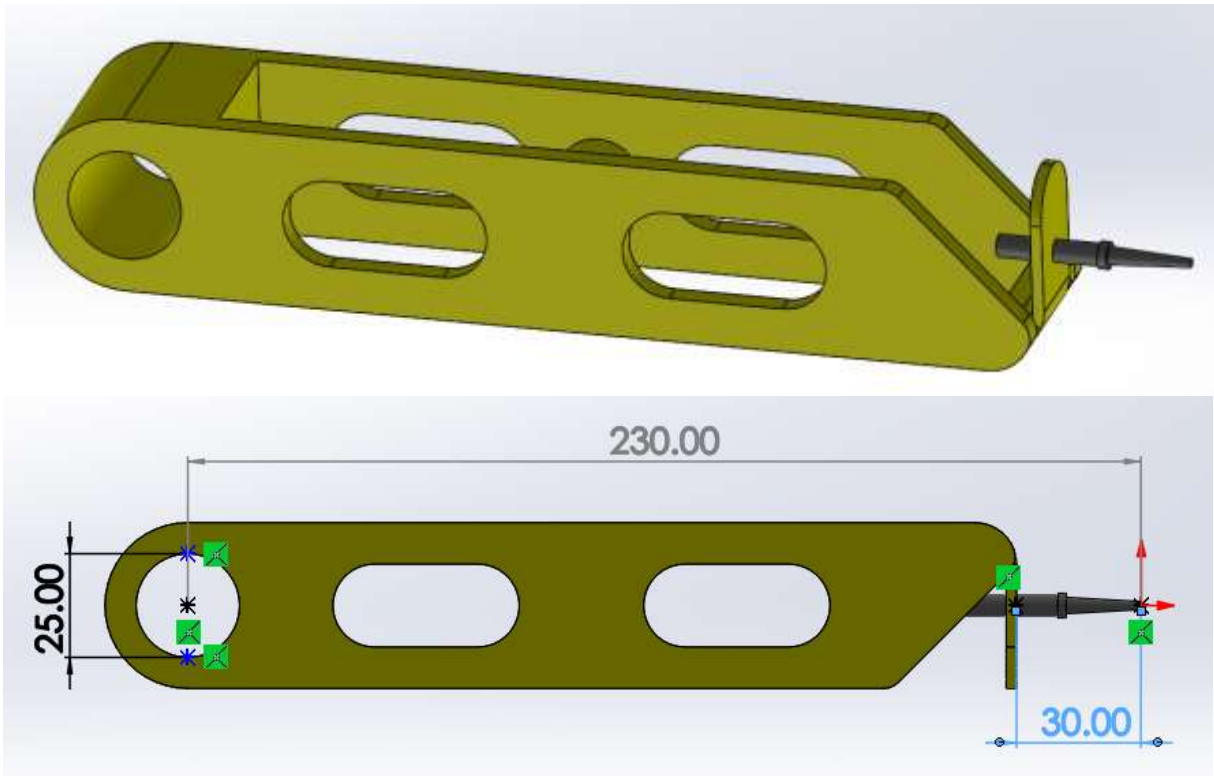
Hình 2.9: Cấu hình khâu 1 cánh tay Robot 3DOF trên SolidWork

Tiếp theo là cấu trúc khâu 2 của cánh tay Robot gần giống mô hình thực tế được mô phỏng trên phần mềm SolidWork như hình 2.10:



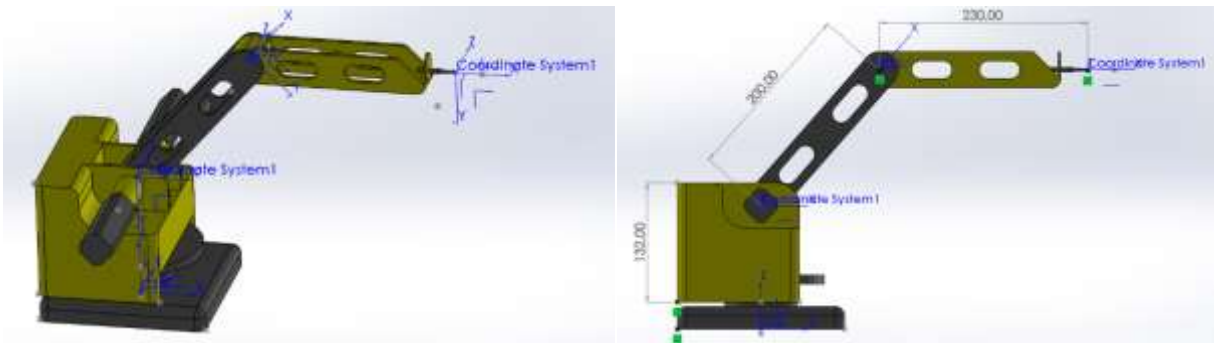
Hình 2.10: Cấu trúc khâu 2 cánh tay Robot 3DOF trên SolidWork

Cuối cùng là cấu trúc khâu 3 (khâu chấp hành) của cánh tay Robot gần giống mô hình thực tế được mô phỏng trên phần mềm SolidWork như hình 2.11:



Hình 2.11: Cấu trúc khâu 3 cánh tay Robot 3DOF trên SolidWork

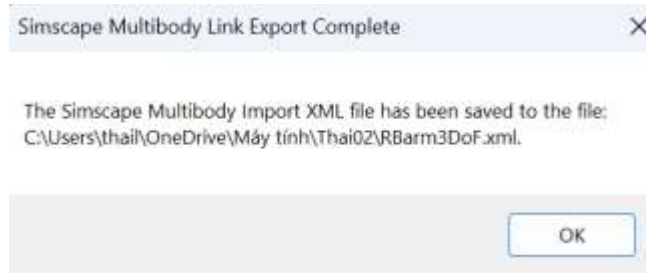
Cánh tay Robot 3 bậc tự do gần giống với mô hình thực tế sau khi kết hợp các khâu lại với nhau như hình 2.12:



Hình 2.12: Cấu trúc cánh tay Robot 3 bậc tự do trên SolidWork

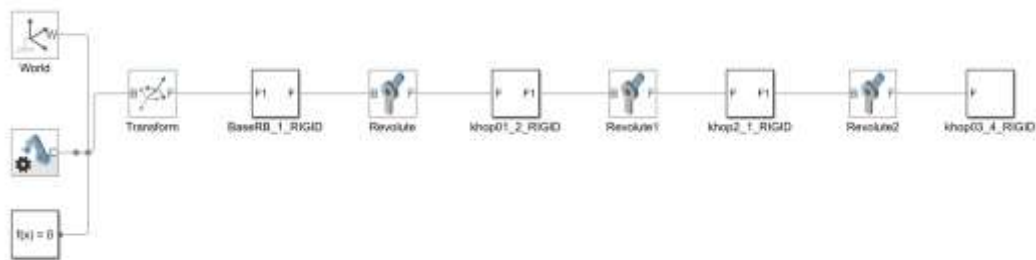
Sau khi hoàn tất kết nối các link của cánh tay Robot lại với nhau, mô hình cánh tay Robot trên Solidwork sẽ được chuyển qua phần mềm Matlab Simulink để tiến hành mô phỏng bằng cách sau:

- Trên thanh công cụ chọn Tools → Simscape Multibody link → Export → Simscape Multibody...
- Đặt tên và bấm Save, sau khi chuyển qua Matlab thành công sẽ có thông báo như hình 2.13 sau:



Hình 2.13: Thông báo chuyển mô hình từ SolidWork sang Matlab thành công

- Tiếp theo vào Matlab gõ lệnh `smimport('RBarm3DOF.xml')` vào Command Window để đưa mô hình cánh tay Robot vào Matlab Simulink. Sau đó, mô hình cánh tay Robot như hình 2.14 sẽ được tạo trên Matlab Simulink để tiến hành mô phỏng điều khiển.



Hình 2.14: Mô hình cánh tay Robot trên Solid khi được chuyển sang Matlab Simulink

2.6. Nhận dạng mô hình động cơ

Chúng ta đã xây dựng mô hình động lực học cánh tay Robot ở phần trước, bây giờ chúng ta tiến hành xây dựng mô hình động cơ cho mỗi khớp của cánh tay Robot 3 bậc tự do. Để xây dựng mô hình động cơ thì chúng ta cần dựa vào mô hình cánh tay Robot trong SolidWork để lấy các thông số phù hợp, vì ta đã nói ở phần trước là mô hình SolidWork là mô hình gần giống với thực tế nhất. Thực tế, mô hình SolidWork chỉ cung cấp cho ta cấu trúc vật lý của các khớp, nên mô hình động cơ cần được giả lập cho phù hợp với thực tế. Do đó, nhóm đề xuất dùng phương pháp nhận dạng mô hình để lựa chọn các thông số động cơ với giả thiết mô hình SolidWork là mô hình đối tượng.

Để thiết kế được mô hình cho động cơ thì chúng ta cần tìm được phương trình tốc độ của động cơ bằng cách sau:

Ta có 2 phương trình điện áp và momen của động cơ điện DC [1].

- Phương trình điện áp:

$$U(t) = L \frac{di}{dt} + R.i(t) + K_e \phi . \omega(t)$$

- Biến đổi sang miền S, ta được:

$$U(s) = Ls.I(s) + R.I(s) + K_e \phi . \omega(s)$$

$$\Rightarrow I(s) = \frac{U(s) - K_e \phi \omega(s)}{Ls + R} \quad (2.18)$$

- Phương trình momen:

$$J \frac{d\omega}{dt} = K_m \phi I - T_{load}$$

- Biến đổi sang miền S, ta được:

$$Js \cdot \omega(s) = K_m \phi I(s) - T_{load} \quad (2.19)$$

- Thay $I(s)$ của (2.18) vào (2.19), ta được:

$$Js \cdot \omega(s) = K_m \phi \frac{U(s) - K_e \phi \omega(s)}{Ls + R} - T_{load}$$

$$\Leftrightarrow \omega(s) = \frac{K_m \phi U(s) - K_e K_m \phi^2 \omega(s)}{Js(Ls + R)} - \frac{T_{load}}{Js}$$

$$\Leftrightarrow \omega(s) = \frac{K_m \phi U(s)}{JLs^2 + JRs + K_e K_m \phi^2} - \frac{T_{load}(Ls + R)}{JLs^2 + JRs + K_e K_m \phi^2} \quad (2.20)$$

- Xét động cơ của cánh tay Robot lúc không tải $T_{load} = 0$

$$\Rightarrow \omega(s) = \frac{U(s) K_m \phi}{JLs^2 + JRs + K_e K_m \phi^2} \quad (2.21)$$

- Dạng hàm truyền đạt của động cơ như sau:

$$\omega = \frac{Kd}{T_u T_c s^2 + T_c s + 1} \quad (2.22)$$

- So sánh 2 phương trình (2.21) và (2.22) ta được:

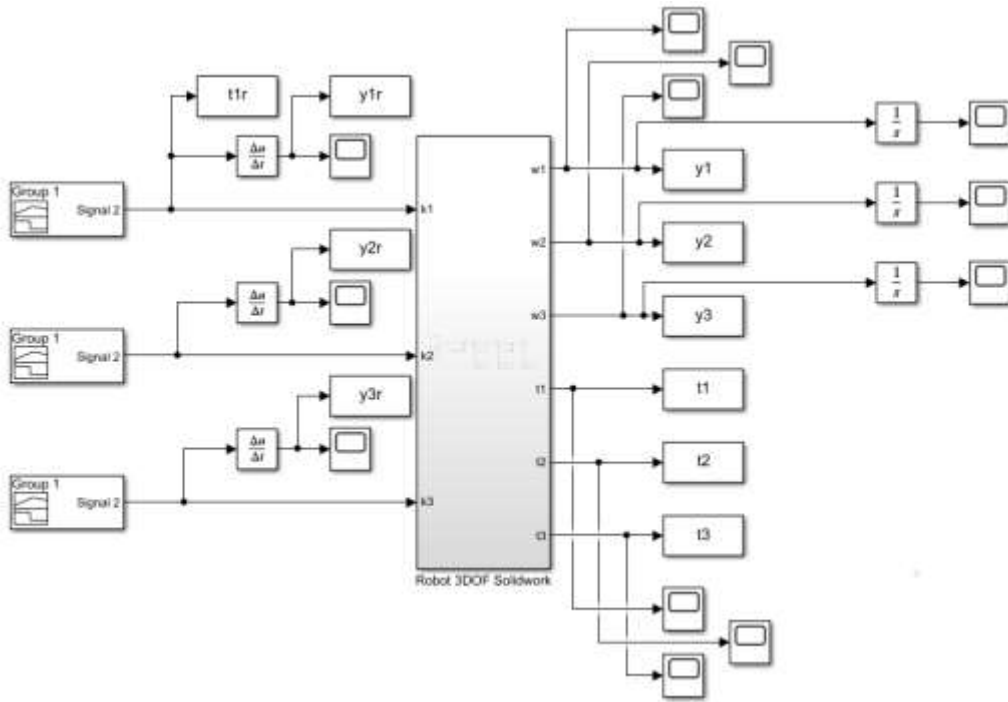
$$\Rightarrow T_u = \frac{L}{R}$$

$$\Rightarrow T_c = \frac{RJ}{K_e K_m \phi^2}$$

$$\Rightarrow K_d = \frac{1}{K_e \phi} = K_m \phi$$

Sau khi tìm được dạng phương trình tốc độ cho động cơ, ta tiến hành nhận dạng đối tượng động cơ từ mô hình SolidWork bằng Matlab Identification Toolbox.

Đầu tiên, ta lấy dữ liệu đầu vào và đầu ra của mô hình SolidWork trên Matlab Simulink:

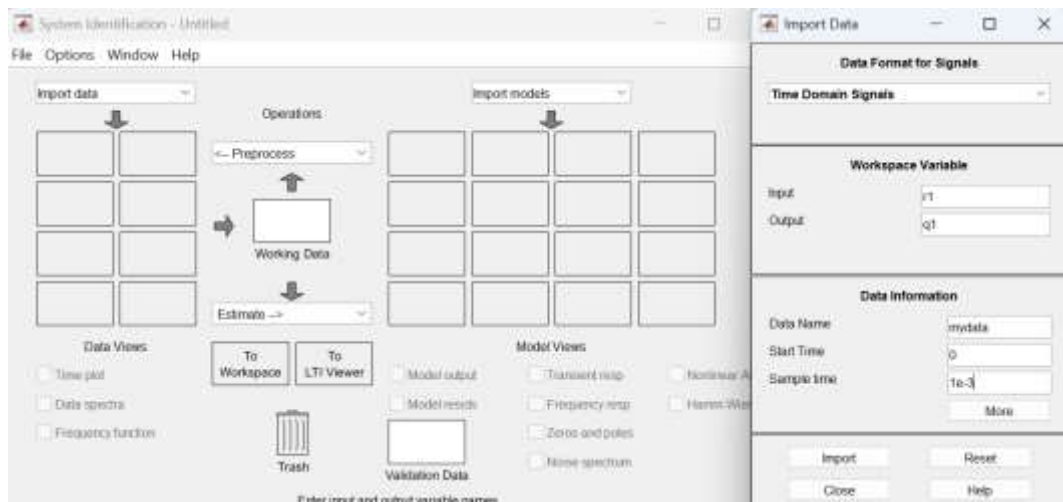


Hình 2.15: Mô hình lấy dữ liệu đầu vào và đầu ra từ mô hình SolidWork

Tiếp theo, ta lấy dữ liệu đã tìm được đưa vào Matlab Identification Toolbox để tìm hàm truyền đạt cho động cơ:

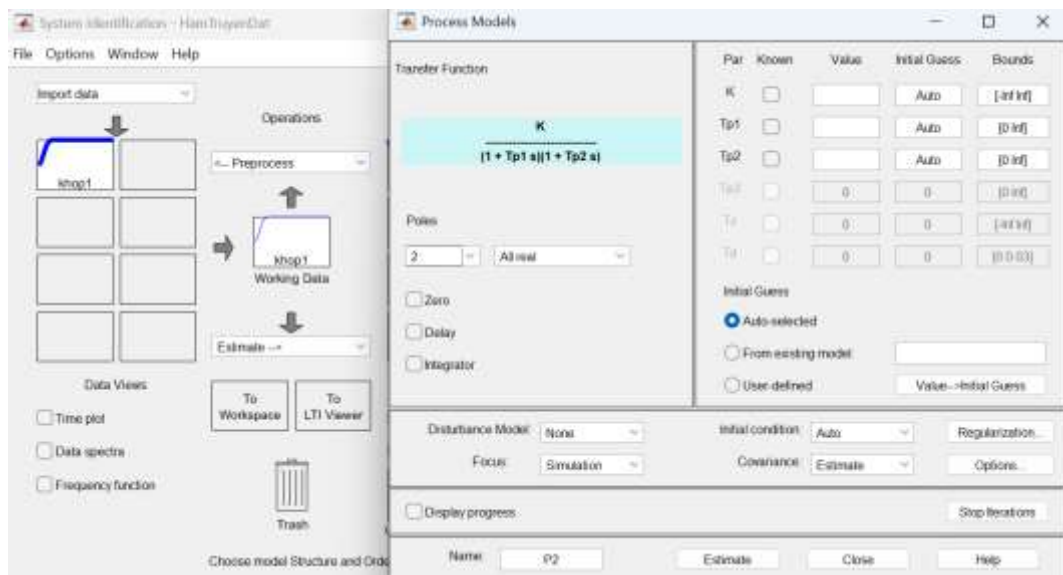
- Bước 1: Lưu dữ liệu dưới dạng ma trận 1 cột, bằng câu lệnh sau:

```
>> load('t1r.mat')  
>> r1 = ans(2,:)'  
>> load('t1.mat')  
>> q1 = ans(2,:)'  
>> save ident r1 q1
```
- Bước 2: Đưa dữ liệu vào Matlab Identification Toolbox:



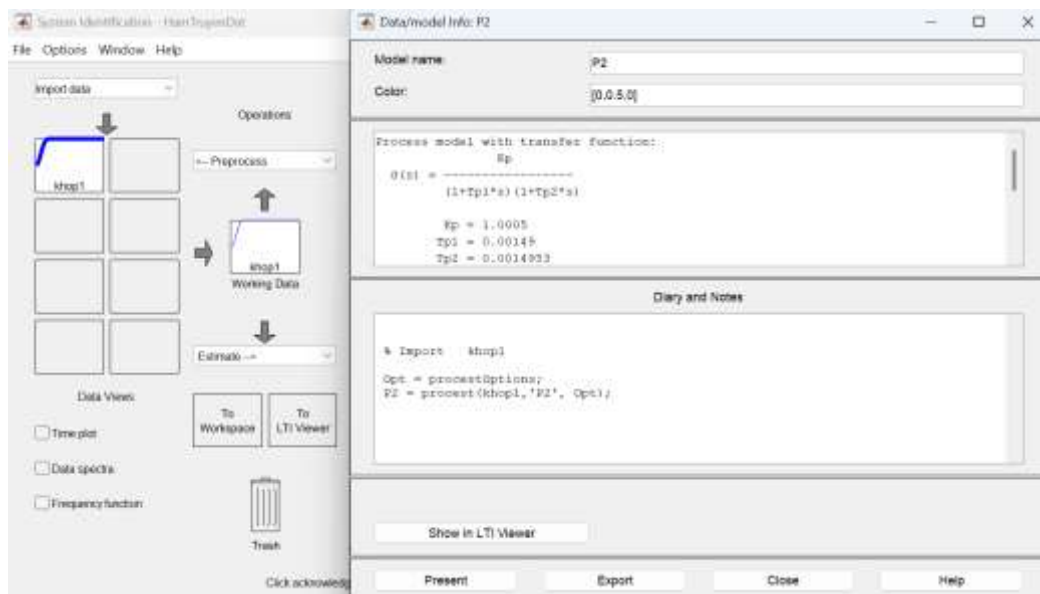
Hình 2.16: Giao diện nhập dữ liệu vào Matlab Identification Toolbox

- Bước 3: Chọn dạng hàm truyền giống với dạng hàm truyền tốc độ:



Hình 2.17: Giao diện nhận dạng hàm truyền đặt Matlab Identification Toolbox

- Bước 4: Lấy thông số hàm truyền đặt vừa nhận dạng được:



Hình 2.18: Giao diện kết quả hàm truyền đặt nhận dạng được từ Matlab Identification Toolbox

- Vậy ta có hàm truyền đặt của động cơ là:

$$G(s) = \frac{1.0005}{(1 + 0.00149s)(1 + 0.0014953s)}$$

$$\Leftrightarrow G(s) = \frac{1.0005}{2.228 \times 10^{-6} s^2 + 2.9853 \times 10^{-3} s + 1}$$

- Từ phương trình (2.22), ta có: $\omega = \frac{Kd}{T_u T_c s^2 + T_c s + 1}$

$$\Rightarrow K_d = \frac{1}{K_e \phi} = K_m \phi = 1.0005$$

$$\Rightarrow K_e \phi = 0.9995$$

$$T_c = 2.9853 \times 10^{-3} = \frac{RJ}{K_e K_m \phi^2} = RJ$$

$$T_u T_c = 2.228 \times 10^{-6}$$

$$\Rightarrow T_u = 7.4623 \times 10^{-4} = \frac{L}{R}$$

$$\Rightarrow \begin{cases} RJ = 2.9853 \times 10^{-3} \\ \frac{L}{R} = 7.4623 \times 10^{-4} \end{cases}$$

- Vì đây là hệ phương trình 3 ẩn 2 phương trình nên ta phải chọn thông số R của động cơ:

$$R = 0.05 (\Omega)$$

$$L = 3.71615 \times 10^{-5} (H)$$

$$J = 0.0597 (Kg.m^2)$$

- Vậy thông của động cơ tìm được là:

$$L = 3.71615 \times 10^{-5} (H)$$

$$R = 0.05 (\Omega)$$

$$J = 0.0597 (Kg.m^2)$$

$$K_e \phi = 0.9995$$

$$K_m \phi = 1.0005$$

Kết luận: Mô hình động cơ mà chúng ta xây dựng ở trên có thông số được lựa chọn dựa trên mô hình đối tượng cánh tay robot trong SolidWork đảm bảo gần giống với thực tế nhất nhờ sử dụng công cụ nhận dạng hàm truyền đạt đối tượng có trong phần mềm MATLAB (Matlab Identification Toolbox) để đưa ra hàm truyền đạt cho động cơ. Từ đó, chúng ta có thể mô phỏng điều khiển các khớp của cánh tay robot nhờ điều khiển mô hình động cơ gắn tại mỗi khớp để cánh tay robot có thể hoạt động theo các yêu cầu từ trước.

Chương 3: CÁC PHƯƠNG PHÁP ĐIỀU KHIỂN CÁNH TAY ROBOT

3.1. Tổng quan về điều khiển Robot

Dựa trên cơ sở phương trình động lực học của cánh tay Robot đã được xây dựng ở chương 2, thì mục tiêu chính của việc điều khiển cánh tay robot là đảm bảo cho hệ thống thực hiện chính xác các chuyển động theo quỹ đạo đã được thiết kế trước đó. Nói cách khác, các khớp của robot cần phải vận động một cách đồng bộ và chính xác để đầu cuối (end-effector) di chuyển đúng theo quỹ đạo mong muốn về cả vị trí và vận tốc theo thời gian. Lời giải cho bài toán điều khiển thường được trình bày khá phức tạp do ảnh hưởng của các yếu tố phi tuyến, lực quán tính, tương tác giữa các khâu, cũng như nhiễu từ môi trường xung quanh. Do đó, bài toán điều khiển Robot sẽ được chia thành hai nội dung chính:

- Xây dựng mô hình động lực học của Robot, phản ánh đầy đủ các yếu tố ảnh hưởng đến chuyển động của Robot, nội dung này đã được trình bày ở chương 2.
- Dựa trên mô hình này ta sẽ thiết kế bộ điều khiển nhằm đạt được đáp ứng đầu ra mong muốn một cách chính xác và ổn định.

Phân tích việc điều khiển thì quá trình điều khiển chuyển động của cánh tay Robot thường được chia thành hai giai đoạn chính. Giai đoạn đầu là giai đoạn điều khiển quỹ đạo, làm cho Robot đi từ vị trí này sang vị trí kia theo đúng quỹ đạo định trước. Giai đoạn thứ hai là giai đoạn điều khiển độ chính xác của chuyển động, trong giai đoạn này các tương tác động lực học của cánh tay Robot với quỹ đạo cho trước được xác định bằng thông tin thu từ cảm biến để đưa Robot đi đúng quỹ đạo mong muốn.

Trong công nghiệp thời này, có rất nhiều loại Robot được điều khiển bằng các cấu trúc đơn giản như là điều khiển từng khớp riêng lẻ của Robot mà không xét đến các yếu tố phi tuyến và tương tác giữa các khớp với nhau làm cho kết quả của việc điều khiển xuất hiện nhiều hạn chế, đặc biệt là với các bài toán yêu cầu độ chính xác cao hoặc quỹ đạo lặp lại. Đã có nhiều phương pháp được nghiên cứu và ứng dụng nhằm nâng cao độ chính xác và độ ổn định của hệ thống. Phương pháp điều khiển PID được sử dụng phổ biến nhờ tính đơn giản và hiệu quả trong các hệ thống tuyến tính cơ bản. Tuy nhiên, với các hệ thống phi tuyến như robot, PID thường không đủ mạnh để xử lý các tương tác giữa các khớp. Do đó, các phương pháp hiện đại như điều khiển tuyến tính hóa phản hồi, điều khiển trượt và điều khiển tối ưu như LQR hoặc MPC đã được phát triển nhằm cải thiện hiệu suất [6]. Đặc biệt, phương pháp điều khiển học lặp (ILC) cho phép robot cải thiện dần hiệu suất dựa trên sai số ở các chu kỳ trước, rất phù hợp với các bài toán có tính lặp lại như vẽ hình hoặc gấp-thả lặp. Bên cạnh đó, các phương pháp điều khiển

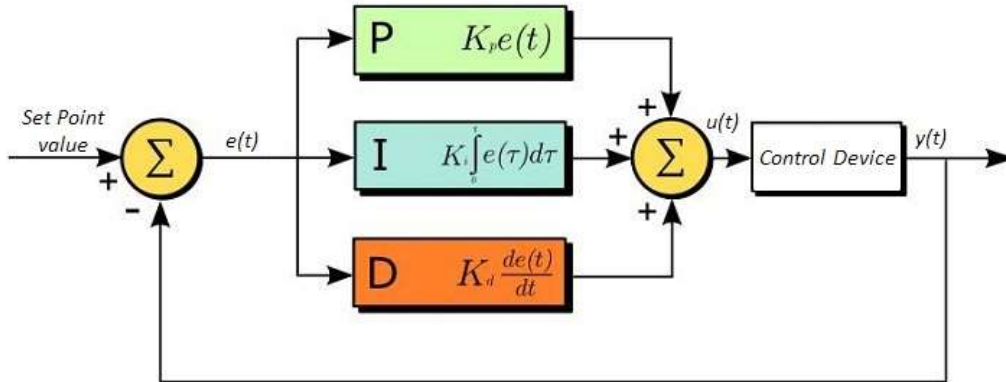
thông minh như mạng nơ-ron hay học tăng cường cũng đang được nghiên cứu để áp dụng trong các tình huống phức tạp hơn.

Để việc điều khiển đạt được chất lượng cao thì bộ điều khiển cần tính đầy đủ đến các yếu tố ảnh hưởng đến động lực học của hệ thống. Vì vậy, việc lựa chọn một bộ điều khiển phù hợp cho cánh tay Robot ở nghiên cứu này ảnh hưởng rất nhiều đến kết quả nghiên cứu, yếu tố then chốt là phải đảm bảo cho cánh tay Robot chuyển động chính xác và ổn định. Bộ điều khiển PID (Proportional - Integral - Derivative) được sử dụng phổ biến nhờ tính đơn giản, dễ triển khai và hiệu quả với nhiều hệ thống. Tuy nhiên, trong bài toán điều khiển cánh tay Robot có tính lặp lại này thì bộ điều khiển PID thường chưa đạt được hiệu suất tốt nhất. Do đó, chúng ta đề xuất thêm phương pháp điều khiển học lặp ILC (Iterative Learning Control - ILC), ILC cho phép hệ thống cải thiện hiệu suất sau mỗi lần học bằng cách sử dụng lỗi từ chu kỳ trước.

Từ đó, nghiên cứu lần này sẽ tập trung nghiên cứu và so sánh hai phương pháp điều khiển là PID và ILC để đánh giá khả năng đáp ứng của cánh tay Robot trong các điều kiện khác nhau (vẽ 2D hay in 3D).

3.2. Phương pháp điều khiển cánh tay Robot sử dụng bộ điều khiển PID

3.2.1. Tổng quan về bộ điều khiển PID



Hình 3.1: Sơ đồ khối bộ điều khiển PID

Bộ điều khiển PID (Proportional – Integral – Derivative) là một cơ chế phản hồi vòng điều khiển tổng quát được sử dụng rộng rãi trong các hệ thống công nghiệp, bao gồm cả hệ thống cánh tay Robot. Một bộ điều khiển PID tính toán một giá trị sai số là hiệu số giữa giá trị phản hồi đo được thực tế và giá trị đặt, từ đó PID điều chỉnh giá trị điều khiển đầu vào để giảm sai số theo thời gian [7]. Dạng toán học của bộ điều khiển PID như sau:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Trong đó:

- $u(t)$: tín hiệu điều khiển đầu ra
- $e(t) = r(t) - y(t)$: sai số giữa giá trị đặt và giá trị phản hồi
- K_p, K_i, K_d : các hệ số tỉ lệ, tích phân và vi phân

❖ Khâu tỉ lệ (P):

$$u_p(t) = K_p e(t)$$

Khâu tỉ lệ là khâu giúp thay đổi giá trị đầu ra và tỉ lệ với giá trị sai số. Để đáp ứng được yêu cầu về tỷ lệ, ta thực hiện điều chỉnh độ lợi bằng cách nhân sai số với 1 hằng số tỉ lệ (K_p) như công thức trên. Ở khâu này, sai số càng lớn thì phản ứng sẽ càng mạnh, giúp hệ thống phản hồi nhanh chóng.

❖ Khâu tích phân (I):

$$u_i(t) = K_i \int_0^t e(\tau) d\tau$$

Khâu này sẽ tỉ lệ thuận với biên độ sai số, quãng thời gian xảy ra sai số, nhằm loại bỏ sai số tĩnh và cải thiện độ chính xác lâu dài. Đầu ra của khâu này sẽ được tính toán như công thức trên.

❖ Khâu vi phân (D):

$$u_p(t) = K_d \frac{de(t)}{dt}$$

Khâu này giúp ta dự đoán xu hướng thay đổi của sai số bằng cách tính tốc độ biến đổi của sai số, giúp giảm độ quá điều chỉnh và cải thiện độ ổn định. Được tính bằng công thức trên.

Ưu điểm của bộ điều khiển PID là hoạt động rất hiệu quả, dễ tinh chỉnh đối với mô hình hệ thống tuyến tính còn khi sử dụng bộ PID cho cánh tay Robot là một mô hình hệ thống phi tuyến thì bộ PID truyền thống sẽ tồn tại một vài hạn chế. Cụ thể, nó không xét đến sự tương tác động lực học giữa các khớp, mà chỉ tập trung điều khiển tại từng khớp riêng lẻ. Khi chuyển sang điều khiển một khớp khác, quá trình sẽ bắt đầu lại từ đầu, điều này có thể dẫn đến sai lệch tích lũy trong quỹ đạo tổng thể của đầu cuối robot. Do đó, để nâng cao hiệu suất điều khiển và giảm thiểu sai số, cần xem xét kết hợp PID với các phương pháp điều khiển khác nhằm bù trừ tương tác và cải thiện độ chính xác của hệ thống.

3.2.2. Các phương pháp tính chọn tham số bộ điều khiển PID

Để hệ thống điều khiển đạt được hiệu suất tối ưu nhất thì việc then chốt là chúng ta phải lựa chọn được hệ số PID. Có nhiều phương pháp lựa chọn thông số PID khác nhau trong thực tiễn để tinh chỉnh bộ điều khiển PID, như là:

❖ **Phương pháp điều chỉnh thủ công:**

Đây là một phương pháp có cách tiếp cận đơn giản và trực quan nhất cho người dùng. Chúng ta sẽ lần lượt thay đổi các thông số điều khiển PID dựa trên quan sát phản hồi của hệ thống nhằm đạt được mục tiêu điều khiển như giảm thời gian quá độ, tăng được độ ổn định, giảm sai số xác lập và hạn chế dao động [7].

Đầu tiên chúng ta sẽ cho 2 hệ số K_i và K_d bằng 0 và bắt đầu tăng dần K_p rồi quan sát hệ thống, nếu hệ thống phản hồi chậm chúng ta sẽ tăng K_p để hệ thống đáp ứng nhanh hơn và ngược lại giảm K_p khi hệ thống dao động và mất ổn định. Tiếp theo, chúng ta sẽ thêm K_i vào để giảm sai số xác lập. Tăng K_i dần dần và quan sát hệ thống, nếu hệ thống xác lập chậm thì tăng hệ số K_i lên và ngược lại nếu hệ thống dao động và xuất hiện hiện tượng quá điều chỉnh thì giảm K_i xuống. Cuối cùng, thêm K_d vào để cải thiện dao động và cải thiện ổn định nếu K_p và K_i chưa đáp ứng các điều trên, lựa chọn hệ số K_d phù hợp sao cho hệ thống phản hồi mượt, giảm dao động, cải thiện quá điều chỉnh, không gây ra nhiễu hoặc chậm phản hồi do khuếch đại nhiễu cao tần. Vậy, chúng ta sẽ lặp lại các bước trên để tinh chỉnh cho đến khi hệ thống đạt được các yêu cầu về quá độ, ổn định và sai số.

❖ **Phương pháp Ziegler-Nichols:**

Phương pháp Ziegler-Nichols được phát triển vào những năm 1940 là một kỹ thuật thực nghiệm phổ biến để điều chỉnh các hệ số trong bộ điều khiển PID. Khi áp dụng với vòng điều khiển kín, thì đầu tiên ta cũng sẽ cho các hệ số K_i K_d bằng 0 và bắt đầu tăng K_p cho đến khi hệ thống bắt đầu dao động điều hòa liên tục thì chúng ta sẽ ghi lại giá trị K_p tại thời điểm dao động rồi gán cho giá trị K_u (gọi là hệ số ultimate gain). Tiếp theo ta đo khoảng thời gian giữa 2 đỉnh liền kề nhau của các dao động và đặt đó là P_u (chu kỳ dao động – ultimate period) [7]. Cuối cùng, ta sẽ áp dụng các công thức trong bảng sau để tìm các hệ số K_i và K_d :

Loại điều khiển	K_p	K_i	K_d
P	$0.5K_u$		
PI	$0.45K_u$	$1.2K_p / P_u$	
PID	$0.6K_u$	$2K_p / P_u$	$K_p P_u / 8$

Bảng 3.1: Bảng phương pháp Ziegler-Nichols

❖ **Phương pháp Auto tuning được tích hợp trong phần mềm Matlab Simulink:**

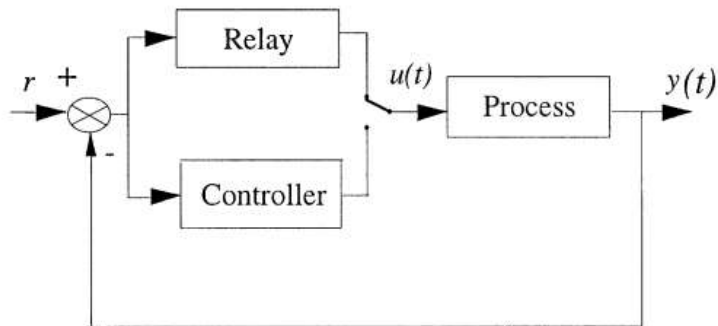
Trong những năm gần đây, việc tinh chỉnh bộ điều khiển PID trở nên thuận tiện và chính xác hơn nhờ các công cụ hỗ trợ thiết kế một cách tự động trong các phần mềm mô phỏng, điển hình là phần mềm Matlab Simulink với công cụ PID Tuner, giúp người dùng điều chỉnh và đánh giá nhanh các thông số PID mà không cần phải tính toán thủ công hay thử quá nhiều lần gây tốn kém thời gian. Với công cụ này sẽ cho phép tự động xác định các thông số tối ưu của bộ điều khiển PID dựa trên mô phỏng các hệ thống tuyến tính, nó sẽ đáp ứng các tiêu chí như tốc độ đáp ứng, độ ổn định và khả năng loại bỏ nhiễu [7].

Để tiến hành sử dụng công cụ này thì đầu tiên chúng ta cần tạo mô hình hệ thống trong Matlab Simulink sau đó chèn các khối PID controller từ thư viện Simulink vào đúng vị trí cần đặt bộ điều khiển PID vào mô hình mô phỏng. Tiếp theo, cho chạy thử nghiệm một lần để bộ điều khiển xác định được đối tượng điều khiển, sau đó bấm vào khối PID Controller rồi nhấn “Tune...” để mở PID Tuner, PID Tuner sẽ tự động tính toán và đưa ra bộ thông số phù hợp với đối tượng cần điều khiển. Ta có thể sử dụng các thanh trượt Response Time (thời gian đáp ứng) và Transient Behavior (độ dao động) để tinh chỉnh theo ý muốn của mình và xem đồ thị đáp ứng. Cuối cùng, khi đã ưng ý với đồ thị đáp ứng đó thì ta bấm “Update Block” để gán các hệ số PID vào khối PID Controller rồi tiến hành chạy mô phỏng lại xem kết quả.

❖ **Phương pháp Relay-feedback:**

Phương pháp Relay-feedback [8] là một kỹ thuật tự động tinh chỉnh bộ điều khiển PID được đề xuất vào đầu những năm 1990. Phương pháp này cho phép ước lượng các thông số quan trọng của hệ thống như biên độ dao động và tần số dao động mà không cần mô hình toán học chi tiết, từ đó suy ra các thông số PID thích hợp.

Phương pháp này thay thế bộ điều khiển PID ban đầu bằng một relay ON – OFF phi tuyến trong một khoảng thời gian ngắn, đưa tín hiệu điều khiển dạng sóng vuông vào hệ thống, điều khiển đầu ra dao động điều hòa xung quanh điểm làm việc.



Hình 3.2: Sơ đồ khối phương pháp Relay - Feedback

Để thực hiện tinh chỉnh bộ điều khiển PID bằng phương pháp Relay – feedback thì đầu tiên chúng ta sẽ thay thế PID bằng khối relay ($\pm d$) trong vòng kín như hình.

Theo thời gian thì hệ thống sẽ dao động điều hòa ổn định, chúng ta tiến hành đo biên độ đầu ra (a) và chu kỳ dao động (P_u) được đo trực tiếp từ đáp ứng đầu ra và áp dụng công thức sau:

$$K_u = \frac{4d}{\pi a}$$

Với:

K_u : độ lợi tới hạn của hệ thống.

a : Biên độ dao động đầu ra của hệ thống.

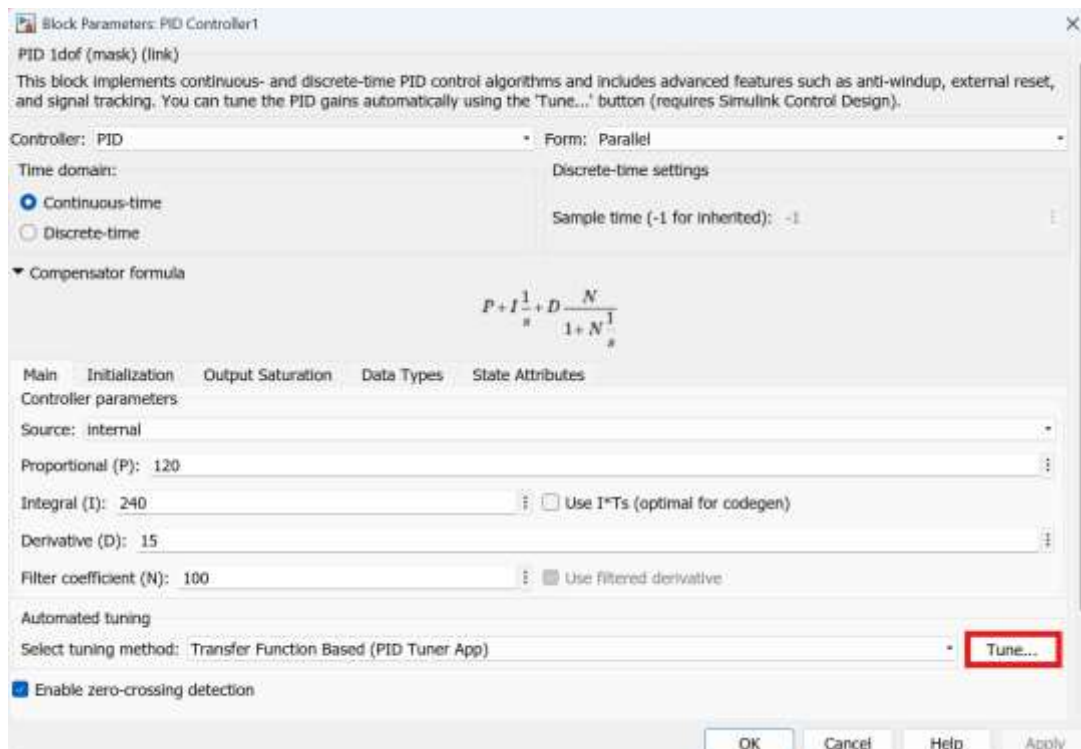
d : Biên độ của tín hiệu relay

Sau khi đã tìm được K_u và P_u , ta áp dụng phương pháp Ziegler – Nichols để tính toán các thông số K_p, K_i, K_d .

3.2.3. Lựa chọn thông số bộ điều khiển PID cho cánh tay Robot 3 bậc tự do

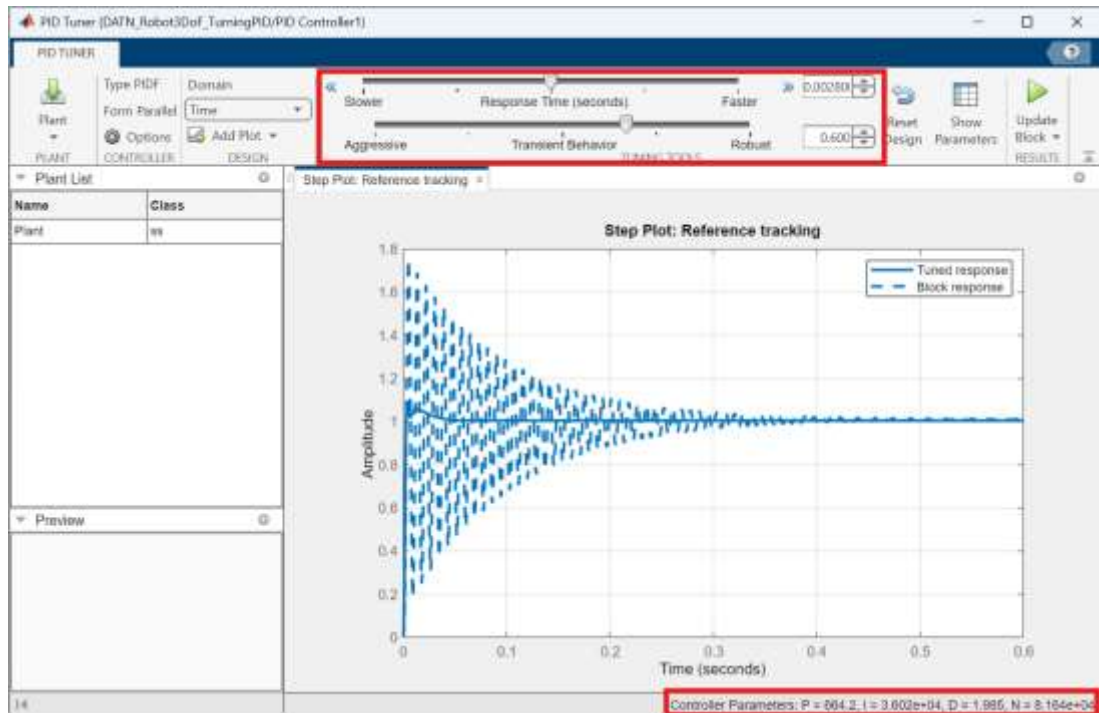
Để đạt được hiệu quả tối ưu nhất cho việc lựa chọn các thông số PID thì chúng ta sẽ sử dụng PID Tuner trong Matlab/Simulink để tinh chỉnh thông số cho bộ điều khiển PID. Sau đây chúng ta sẽ sử dụng bộ điều khiển PID cho cánh tay Robot 3 bậc tự do. Ta tiến hành theo các bước sau:

- Bước 1: Đầu tiên ta mở PID Tuner trong khối PID Controller:



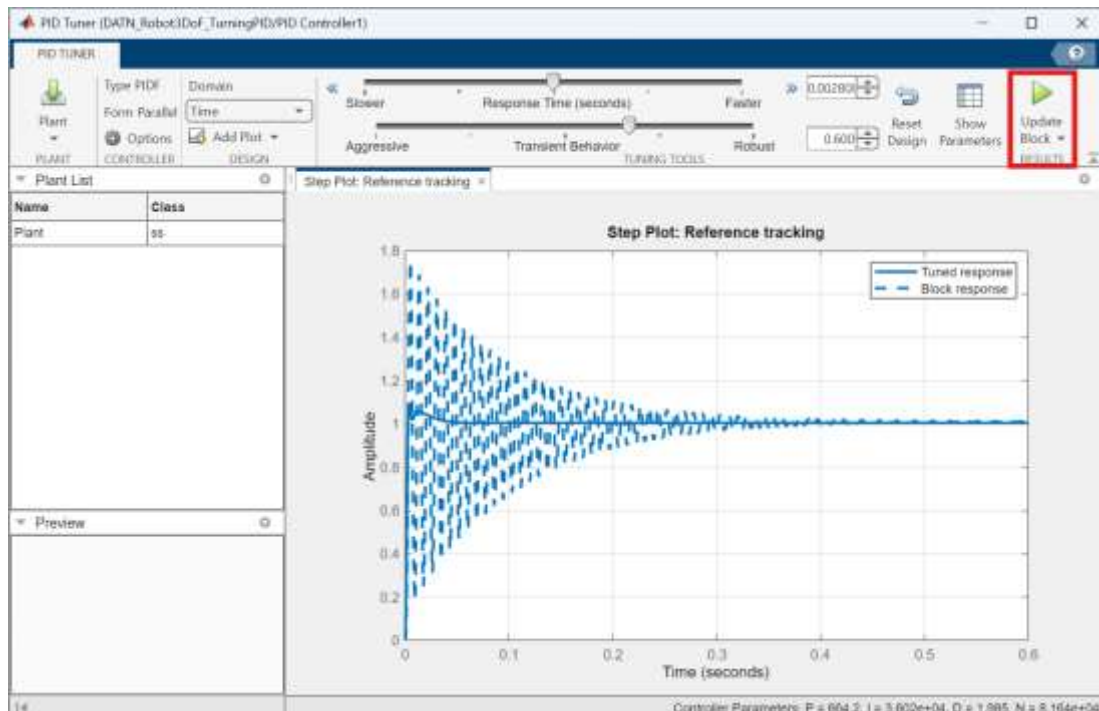
Hình 3.3: Hướng dẫn sử dụng PID Tuner để lựa chọn thông số PID

- Bước 2: PID Tuner sẽ tự động tính toán các hệ số K_p, K_i, K_d và đưa ra đồ thị đáp ứng tương ứng với hệ số K_p, K_i, K_d đó. Chúng ta sẽ sử dụng các thanh trượt Response Time và Transient Behavior để cải thiện chất lượng điều khiển.



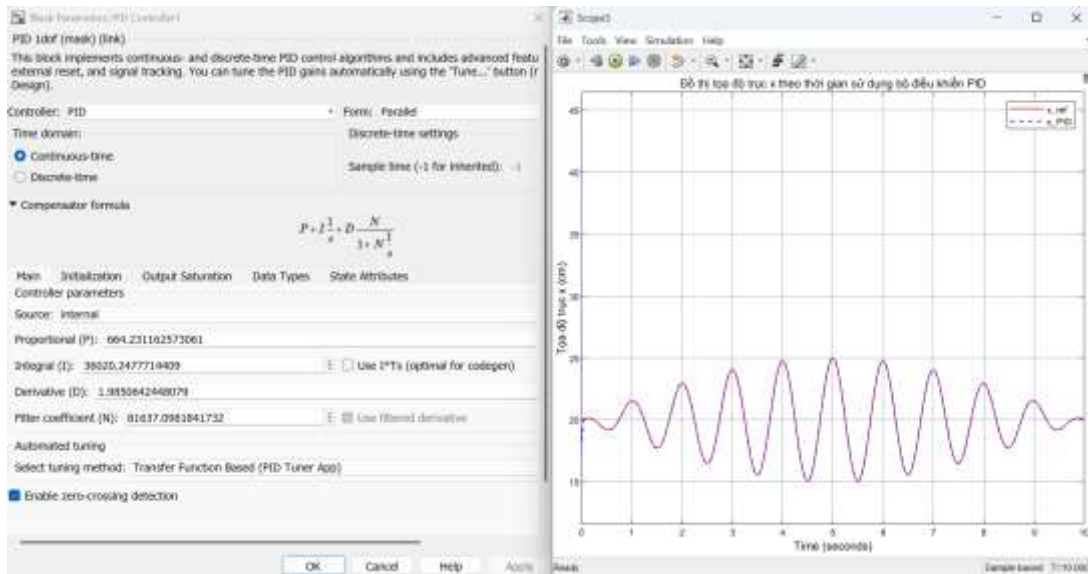
Hình 3.4: Hướng dẫn sử dụng PID Tuner để lựa chọn thông số PID

- Bước 3: Sau khi đã hài lòng với đồ thị đáp ứng thì tiến hành đưa các thông số vào khối PID Controller bằng cách nhấn **Update Block**.



Hình 3.5: Hướng dẫn sử dụng PID Tuner để lựa chọn thông số PID

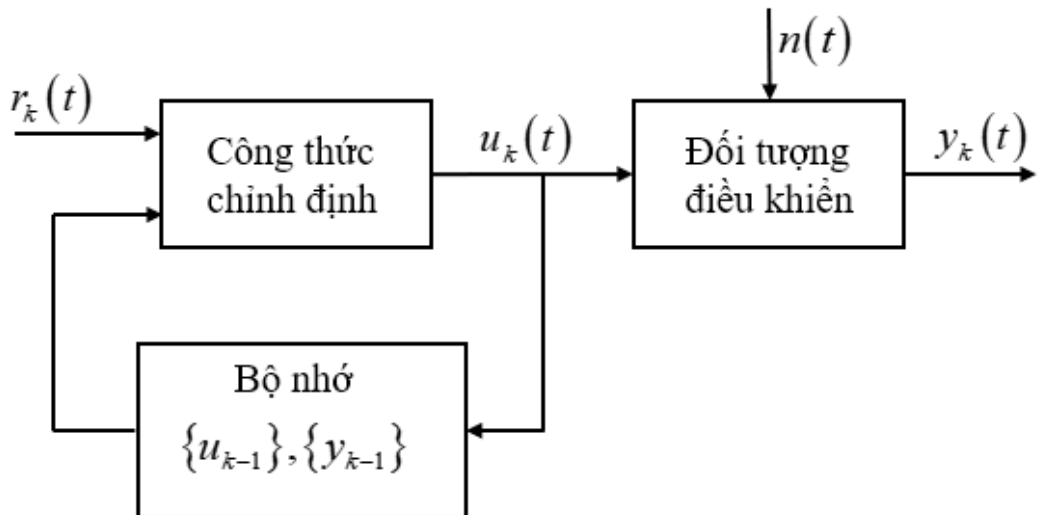
- Bước 4: Cuối là chạy mô phỏng và xem kết quả đáp ứng của bộ điều khiển PID sau khi đã được tối ưu bằng PID Tuner.



Hình 3.6: Đồ thị kết quả sau khi sử dụng PID Tuner để chọn thông số PID

3.3. Phương pháp điều khiển cánh tay Robot sử dụng bộ điều khiển ILC

3.3.1. Tổng quan về bộ điều khiển ILC



Hình 3.7: Sơ đồ hệ điều khiển học lặp với bộ điều khiển ILC

Bộ điều khiển học lặp (ILC – Iterative Learning Control) là một phương pháp điều khiển phi truyền thống được thiết kế để cải thiện hiệu suất điều khiển của các hệ thống lặp lại cùng một tác vụ trong một khoảng thời gian xác định (finite time interval).

Trong phương pháp điều khiển ILC, thông tin từ các lần lặp trước được sử dụng để điều chỉnh tín hiệu điều khiển cho các lần thực hiện sau, nhằm giảm sai số và đạt hiệu

suất tối ưu. Hệ thống “học” từ những sai lệch trong quá khứ để tự cải thiện qua từng chu kì.

Điều khiển học lặp (Iterative Learning Control) là một trong số ít các phương pháp thiết kế bộ điều khiển mà không cần mô hình toán học chính xác của đối tượng điều khiển. Vì nguyên lý hoạt động của nó dựa vào thông tin của các lần lặp lần trước, chứ không dựa vào mô hình hệ thống để tính toán điều khiển. Vì vậy, nó có sự khác biệt so với các phương pháp điều khiển truyền thống. Trong nhiều trường hợp ứng dụng, khi mà ở đó một mình bộ điều khiển học lặp không thể giải quyết được các nhiệm vụ điều khiển đặt ra, người ta có thể kết hợp thêm với nó những bộ điều khiển truyền thống. Công thức tổng quát của ILC có dạng như sau:

$$u_{k+1}(t) = u_k(t) + Ke_k(t)$$

$$\text{Với } e_k(t) = r(t) - y_k(t)$$

Trong đó:

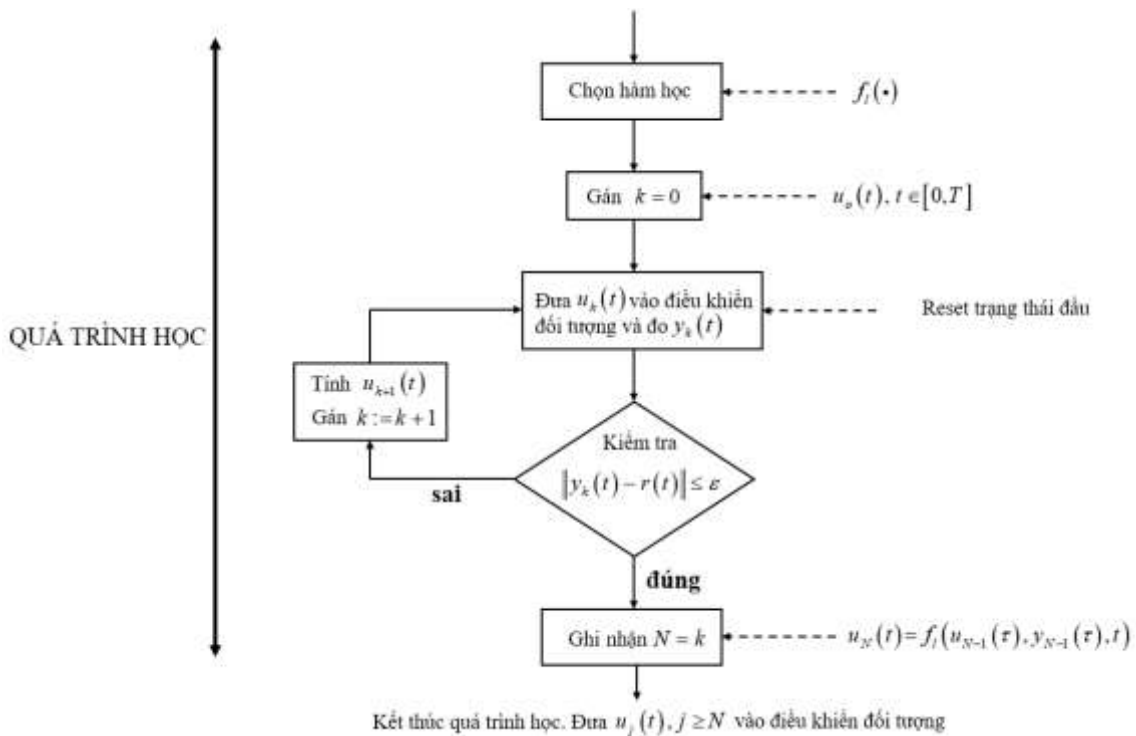
- $u_k(t)$: Tín hiệu điều khiển tại lần lặp thứ k
- $u_{k+1}(t)$: Tín hiệu điều khiển tại lần lặp thứ $k + 1$
- $e_k(t) = r(t) - y_k(t)$: Sai số giữa tín hiệu mong muốn $r(t)$ và đầu ra hệ thống $y_k(t)$
- K : Toán tử học (learning operator) – xác định cách cải tiến điều khiển dựa trên sai số

Bộ điều khiển học lặp này có tên gọi là “học”, vì thay cho mô hình toán, nó lấy thông tin động học của đối tượng điều khiển qua các lần thử rồi chỉnh sửa tín hiệu điều khiển nhằm giảm sai lệch bám cho tới khi vector tín hiệu điều khiển cuối cùng thỏa mãn chất lượng yêu cầu, được hiểu là tín hiệu điều khiển mà với nó, đáp ứng được yêu cầu đặt ra về sai lệch bám theo nghĩa:

$$\|r - y\| \approx 0 \text{ hoặc } \|r - y\| < \varepsilon$$

Với một chuẩn tín hiệu nào đó, chẳng hạn như chuẩn bậc p , chuẩn Euclid, chuẩn hàm mũ, chuẩn vô cùng v.v., trong đó ε là một số thực dương đủ nhỏ cho trước. Cũng chỉ sau khi có được chất lượng, tín hiệu điều khiển cuối cùng tìm được mới thực sự đủ điều kiện áp dụng vào thực tế. Vì lẽ đó mà phương pháp có tên gọi là “lặp” (iterative). Đây cũng là điểm khác biệt so với những thuật toán điều khiển “hồi quy” (recursive), là phương thức mà ở đó những kết quả trung gian cũng vẫn có thể tạm dùng để điều khiển.

3.3.2. Quy trình học lặp của bộ điều khiển ILC



Hình 3.8: Sơ đồ khối các bước học và chỉnh định trong điều khiển học lặp

Sau đây, là tóm tắt lại các bước thiết kế của bộ điều khiển học lặp ILC như ở hình

3.10.

- Bước 1: Trước tiên, ta cần lựa chọn hàm học $f_l(\square)$ phù hợp. Hàm học này đóng vai trò là quy luật cập nhật tín hiệu điều khiển qua mỗi vòng lặp. Hàm học có thể dựa trên luật cộng sai số hoặc dựa trên mô hình hệ thống.
- Bước 2: Ta gán chỉ số vòng lặp đầu tiên $k = 0$. Sau đó khởi tạo tín hiệu điều khiển ban đầu $u_0(t)$ trong khoảng thời gian $t \in [0, T]$
- Bước 3: Ta áp dụng tín hiệu điều khiển và đo đầu ra. Đưa tín hiệu $u_k(t)$ vào hệ thống điều khiển. Reset trạng thái đầu của hệ thống (trở về điều kiện ban đầu trước mỗi chu kỳ để đảm bảo tính lặp). Tiến hành đo đầu ra $y_k(t)$ sau khi áp dụng $u_k(t)$
- Bước 4: Ta kiểm tra và tính toán sai số theo dõi theo công thức:

$$e_k(t) = y_k(t) - r(t)$$

Tiếp theo đó ta so sánh với sai số cho phép ε :

$$|y_k(t) - r(t)| \leq \varepsilon$$

Nếu đúng, tức là sai số nhỏ, đã hội tụ thì kết thúc quá trình học. Nếu sai, tức là chưa đạt yêu cầu chính xác thì ta tăng vòng lặp và tính tín hiệu điều khiển mới $u_{k+1}(t)$ (như ở bước 5).

- Bước 5: Dựa trên hàm học $f_l(\square)$ đã chọn để tính toán cập nhật tín hiệu điều khiển mới. Tăng chỉ số vòng lặp $k = k + 1$, sau đó quay lại bước 3 để tiếp tục vòng lặp với tín hiệu điều khiển mới.
- Bước 6: Sau khi điều kiện hội tụ được thỏa mãn, ghi nhận số vòng lặp hội tụ $N = k$. Ta có tín hiệu điều khiển ổn định lúc này là:

$$u_N(t) = f_l(u_{N-1}(\tau), y_{N-1}(\tau), t)$$

Từ vòng lặp $j \geq N$, sử dụng tín hiệu $u_j(t)$ để điều khiển hệ thống chính thức vì nó đã hội tụ và đảm bảo bám quỹ đạo với sai số nhỏ.

Trong điều khiển học lặp (Iterative Learning Control – ILC), số lần học (hay còn gọi là số vòng lặp) là số lần hệ thống thực hiện lại cùng một tác vụ để cải thiện hiệu suất điều khiển dựa trên sai số của các lần trước. Việc xác định số lần học là yếu tố quan trọng ảnh hưởng đến độ chính xác và hiệu quả của quá trình điều khiển. Có hai cách phổ biến để xác định số lần học:

- *Chọn trước số lần học (fixed number of iterations)*

Trong phương pháp này, số lần học được xác định trước, thường dựa trên kinh nghiệm, giới hạn thời gian thực nghiệm. Hệ thống sẽ thực hiện điều khiển lặp lại đúng N vòng lặp cho trước, sau đó dừng lại và sử dụng kết quả ở vòng cuối cùng, bất kể sai số lớn hay nhỏ.

Ưu điểm:

- Dễ triển khai
- Thời gian thực hiện được kiểm soát tốt

Nhược điểm:

- Có thể dừng quá sớm khi sai số còn quá lớn
- Hiệu suất điều khiển chưa tối ưu
- *Học theo hàm mục tiêu (goal-based or performance-based learning)*

Với phương pháp này, quá trình học tiếp tục cho đến khi đạt được một mục tiêu định trước về hiệu suất điều khiển. Mục tiêu này thường là một ngưỡng sai số đủ nhỏ hoặc một điều kiện hội tụ nhất định. Quá trình học dừng lại khi sai số tuyệt đối nhỏ hơn một giá trị ε (ngưỡng sai số tối đa cho phép):

$$|y_k(t) - r(t)| \leq \varepsilon$$

Nói theo cách khác, nếu ở vòng lặp thứ k , sai số tại mọi thời điểm đều nhỏ hơn ε , thì coi như hệ thống đã học đủ tốt, không cần học lặp thêm nữa.

Ưu điểm:

- Giúp hệ thống đạt được hiệu suất điều khiển tốt hơn
- Tránh lãng phí học quá ít hoặc quá nhiều

Nhược điểm:

- Thời gian học có thể kéo dài nếu hệ chậm hội tụ
- Cần xác định ngưỡng sai số phù hợp

Tùy vào yêu cầu thực tế của hệ thống, người thiết kế có thể lựa chọn số lần học cố định cho trước nếu muốn đơn giản, hoặc số lần học theo hàm mục tiêu để đạt độ chính xác cao hơn. Trong các ứng dụng yêu cầu chất lượng đầu ra cao và độ chính xác lớn (ví dụ như Robot hàn, Robot y tế), phương pháp học theo mục tiêu thường được ưu tiên sử dụng.

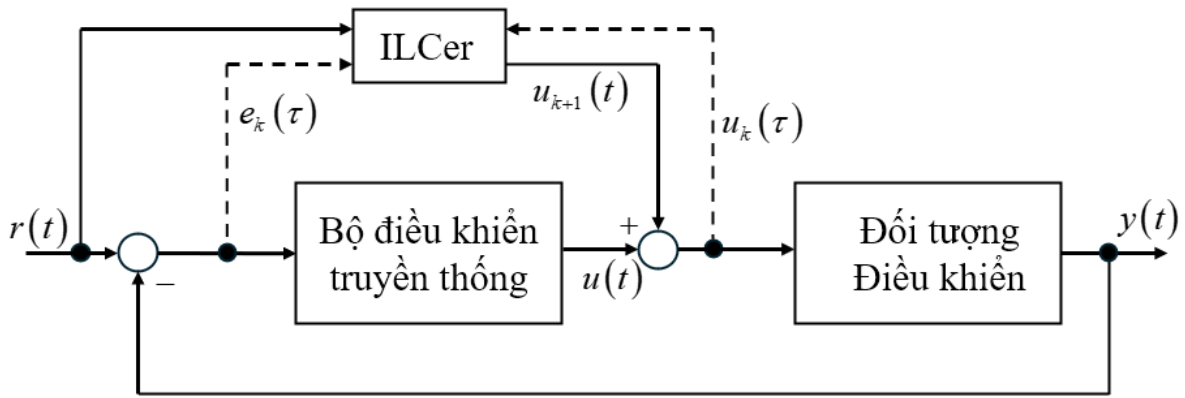
3.3.3. Thuật toán điều khiển học lặp ILC cho cánh tay Robot 3 bậc tự do

Đặc điểm của cánh tay Robot 3 bậc tự do là một hệ không ổn định có thể dao động, nên không thể dùng ILC dạng đơn giản như cho các hệ ổn định. Ngoài ra hệ còn có tính phi tuyến mạnh, các khớp liên kết chặt chẽ với nhau nên nhiễu hay lỗi ở một khớp ảnh hưởng đến toàn hệ.

Để có thể áp dụng được điều khiển học lặp cho các đối tượng có động học không ổn định, ta cần phải ổn định hóa nó trước bằng một bộ điều khiển phản hồi tín hiệu đầu ra hoặc phản hồi trạng thái dạng truyền thống được gọi là mạch vòng điều khiển kín truyền thống rồi sau đó mới đưa bộ điều khiển học lặp ILC vào với vai trò là một bộ điều khiển hở cho mạch vòng điều khiển kín này.

Như vậy, nếu xét về cấu trúc tổng thể thì ở đây bộ điều khiển học lặp ILC sẽ giữ vai trò như một bộ điều khiển thông minh dạng tiền xử lý hoặc dạng truyền thẳng trong mạch vòng kín, tức là một bộ điều khiển tác động tới đầu vào của đối tượng từ các đầu vào khác của mạch vòng kín đó, chẳng hạn như từ tín hiệu đặt hoặc nhiễu v.v., thay vì từ đầu ra của đối tượng.

Để điều khiển cho cánh tay Robot 3 bậc tự do trong đề tài này, nhóm sử dụng phương pháp điều khiển học lặp ILC kết hợp với bộ điều khiển truyền thống là PID. Hệ thống này áp dụng cho các bài toán điều khiển có tính lặp lại theo chu kỳ.



Hình 3.9: Sơ đồ mô tả một hệ thống kết hợp giữa bộ điều khiển truyền thống và bộ điều khiển học lặp ILC

Sơ đồ điều khiển mô tả một hệ thống kết hợp giữa bộ điều khiển truyền thống PID và bộ điều khiển học lặp (ILC – Iterative Learning Control) nhằm nâng cao hiệu quả điều khiển qua các chu kỳ lặp lại. Trong hệ thống này, đầu vào là tín hiệu tham chiếu $r(t)$, đại diện cho quỹ đạo hoặc đầu ra mong muốn theo thời gian. Bộ điều khiển truyền thống PID xử lý sai số hiện thời giữa tín hiệu mong muốn và đầu ra thực tế của hệ thống, được tính là $e(t) = r(t) - y(t)$, để tạo ra tín hiệu điều khiển cơ bản $u(t)$.

Tín hiệu điều khiển này sau đó được đưa vào đối tượng điều khiển, trong bài toán điều khiển này đối tượng điều khiển là cánh tay Robot 3 bậc tự do. Đối tượng điều khiển nhận tín hiệu $u(t)$ và phản hồi lại đầu ra $y(t)$. Tuy nhiên, để cải thiện chất lượng điều khiển theo thời gian, đặc biệt trong các tác vụ mang tính lặp lại, hệ thống còn sử dụng thêm một bộ điều khiển học lặp ILC. Bộ điều khiển này ghi nhớ và sử dụng thông tin từ các lần lặp trước như sai số $e_k(\tau) = r(\tau) - y_k(\tau)$ và lệnh điều khiển $u_k(t)$, từ đó tính toán lệnh điều khiển cải tiến cho lần lặp tiếp theo $u_{k+1}(t)$.

Tín hiệu điều khiển cuối cùng đưa vào hệ thống là sự kết hợp giữa hai thành phần: lệnh từ bộ điều khiển truyền thống và lệnh học được từ ILC, cụ thể là:

$$u(t) = u_{truyền\ thong}(t) + u_k ILC(t)$$

Bộ điều khiển truyền thống PID giúp nhận biết sai số giữa đầu ra thực tế và quỹ đạo mong muốn rồi từ đó tính toán đưa ra một lệnh điều khiển để điều chỉnh Robot hoạt động gần đúng hơn.

Bộ điều khiển ILC giúp cánh tay Robot ghi lại quỹ đạo mong muốn, sai số giữa thực tế và đầu ra mong muốn $e_k(t)$ và tín hiệu điều khiển đã dùng ở lần trước $u_k(t)$. Dựa vào đó, ở lần thực hiện tiếp theo bộ điều khiển sẽ cập nhật lệnh điều khiển sao cho Robot bám quỹ đạo mong muốn một cách tốt nhất.

Cách kết hợp này đảm bảo cánh tay Robot vừa ổn định ngay từ lần điều khiển đầu tiên, vừa có khả năng cải thiện độ chính xác trong các lần lặp sau. Nhờ đó, cánh tay Robot có thể tự học từ sai số của chính nó và ngày càng điều khiển chính xác hơn. Phương pháp này đặc biệt hiệu quả trong các ứng dụng điều khiển lặp như Robot vẽ, Robot hàn, in 3D.

3.3.4. Thiết kế bộ điều khiển học lặp ILC cho cánh tay Robot 3 bậc tự do

3.3.4.1 Thiết kế học lặp kiểu P

Trong số các thuật toán ILC, ILC kiểu P là dạng cơ bản và phổ biến nhất. Nó có ưu điểm là cấu trúc đơn giản nhất của điều khiển học lặp, trong đó tín hiệu điều khiển ở mỗi lần lặp được xây dựng dựa trên sai số hiện tại mà không xét đến tốc độ thay đổi sai số từ các lần lặp trước đó. Giả sử hệ thống rời rạc tuyến tính bất biến theo thời gian, tín hiệu đầu ra được xác định:

$$Y_{j+1} = PU_{j+1} \quad (1)$$

$$Y_j = PU_j \quad (2)$$

Ta lấy (1) – (2), được phương trình:

$$Y_{j+1} - Y_j = P(U_{j+1} - U_j)$$

Trong ILC, tín hiệu điều khiển được cập nhật sau mỗi lần lặp: $U_{j+1} = U_j + V_j$, với V_j là thành phần điều chỉnh dựa trên sai số.

$$\Rightarrow Y_{j+1} - Y_j = PV_j$$

Trong đó:

- Y_j : Đầu ra hệ thống tại lần lặp j
- Y_{j+1} : Đầu ra hệ thống tại lần lặp $j + 1$
- U_j : Tín hiệu điều khiển tại lần lặp j
- U_{j+1} : Tín hiệu điều khiển tại lần lặp $j + 1$
- P : Toán tử biểu diễn động học của hệ thống (ma trận)

Sai số đầu ra được xác định:

$$E_{j+1} = Y_d - Y_{j+1} \quad (3)$$

$$E_j = Y_d - Y_j \quad (4)$$

Ta lấy (3) – (4), được phương trình:

$$E_{j+1} - E_j = -(Y_{j+1} - Y_j) \quad (5)$$

$$\Leftrightarrow E_{j+1} = E_j - PV_j \quad (6)$$

Trong đó:

- E_j : Sai số tại lần lặp j
- E_{j+1} : Sai số tại lần lặp $j + 1$
- Y_j : Tín hiệu đầu ra tại lần lặp j
- Y_{j+1} : Tín hiệu đầu ra tại lần lặp $j + 1$
- Y_d : Tín hiệu đặt

Ta có công thức dạng P (Proportional) của V_j :

$$V_j = K_p E_j$$

Trong đó với K_p là hằng số tỷ lệ giúp điều chỉnh lệnh điều khiển dựa trên độ lớn sai số tại thời điểm hiện tại. Nếu sai số lớn thì tín hiệu điều khiển sẽ được điều chỉnh nhiều hơn, ngược lại nếu sai số nhỏ thì điều chỉnh ít lại.

Ta thay $V_j = K_p E_j$ vào (6), ta sẽ được phương trình sai số lặp ILC kiểu P :

$$\begin{aligned} E_{j+1} &= E_j - PK_p E_j \\ \Leftrightarrow E_{j+1} &= (I - PK_p) E_j \end{aligned}$$

Giả sử ta cho $P = G$, vai trò của G là toán tử của hệ thống (biểu diễn các ma trận hoặc hàm truyền), ta sẽ có phương trình rút gọn của ILC kiểu P :

$$E_{j+1} = G_c E_j$$

Trong đó:

$$G_c = I - GK_p$$

Trong điều khiển học lặp ILC kiểu P , mục tiêu là làm cho sai số đầu ra giữa tín hiệu mong muốn và tín hiệu thực tế giảm dần sau mỗi chu kỳ lặp. Khi đó cần đảm bảo điều kiện:

$$\|G_c\| < 1$$

Nếu chuẩn $\|G_c\|$ nhỏ hơn 1, thì:

$$\|E_{j+1}\| < \|E_j\|$$

\Rightarrow Sai số giảm dần qua các vòng lặp, hệ thống ổn định theo nghĩa hội tụ.

Việc chọn K_p rất quan trọng và đảm bảo điều kiện hội tụ. Nếu chọn K_p quá cao thì là hệ dễ dao động, sai số tăng và mất ổn định. Còn nếu chọn K_p quá thấp thì hệ học rất chậm, cần rất nhiều vòng lặp mới đạt sai số nhỏ.

Sau nhiều lần tinh chỉnh thực nghiệm K_p để tìm ra giá trị phù hợp trong điều khiển học lặp kiểu P . Từ đó lựa chọn giá trị K_p cuối cùng cho sai số giảm nhanh, ổn định và không dao động cho hệ thống.

Ưu điểm: Điều khiển học lặp ILC kiểu P có cấu trúc điều khiển đơn giản, chỉ cần sử dụng một hệ số học K_p , giúp dễ dàng triển khai trong thực tế mà không yêu cầu mô hình toán học phức tạp của hệ thống. Có khả năng dễ dàng tích hợp với các bộ điều khiển khác, từ đó nâng cao ổn định và tốc độ hội tụ giúp hệ thống hoạt động một cách ổn định và chính xác nhất.

Nhược điểm: Điều khiển học lặp ILC kiểu P khó áp dụng cho các hệ không lặp hoặc đầu vào thay đổi liên tục giữa các chu kỳ. Độ nhạy của hệ thống đối với giá trị của K_p , nếu chọn quá nhỏ thì tốc độ học sẽ chậm và khi chọn quá lớn có thể khiến hệ mất ổn định hoặc dao động mạnh.

Kết luận: Điều khiển học lặp ILC kiểu P là một công cụ mạnh mẽ và hiệu quả khi được áp dụng đúng trong các hệ thống lặp tuần hoàn. Tuy nhiên, để đạt được hiệu quả tối ưu, người thiết kế cần có phương pháp lựa chọn và tinh chỉnh hệ số học K_p phù hợp với đặc tính của hệ thống.

3.3.4.2 Thiết kế học lặp kiểu D

Thuật toán điều khiển học lặp ILC kiểu D là một phương pháp sử dụng biến thiên sai số giữa các lần lặp để cập nhật tín hiệu điều khiển. Không giống như ILC kiểu P , vốn chỉ sử dụng sai số hiện tại thì ILC kiểu D quan tâm đến tốc độ thay đổi sai số, nhằm phản ứng nhanh hơn với xu hướng sai lệch đầu ra. Giả sử hệ thống rời rạc tuyến tính bất biến theo thời gian, tín hiệu đầu ra được xác định:

$$Y_{j+1} = PU_{j+1} \quad (1)$$

$$Y_j = PU_j \quad (2)$$

Ta lấy (1) – (2), được phương trình:

$$Y_{j+1} - Y_j = P(U_{j+1} - U_j)$$

Trong ILC, tín hiệu điều khiển được cập nhật sau mỗi lần lặp: $U_{j+1} = U_j + V_j$, với V_j là thành phần điều chỉnh dựa trên sai số.

$$\Rightarrow Y_{j+1} - Y_j = PV_j$$

Trong đó:

- Y_j : Đầu ra hệ thống tại lần lặp j

- Y_{j+1} : Đầu ra hệ thống tại lần lặp $j + 1$
- U_j : Tín hiệu điều khiển tại lần lặp j
- U_{j+1} : Tín hiệu điều khiển tại lần lặp $j + 1$
- P : Toán tử biểu diễn động học của hệ thống (ma trận)

Sai số đầu ra được xác định:

$$E_{j+1} = Y_d - Y_{j+1} \quad (3)$$

$$E_j = Y_d - Y_j \quad (4)$$

Ta lấy (3) – (4), được phương trình:

$$E_{j+1} - E_j = -(Y_{j+1} - Y_j) \quad (5)$$

$$\Leftrightarrow E_{j+1} = E_j - PV_j \quad (6)$$

Trong đó:

- E_j : Sai số tại lần lặp j
- E_{j+1} : Sai số tại lần lặp $j + 1$
- Y_j : Tín hiệu đầu ra tại lần lặp j
- Y_{j+1} : Tín hiệu đầu ra tại lần lặp $j + 1$
- Y_d : Tín hiệu đặt

Ta có công thức dạng D (Derivative) của V_j :

$$V_j = K_D E_j - K_D E_{j-1}$$

Trong đó K_D là hệ số đạo hàm phản ánh xu hướng thay đổi của sai số. Việc sử dụng K_D giúp hệ thống phản ứng theo tốc độ thay đổi của sai số, giúp làm mượt và ổn định quá trình học lặp, giảm dao động và tăng độ hội tụ.

Ta thay $V_j = K_D E_j - K_D E_{j-1}$ vào (6), ta sẽ được phương trình sai số lặp ILC kiểu D :

$$\begin{aligned} E_{j+1} &= E_j - P(K_D E_j - K_D E_{j-1}) \\ \Leftrightarrow E_{j+1} &= (I - K_D P)E_j + K_D P E_{j-1} \end{aligned}$$

Giả sử ta cho $P = G$, vai trò của G là toán tử của hệ thống (biểu diễn các ma trận hoặc hàm truyền) và F là toán tử trễ theo vòng lặp trong biểu thức tổng quát của ILC ($E_{j-1} = F E_j$), ta sẽ có phương trình rút gọn của ILC kiểu PD :

$$E_{j+1} = G_c E_j$$

Trong đó:

$$G_c = I - K_D G + K_D G F$$

Trong điều khiển học lặp ILC kiểu D , mục tiêu là làm cho sai số đầu ra giữa tín hiệu mong muốn và tín hiệu thực tế giảm dần sau mỗi chu kỳ lặp. Khi đó cần đảm bảo điều kiện:

$$\|G_c\| < 1$$

Nếu chuẩn $\|G_c\|$ nhỏ hơn 1, thì:

$$\|E_{j+1}\| < \|E_j\|$$

\Rightarrow Sai số giảm dần qua các vòng lặp, hệ ổn định và đầu ra dần khớp với mong muốn.

Việc chọn K_D rất quan trọng và nhạy cảm. Nếu chọn K_D quá lớn thì thuật toán điều khiển sẽ phản ứng quá mức. Còn nếu chọn K_D quá nhỏ thì hệ vẫn hội tụ và không bị dao động tuy nhiên tốc độ hội tụ rất chậm.

Sau nhiều lần tinh chỉnh thực nghiệm K_D để tìm ra giá trị phù hợp trong điều khiển học lặp kiểu D . Từ đó lựa chọn giá trị K_D cuối cùng cho sai số giảm nhanh, ổn định và không dao động cho hệ thống.

Ưu điểm: Điều khiển học lặp ILC kiểu D giúp tăng tốc độ hội tụ sai số, đặc biệt khi sai số thay đổi nhanh. Ngoài ra còn làm cho hệ giảm dao động nhanh và ổn định hơn ở các vòng lặp sau. Thích hợp cho các hệ đáp ứng nhanh hoặc chứa nhiễu thấp.

Nhược điểm: Điều khiển học lặp ILC kiểu D rất nhạy với nhiễu do tính đạo hàm từ sai số, dễ khuếch đại nhiễu tần số cao. Không phản ứng với sai số tĩnh, nếu sai số không thay đổi theo thời gian, thành phần D sẽ không đóng góp. Thường kết hợp với thành phần P để đảm bảo hội tụ đầy đủ tạo thành ILC kiểu PD .

Kết luận: Thuật toán điều khiển học lặp kiểu D là một phương pháp điều chỉnh lệnh điều khiển dựa trên đạo hàm của sai số theo thời gian trong cùng một vòng lặp. Khi được sử dụng đúng cách và trong hệ thống có đặc tính động nhanh hoặc ít nhiễu, học lặp kiểu D có thể giúp rút ngắn thời gian hội tụ và cải thiện hiệu quả điều khiển. Tuy nhiên, để tăng độ ổn định và khả năng bù sai số tĩnh, thuật toán này nên được kết hợp với ILC kiểu P để tạo thành bộ điều khiển PD .

3.3.4.3 Thiết kế học lặp kiểu PD

Thuật toán học lặp kiểu PD là sự kết hợp của thành phần tỉ lệ (P) và thành phần đạo hàm (D) của sai số. Đây là dạng ILC nâng cao giúp cải thiện tốc độ hội tụ và giảm dao động. Mục tiêu của ILC là cải thiện chất lượng đầu ra sau mỗi lần lặp bằng cách học từ sai số lần trước. Giả sử hệ thống rời rạc tuyến tính bất biến theo thời gian, tín hiệu đầu ra được xác định:

$$Y_{j+1} = PU_{j+1} \quad (1)$$

$$Y_j = PU_j \quad (2)$$

Ta lấy (1) – (2), được phương trình:

$$Y_{j+1} - Y_j = P(U_{j+1} - U_j)$$

Trong ILC kiểu, tín hiệu điều khiển được cập nhật sau mỗi lần lặp: $U_{j+1} = U_j + V_j$, với V_j là thành phần điều chỉnh dựa trên sai số.

$$\Rightarrow Y_{j+1} - Y_j = PV_j$$

Trong đó:

- Y_j : Đầu ra hệ thống tại lần lặp j
- Y_{j+1} : Đầu ra hệ thống tại lần lặp $j + 1$
- U_j : Tín hiệu điều khiển tại lần lặp j
- U_{j+1} : Tín hiệu điều khiển tại lần lặp $j + 1$
- P : Toán tử biểu diễn động học của hệ thống (ma trận)

Sai số đầu ra được xác định:

$$E_{j+1} = Y_d - Y_{j+1} \quad (3)$$

$$E_j = Y_d - Y_j \quad (4)$$

Ta lấy (3) – (4), được phương trình:

$$E_{j+1} - E_j = -(Y_{j+1} - Y_j) \quad (5)$$

$$\Leftrightarrow E_{j+1} = E_j - PV_j \quad (6)$$

Trong đó:

- E_j : Sai số tại lần lặp j
- E_{j+1} : Sai số tại lần lặp $j + 1$
- Y_j : Tín hiệu đầu ra tại lần lặp j

- Y_{j+1} : Tín hiệu đầu ra tại lần lặp $j + 1$
- Y_d : Tín hiệu đặt

Ta có công thức dạng PD (Proportional-Derivative) của V_j :

$$V_j = (K_P + K_D)E_j - K_D E_{j-1}$$

Trong đó:

- K_P : Hằng số tỷ lệ, giúp điều chỉnh lệnh điều khiển dựa trên độ lớn sai số tại thời điểm hiện tại. Nếu sai số lớn thì tín hiệu điều khiển sẽ được điều chỉnh nhiều hơn, ngược lại nếu sai số nhỏ thì điều chỉnh ít lại.
- K_D : Hệ số đạo hàm, phản ánh xu hướng thay đổi của sai số. Việc sử dụng K_D giúp hệ thống phản ứng theo tốc độ thay đổi của sai số, giúp làm mượt và ổn định quá trình học lặp, giảm dao động và tăng độ hội tụ.

Tổng hợp hai thành phần P và D tạo thành bộ điều khiển học lặp ILC kiểu PD giúp điều khiển hệ lặp dựa trên hai yếu tố là độ lệch hiện tại và xu hướng sai lệch.

Ta thay $V_j = (K_P + K_D)E_j - K_D E_{j-1}$ vào (6), ta sẽ được phương trình sai số lặp ILC kiểu PD :

$$\begin{aligned} E_{j+1} &= E_j - P[(K_P + K_D)E_j - K_D E_{j-1}] \\ \Leftrightarrow E_{j+1} &= [I - (K_P + K_D)P]E_j + K_D P E_{j-1} \end{aligned}$$

Giả sử ta cho $P = G$, vai trò của G là toán tử của hệ thống (biểu diễn các ma trận hoặc hàm truyền) và F là toán tử trễ theo vòng lặp trong biểu thức tổng quát của ILC kiểu PD ($E_{j-1} = F E_j$), ta sẽ có phương trình rút gọn của ILC kiểu PD :

$$E_{j+1} = G_c E_j$$

Trong đó:

$$G_c = I - (K_P + K_D)G + K_D GF$$

Trong điều khiển học lặp ILC kiểu PD , mục tiêu là làm cho sai số đầu ra giữa tín hiệu mong muốn và tín hiệu thực tế giảm dần sau mỗi chu kỳ lặp. Khi đó cần đảm bảo điều kiện:

$$\|G_c\| < 1$$

Nếu chuẩn $\|G_c\|$ nhỏ hơn 1, thì:

$$\|E_{j+1}\| < \|E_j\|$$

\Rightarrow Sai số giảm dần qua các vòng lặp, hệ ổn định và đầu ra dần khớp với mong muốn.

Hai hệ số học lặp K_p và K_D có vai trò rất quan trọng trong điều khiển học lặp ILC kiểu PD là yếu tố quyết định đến tốc độ hội tụ, độ ổn định và khả năng triệt dao động của hệ thống. Nếu chọn quá lớn thì làm cho hệ dao động, phân kỳ và gây ra khuếch đại nhiễu đạo hàm. Còn nếu chọn quá nhỏ thì tốc độ học rất chậm và hiệu suất điều khiển thấp.

Sau nhiều lần tinh chỉnh thực nghiệm K_p và K_D một cách cẩn thận, bắt đầu từ giá trị nhỏ, sau đó theo dõi phản ứng điều khiển và tăng dần hợp lý. Từ đó lựa chọn giá trị K_p và K_D cuối cùng cho sai số giảm nhanh, ổn định và không dao động cho hệ thống.

Ưu điểm: Điều khiển học lặp kiểu PD là một mở rộng của ILC kiểu P , ưu điểm chính của học lặp kiểu PD là tăng tốc độ hội tụ. Thành phần P giúp giảm sai số theo cách trực tiếp, trong khi thành phần D dự đoán xu hướng biến đổi sai số và phản ứng sớm, nhờ đó rút ngắn quá trình học và đưa hệ nhanh chóng tiến gần quỹ đạo mong muốn. Bên cạnh đó, thành phần đạo hàm còn giúp giảm dao động và làm mượt đáp ứng, nhất là trong giai đoạn quá độ. Điều này đặc biệt có lợi trong các hệ thống dễ bị rung hoặc nhiễu động nhẹ. Hơn nữa, ILC kiểu PD vẫn giữ được cấu trúc đơn giản và dễ triển khai trong thực tế, phù hợp với nhiều loại hệ thống điều khiển rời rạc.

Nhược điểm: Tuy nhiên, phương pháp này cũng tồn tại một số nhược điểm nhất định. Đầu tiên, việc lựa chọn các hệ số học lặp K_p và K_D là không dễ dàng, đòi hỏi phải có quá trình tinh chỉnh cẩn thận để đảm bảo hệ thống vừa hội tụ nhanh vừa tránh dao động. Nếu K_D được chọn quá lớn, hệ thống dễ bị khuếch đại nhiễu đạo hàm, gây bất ổn hoặc phân kỳ. Mặt khác, nếu các hệ số quá nhỏ thì tốc độ hội tụ lại rất chậm, làm giảm hiệu quả học. Ngoài ra, trong các hệ có nhiễu mạnh hoặc phi tuyến, việc sử dụng đạo hàm sai số có thể cần kết hợp với bộ lọc hoặc kỹ thuật giảm nhiễu bổ sung để đảm bảo tính ổn định.

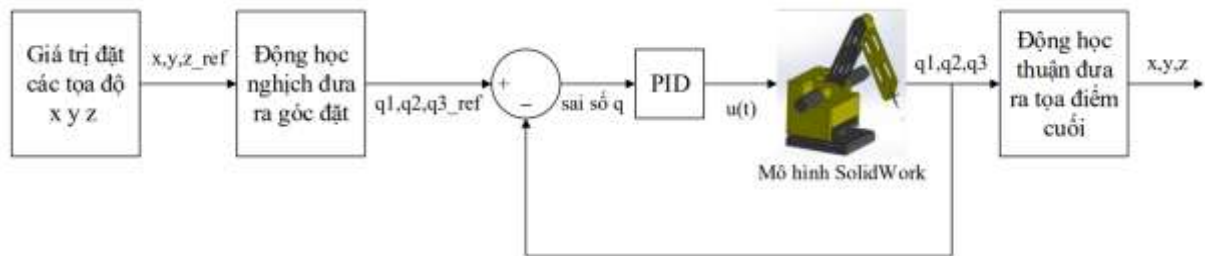
Kết luận: Việc lựa chọn K_p và K_D trong ILC kiểu PD phải dựa trên đặc tính của hệ thống (ổn định, nhiễu, độ dao động), và phải được tinh chỉnh từng bước qua thử nghiệm thực tế. K_p đóng vai trò giảm sai số chính, trong khi K_D giúp làm mượt đáp ứng và ngăn dao động, nhưng rất dễ gây bất ổn nếu không xử lý nhiễu đạo hàm. Do đó, phương pháp an toàn là: bắt đầu nhỏ \rightarrow tăng dần \rightarrow theo dõi sai số \rightarrow dừng tại ngưỡng ổn định tốt nhất.

Chương 4: MÔ PHỎNG ĐIỀU KHIỂN CÁNH TAY ROBOT VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Sơ đồ khối điều khiển và các chỉ tiêu đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do

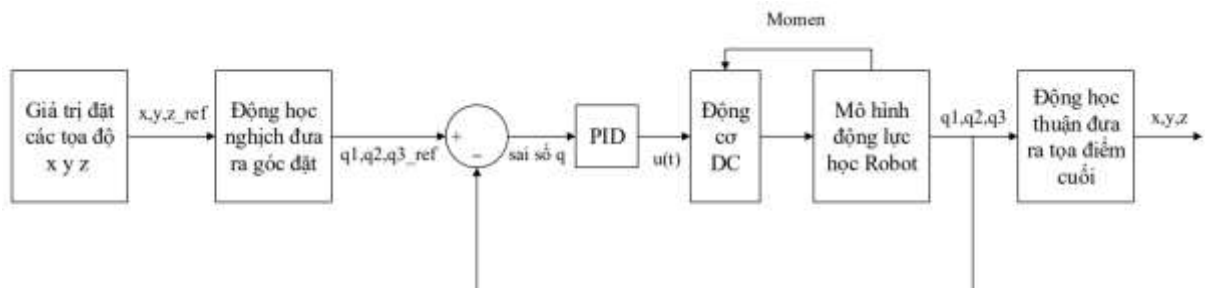
4.1.1. Sơ đồ khối điều khiển mô hình SolidWork và mô hình Động lực học sử dụng bộ điều khiển PID

Sơ đồ khối mô hình cánh tay Robot 3DOF xây dựng trên SolidWork sử dụng bộ điều khiển PID được thể hiện như hình 4.1.



Hình 4.1: Sơ đồ khối mô hình SolidWork sử dụng bộ điều khiển PID

Sơ đồ khối mô hình động lực học cánh tay Robot 3DOF sử dụng bộ điều khiển PID được thể hiện như hình 4.2.



Hình 4.2: Sơ đồ khối mô hình động lực học sử dụng bộ điều khiển PID

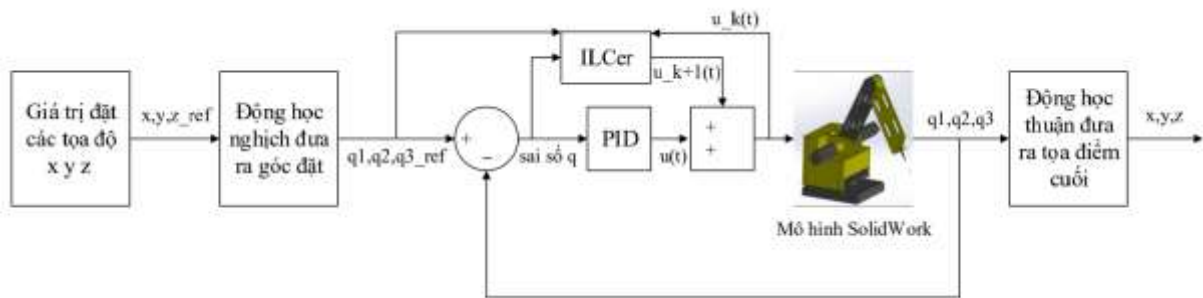
Các giá trị đặt tọa độ $x y z$ của hình vẽ 2D hoặc 3D mong muốn sẽ được đưa vào khối Động học nghịch để tính toán góc đặt q_d cho các khớp quay. Góc đặt q sẽ được so sánh với góc q_r từ mô hình Robot, sai lệch góc quay sẽ được đưa vào bộ điều khiển PID để đưa ra tín hiệu điều khiển u đưa vào cho mô hình Robot để tính toán và đưa ra góc q_r và đưa vào khối Động học thuận để tính toán và cho ra tọa độ $x y z$ tại điểm cuối ở khâu chấp hành của cánh tay Robot.

Yêu cầu điều khiển: Đưa điểm cuối ở khâu chấp hành của cánh tay Robot đến đúng vị trí đặt theo thời gian, đảm bảo chuyển động mượt, chính xác và ổn định khi thực hiện vẽ 2D và in 3D. Bộ điều khiển PID cần điều khiển từng trục tọa độ $x y z$ để giảm sai số

vị trí đến giá trị nhỏ nhất có thể, đảm bảo phản hồi nhanh và chính xác, thời gian ổn định ngắn.

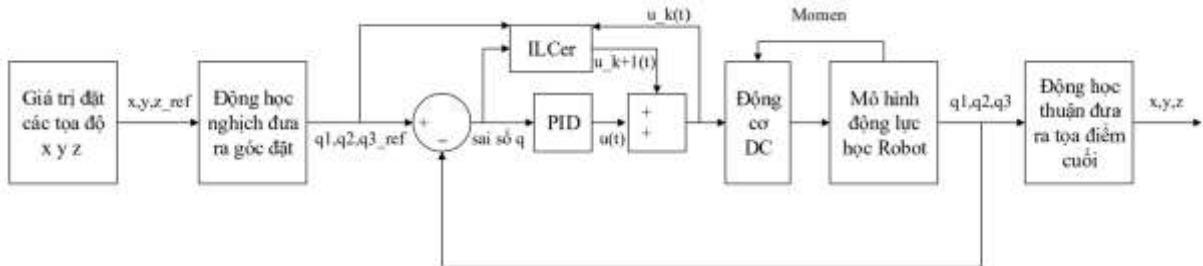
4.1.2. Sơ đồ khối điều khiển mô hình SolidWork và mô hình Động lực học sử dụng bộ điều khiển ILC

Sơ đồ khối mô hình cánh tay Robot 3DOF xây dựng trên SolidWork sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC được thể hiện như hình 4.3.



Hình 4.3: Sơ đồ khối mô hình SolidWork sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC

Sơ đồ khối mô hình động lực học cánh tay Robot 3DOF sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC được thể hiện như hình 4.4.



Hình 4.4: Sơ đồ khối mô hình Động lực học sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC

Các giá trị đặt tọa độ $x y z$ của hình vẽ 2D hoặc 3D mong muốn sẽ được đưa vào khối Động học nghịch để tính toán góc đặt q_a cho các khớp quay. Góc đặt q sẽ được so sánh với góc q_r từ mô hình Robot, sai lệch góc quay sẽ được đưa vào bộ điều khiển PID để đưa ra tín hiệu điều khiển u đưa vào cho mô hình Robot để tính toán và đưa ra góc q_r , tác dụng của việc sử dụng bộ điều khiển PID ở đây là để làm cho giá trị q_r bám với giá trị đặt q_a hay nói cách khác là làm cho hệ thống này ổn định trước khi sử dụng bộ điều khiển ILC. Tiếp theo, sau khi giá trị đầu ra đã bám giá trị đặt thì ta tiến hành kết hợp sử dụng bộ điều khiển PID với bộ điều khiển ILC để giảm sai lệch và làm mượt đáp ứng đầu ra hơn. Cuối cùng giá trị đầu ra q_r được đưa vào khối Động học thuận để tính toán và cho ra tọa độ $x y z$ tại điểm cuối ở khâu chấp hành của cánh tay Robot.

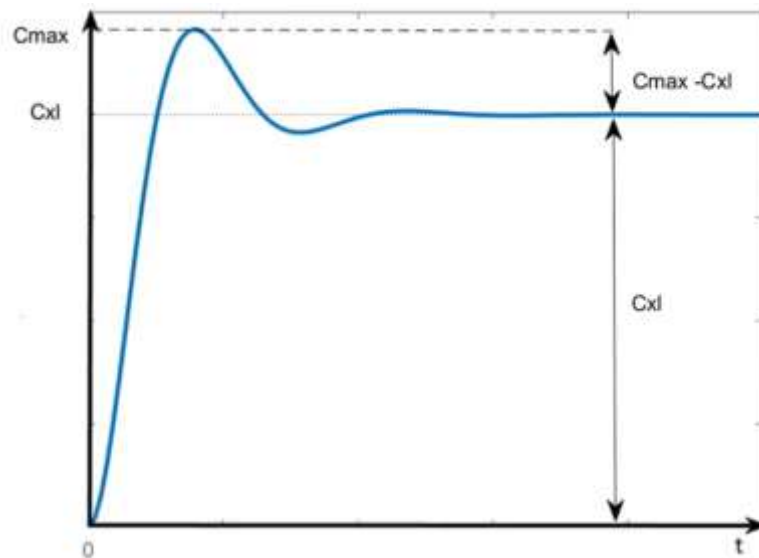
Yêu cầu điều khiển: Đưa điểm cuối ở khâu chấp hành của cánh tay Robot đến đúng vị trí đặt theo thời gian, đảm bảo chuyển động mượt, chính xác và ổn định khi thực hiện vẽ 2D và in 3D. Bộ điều khiển. Bộ điều khiển ILC cần giảm dần sai số giữa đầu ra thực tế và đầu vào mong muốn qua các lần học lặp, đảm bảo không tăng sai số qua các lần lặp, tăng độ mượt và giảm rung cho đầu ra, đặc biệt phải khử được nhiễu từ tác động bên ngoài (rung động, va chạm,...) và giữ cho sai số nhỏ nhất ngay cả khi có nhiễu qua các lần học lặp.

4.1.3. Các chỉ tiêu đánh giá chất lượng của hệ thống

Việc đánh giá các chỉ tiêu chất lượng như độ quá điều chỉnh, thời gian quá độ và sai số xác lập [9] đóng vai trò quan trọng nhằm đảm bảo độ chính xác, ổn định và hiệu quả trong hệ thống điều khiển cánh tay Robot 3 bậc tự do. Các chỉ tiêu này phản ánh khả năng đáp ứng của hệ thống đối với các tín hiệu điều khiển và là cơ sở để đánh giá mức độ phù hợp của bộ điều khiển đối với yêu cầu thực tế.

Trong nghiên cứu lần này, chúng ta sẽ dùng 3 chỉ tiêu (sai số xác lập, độ quá điều chỉnh và thời gian xác lập) để đánh giá chất lượng của hệ thống cánh tay Robot 3 bậc tự do ứng dụng in 3D và vẽ 2D.

- **Độ quá điều chỉnh:** Hiện tượng độ quá điều chỉnh là hiện tượng đáp ứng của hệ thống vượt quá giá trị xác lập của nó như hình 4.5:



Hình 4.5: Đồ thị hiện tượng độ quá điều chỉnh trong đáp ứng quá độ của hệ thống

Để đánh giá hiện tượng độ quá điều chỉnh thì ta dùng tiêu chí độ quá điều chỉnh phần trăm POT là đại lượng đánh giá mức độ quá điều chỉnh của hệ thống, dựa vào đồ thị hình 4.5 ta có công thức tính độ quá điều chỉnh phần trăm POT như sau:

$$POT = \left| \frac{c_{\max} - c_{xl}}{c_{xl}} \right| \times 100\%$$

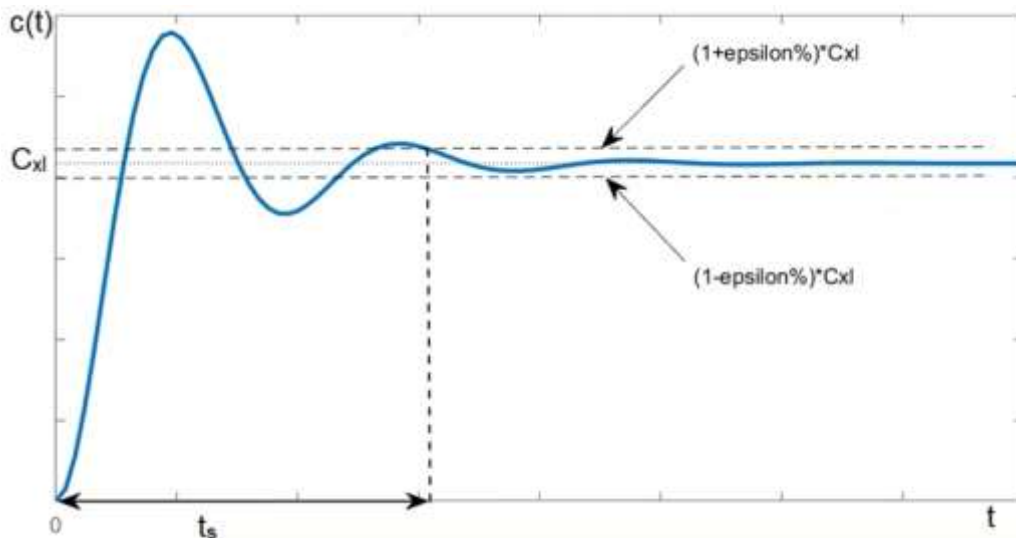
Với:

c_{\max} : giá trị lớn nhất đáp ứng đầu ra của hệ thống.

c_{xl} : giá trị xác lập đáp ứng đầu ra của hệ thống.

Đối với hệ thống cánh tay Robot để đảm bảo cho cánh tay Robot ổn định nhanh nhất có thể thì chúng ta phải chấp nhận một ngưỡng độ quá điều chỉnh cho phép khoảng $POT \leq 60\%$.

- **Thời gian xác lập** t_s : là thời gian cần thiết để sai lệch giữa đáp ứng của hệ thống và giá trị xác lập của nó không vượt quá $\varepsilon\%$ mong muốn như hình 4.6:



Hình 4.6: Độ thị biểu diễn thời gian xác lập của hệ thống

Để xác định được thời gian hệ thống bắt đầu xác lập ta sẽ xác định khoảng biên độ xác lập cho hệ thống với biên độ nằm trong khoảng:

$$\left((1 + \varepsilon\%) \times c_{xl}; (1 - \varepsilon\%) \times c_{xl} \right)$$

Nếu giá trị đáp ứng đầu ra của hệ thống bắt đầu nằm trong khoảng biên độ xác lập thì hệ thống bắt đầu xác lập và thời gian xác lập t_s sẽ được tính từ thời điểm $t = 0$ đến thời điểm hệ thống bắt đầu xác lập.

Với hệ thống điều khiển cánh tay Robot 3 bậc tự do để đảm bảo cho hệ thống đáp ứng nhanh và chính xác thì thời gian xác lập tối thiểu cần đạt được là $t_s \leq 0.15$ giây.

- **Sai số xác lập**: là sai số đáp ứng đầu ra của hệ thống so với giá trị đặt khi hệ thống ở trạng thái xác lập. Để đánh giá chỉ tiêu sai số xác lập ta sử dụng phần trăm sai số xác lập $\varepsilon\%$ được biểu diễn bằng công thức:

$$\varepsilon\% = \left| \frac{y_{xl} - y_{ref}}{y_{ref}} \right| \times 100\%$$

Với:

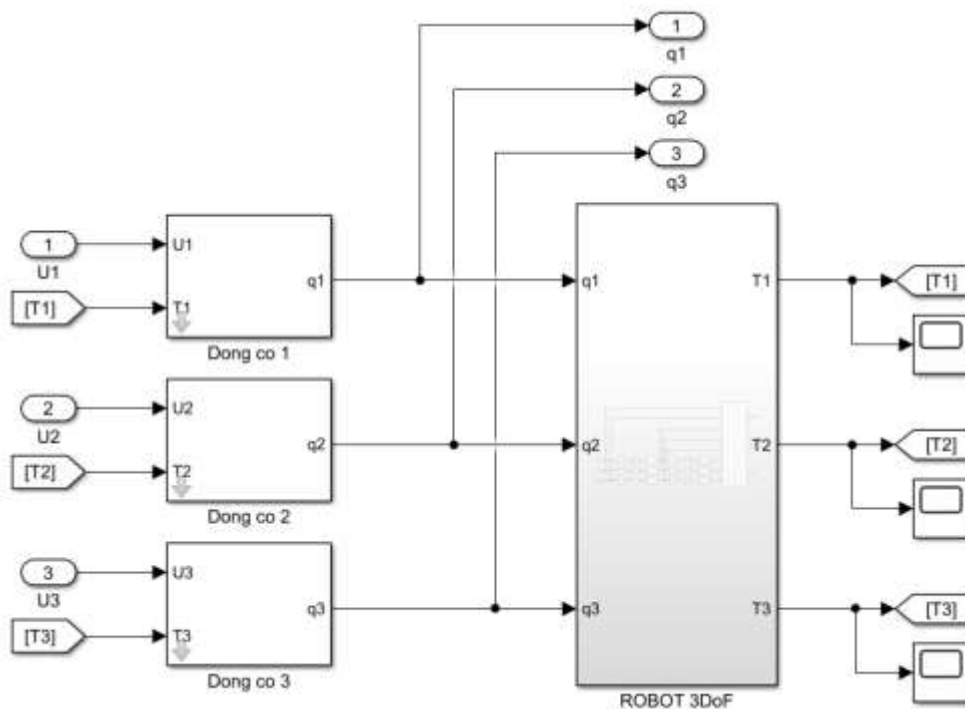
y_{xl} : Giá trị đáp ứng đầu ra của hệ thống khi xác lập.

y_{ref} : Giá trị đầu vào mong muốn của hệ thống.

Để đảm bảo độ chính xác trong các hình vẽ 2D và in 3D thì giá trị phần trăm sai số xác lập của hệ thống cánh tay Robot 3 bậc tự do cần đạt được là $\varepsilon\% \leq 2\%$.

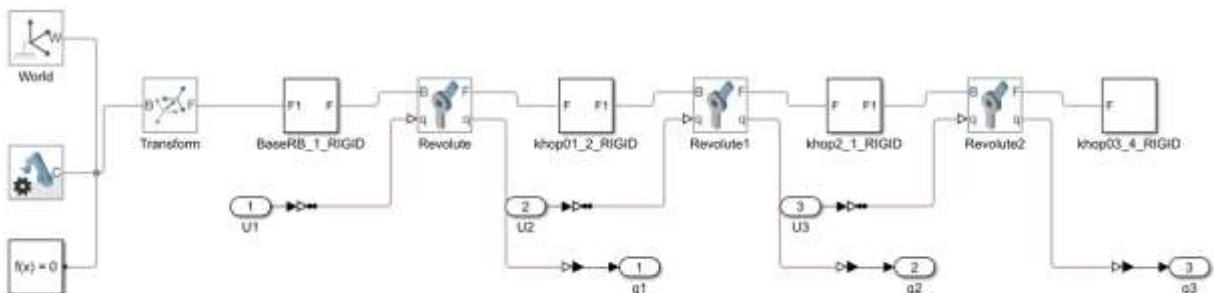
4.2. Kết quả mô phỏng hệ thống điều khiển cánh tay robot 3 DOF sử dụng bộ điều khiển PID

Sau khi tính toán xác định được phương trình động lực học và nhận dạng động cơ cho từng khớp của cánh tay Robot thì ta tiến hành thực hiện mô phỏng mô hình cánh tay Robot 3 bậc tự do trên phần mềm Matlab Simulink như hình 4.7:



Hình 4.7: Sơ đồ mô phỏng mô hình động lực học cánh tay Robot

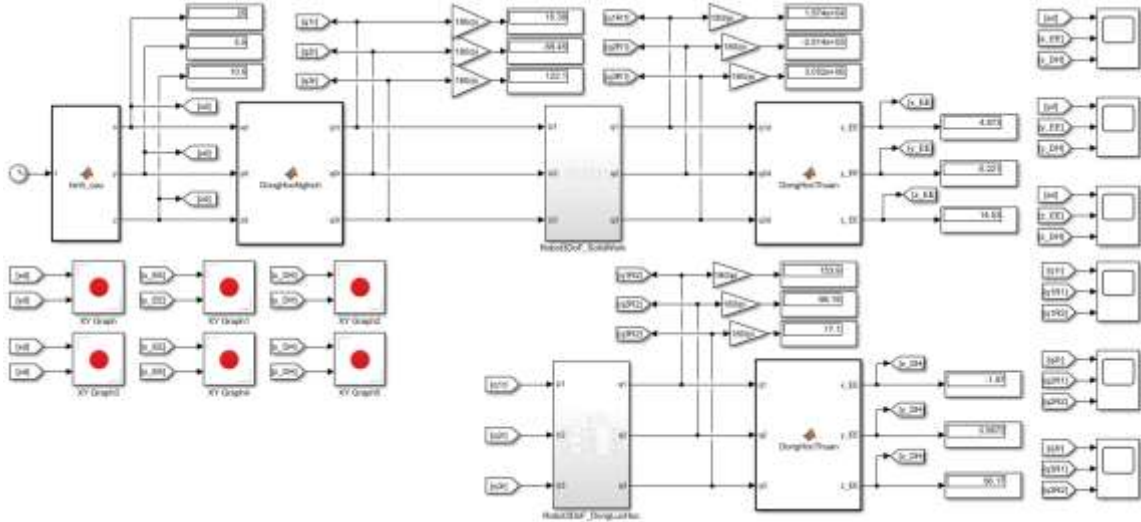
Đối với mô hình cánh tay Robot thiết kế trên SolidWork sau khi đã gắn các khâu với nhau thành một cánh tay Robot hoàn chỉnh thì mô hình sau khi được đưa qua Matlab Simulink như hình 4.8:



Hình 4.8: Sơ đồ mô phỏng cánh tay Robot SolidWork trên Matlab Simulink

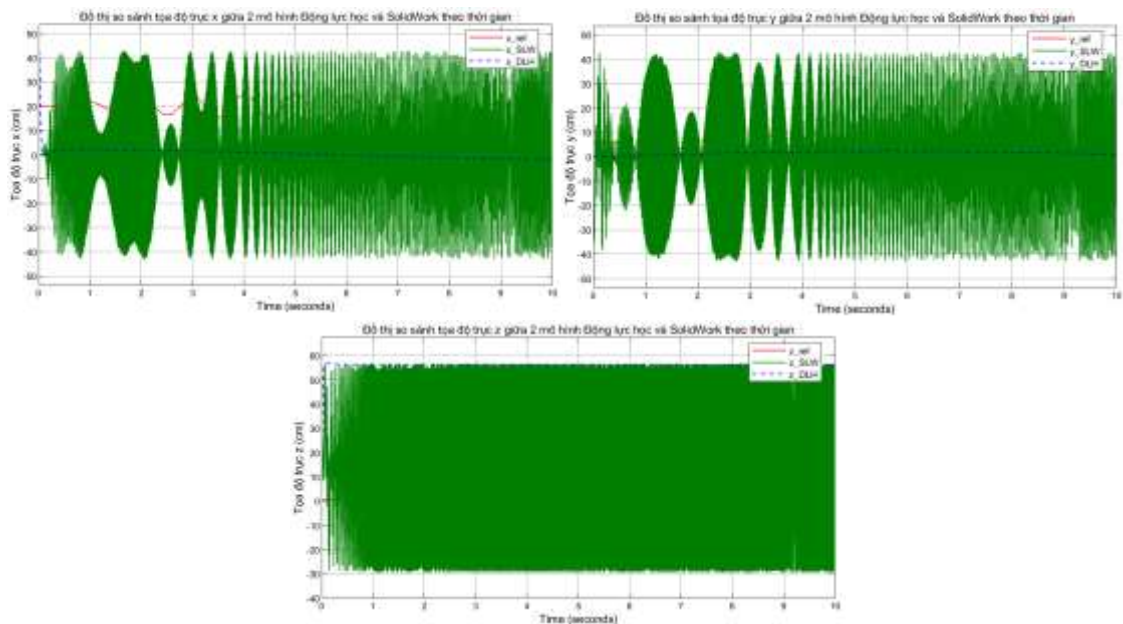
4.2.2. Mô phỏng điều khiển mô hình SolidWork và mô hình Động lực học khi chưa sử dụng bộ điều khiển

Sơ đồ khối mô phỏng cả hai mô hình động lực học và SolidWork khi chưa sử dụng bộ điều khiển như hình 4.9.



Hình 4.9: Sơ đồ mô phỏng điều khiển mô hình SolidWork và Động lực học khi chưa có bộ điều khiển.

- Kết quả mô phỏng cả hai mô hình khi chưa sử dụng bộ điều khiển như hình 4.10.

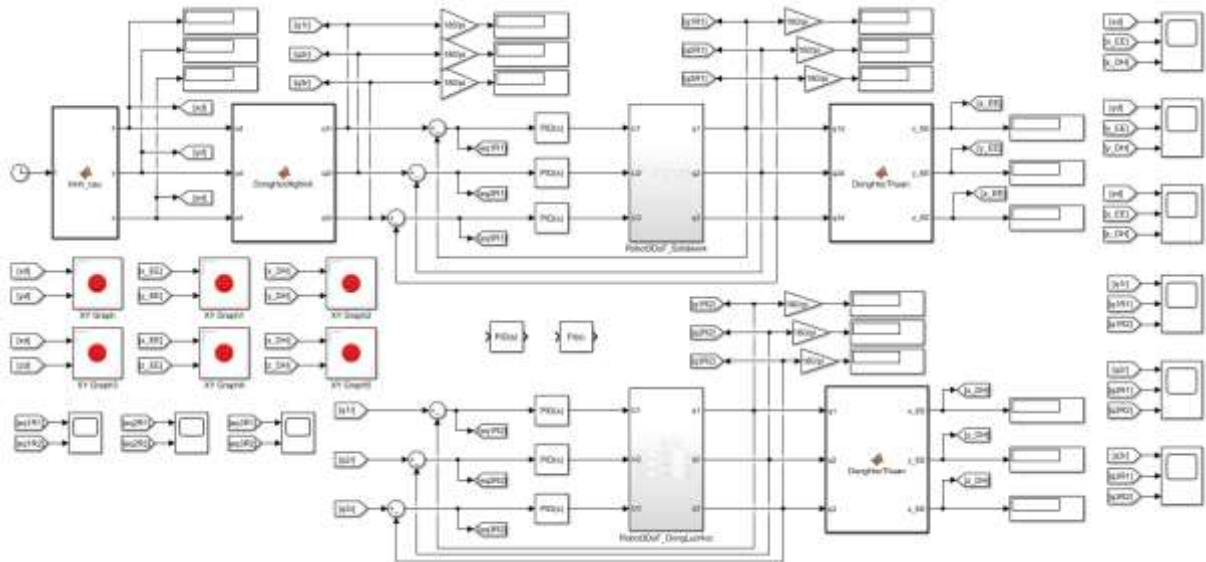


Hình 4.10: Đồ thị tọa độ trục x y z của 2 mô hình SolidWork và Động lực học khi chưa có bộ điều khiển

Nhận xét: Khi chưa sử dụng bộ điều khiển để điều khiển cho 2 mô hình cánh tay Robot thì đồ thị kết quả đầu ra hoàn toàn không bám với giá trị đặt và bị mất ổn định. Vì vậy, chúng ta cần sử dụng bộ điều khiển để điều khiển cánh tay Robot 3 bậc tự do chuyển động ổn định, bám đúng giá trị đặt theo thời gian.

4.2.3. Sơ đồ mô phỏng điều khiển mô hình SolidWork và mô hình Động lực học khi sử dụng chung 1 thông số bộ điều khiển PID

Ở đây, chúng ta sẽ dùng một bộ điều khiển PID chung cho 2 mô hình động lực học cánh tay Robot và mô hình cánh tay Robot trên phần mềm SolidWork để đánh giá xem mô hình động lực học cánh tay Robot mà chúng ta xây dựng đã giống với mô hình thực tế hay chưa.



Hình 4.11: Sơ đồ mô phỏng điều khiển mô hình SolidWork và Động lực học sử dụng bộ điều khiển PID

4.2.4. Kết quả mô phỏng điều khiển mô hình SolidWork và mô hình Động lực học khi sử dụng chung 1 thông số bộ điều khiển PID

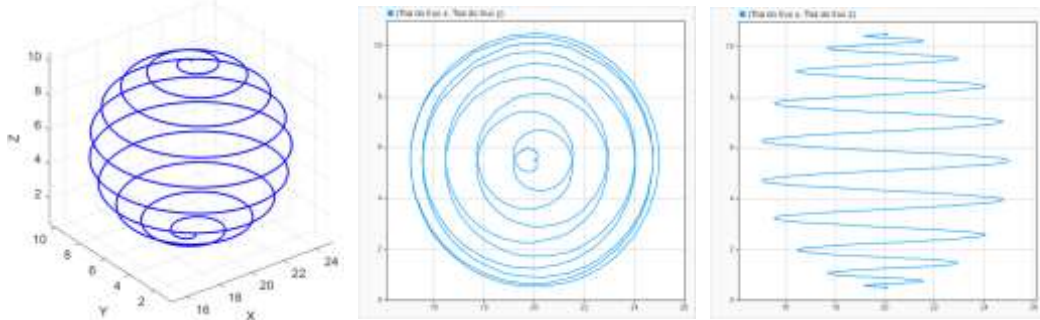
Bằng cách sử dụng phương pháp tinh chỉnh bộ điều khiển PID ở chương 2 thì chúng ta lựa chọn được thông số bộ điều khiển PID cho 2 mô hình là:

$$K_p = 664.23; K_i = 36020.25; K_d = 1.985$$

4.2.4.1 Kết quả mô phỏng in 3D

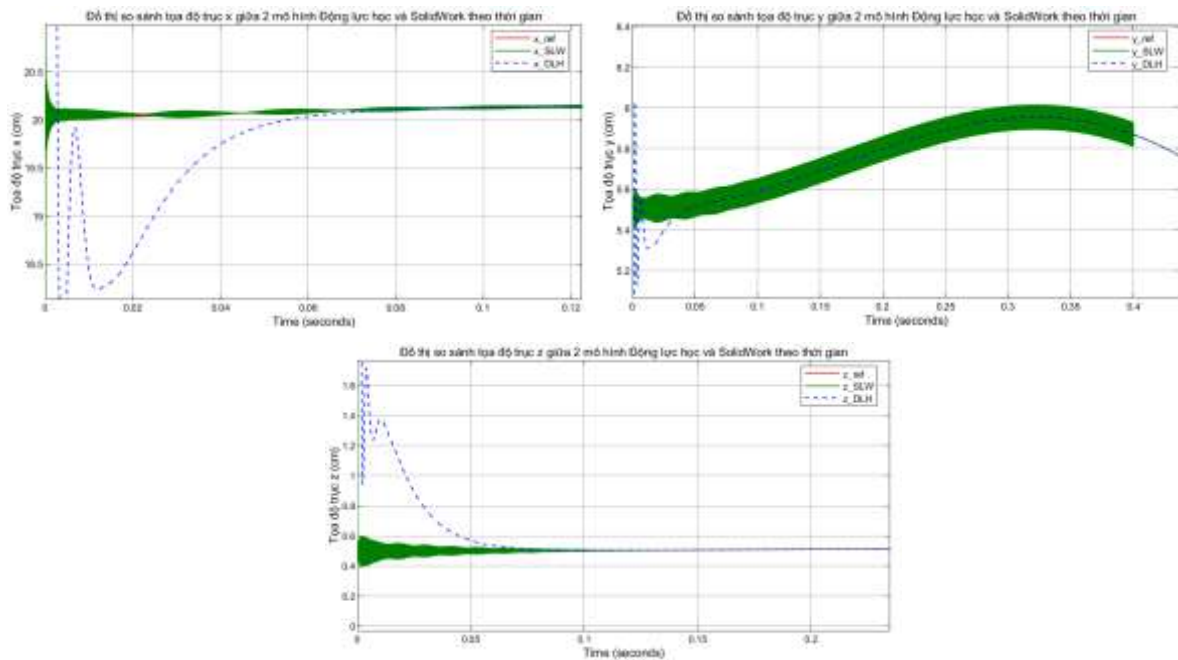
Quỹ đạo hình in 3D đầu vào cho mô hình cánh tay Robot gồm hình trong tọa độ không gian Oxyz (hình bên trái), hình trong tọa độ mặt phẳng Oxy (hình ở giữa) và hình trong tọa độ mặt phẳng Oxz (hình bên phải) như trong hình 4.12.

Quy đạo xoắn ốc trên mặt cầu với tâm (20, 0, 5)



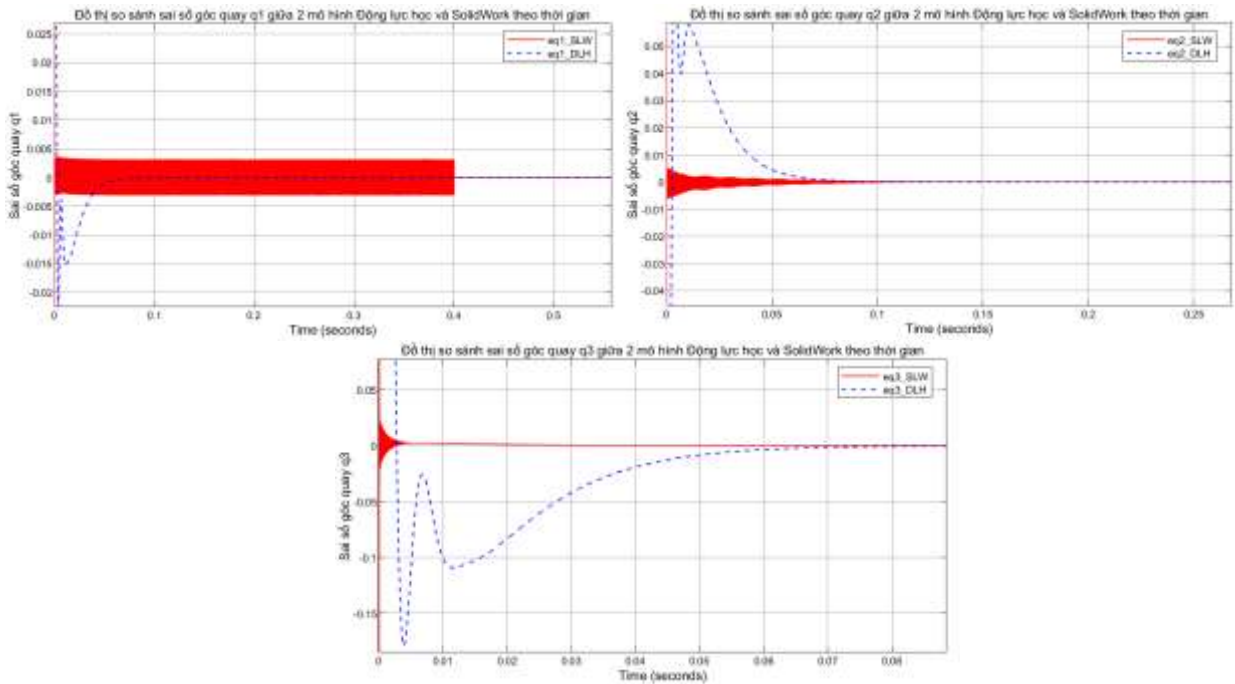
Hình 4.12: Hình in 3D mong muốn

- Kết quả tọa độ x y z khi thực hiện in 3D sử dụng chung bộ điều khiển PID cho cả hai mô hình cho thấy các tọa độ x y z đầu ra của mô hình SolidWork dao động với tần số rất cao như hình 4.13.



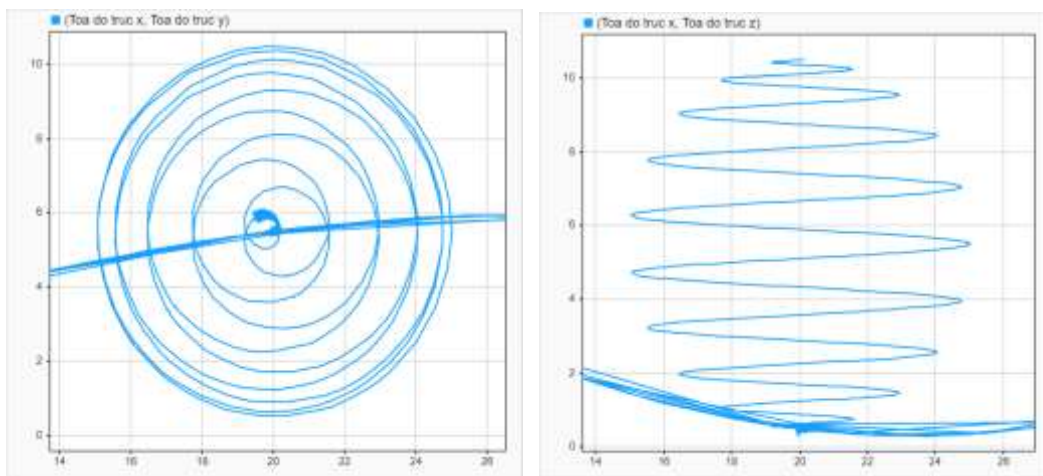
Hình 4.13: Đồ thị tọa độ x y z đầu ra cho hình in 3D của 2 mô hình sử dụng chung 1 bộ điều khiển PID

- Kết quả sai lệch góc quay trong hình 4.15 cho thấy tại mô hình SolidWork thì sai lệch bị dao động với tần số cao.



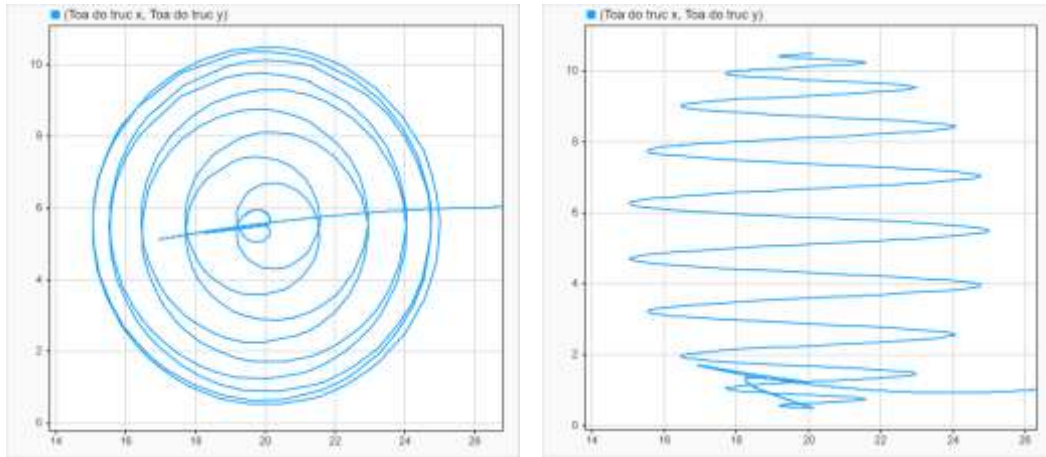
Hình 4.14: Đồ thị sai số góc quay cho hình in 3D của 2 mô hình sử dụng chung 1 bộ điều khiển PID

- Kết quả đồ thị mặt phẳng Oxy (hình bên trái) và Oxz (hình bên phải) của mô hình SolidWork như trong hình 4.15 cho thấy các nét dao động mạnh.



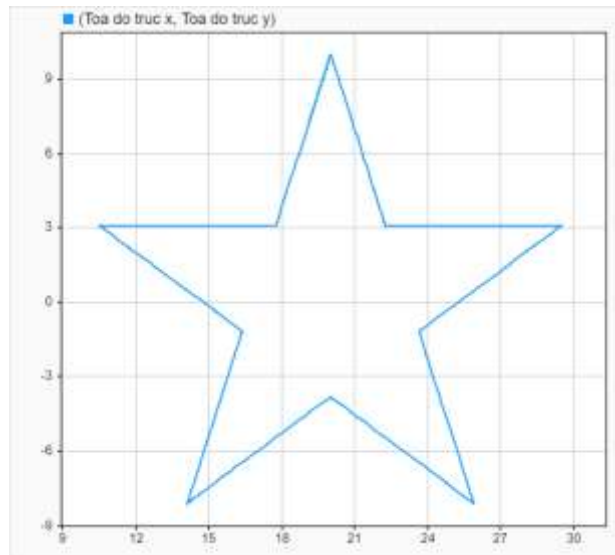
Hình 4.15: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork

- Kết quả đồ thị mặt phẳng Oxy (hình bên trái) và Oxz (hình bên phải) của mô hình động lực học như trong hình 4.16 cho thấy các nét dao động ban đầu rất nhẹ nhàng chứng tỏ cánh tay robot ít bị giật lúc đầu hơn mô hình SolidWork.



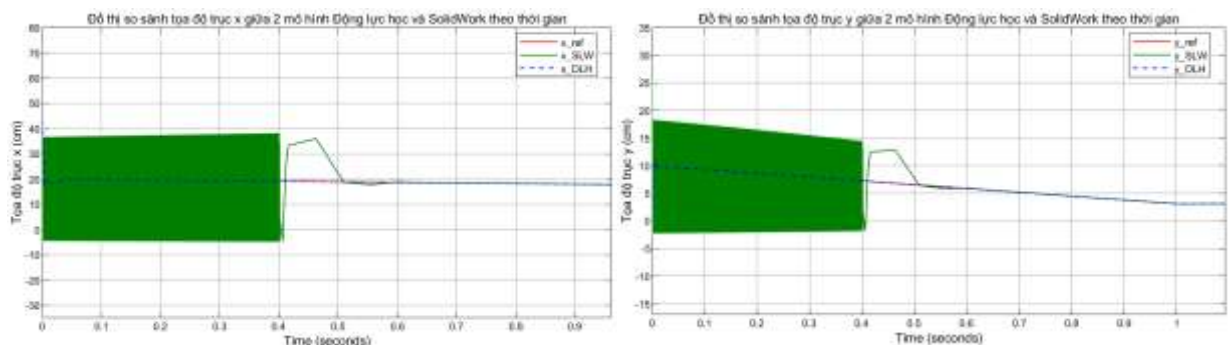
Hình 4.16: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học

4.2.4.2 Kết quả mô phỏng vẽ 2D



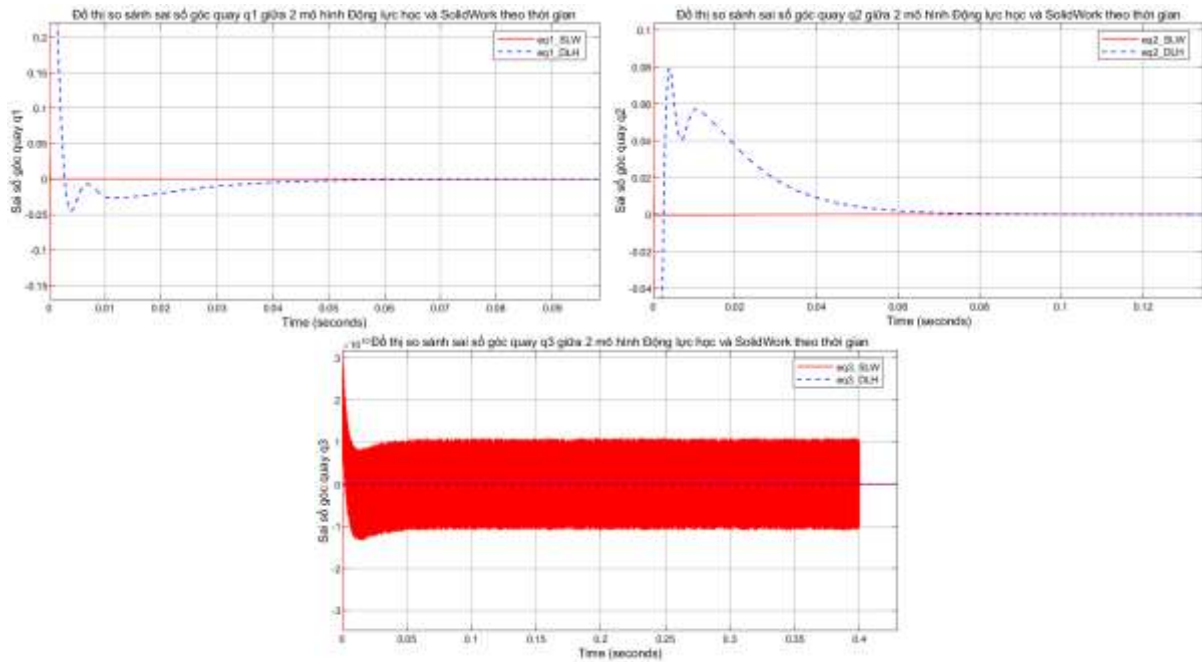
Hình 4.17: Hình vẽ 2D mong muốn

- Kết quả đồ thị tọa độ x và y khi thực hiện vẽ 2D như hình 4.18 cho thấy các tọa độ đầu ra x và y bị dao động mạnh ở mô hình SolidWork.



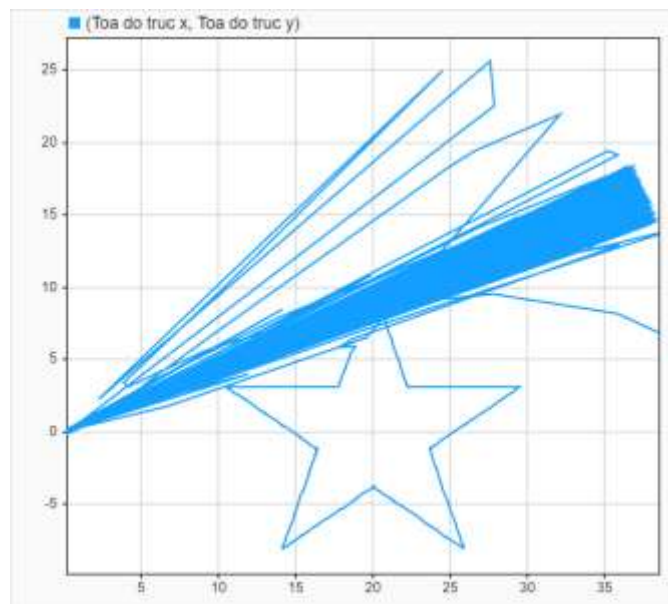
Hình 4.18: Đồ thị tọa độ đầu ra cho hình vẽ 2D của 2 mô hình sử dụng chung 1 bộ điều khiển PID

- Kết quả sai lệch các khớp như hình 4.19 cho thấy tại khớp 3 sai lệch rất cao.



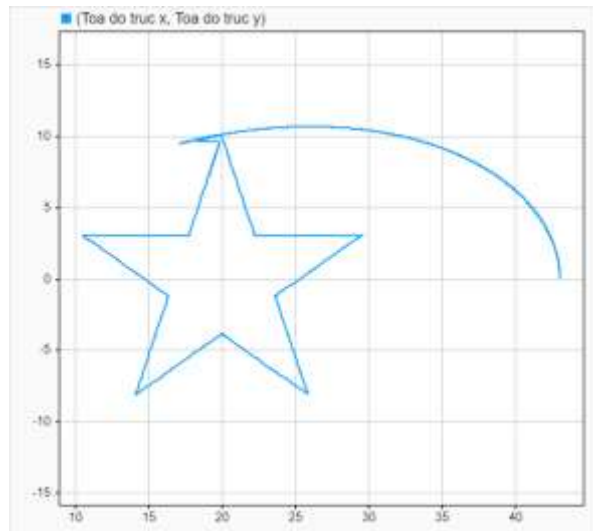
Hình 4.19: Đồ thị sai số góc quay khi vẽ 2D của 2 mô hình sử dụng chung 1 bộ điều khiển PID

- Kết quả hình vẽ 2D đầu ra của mô hình SolidWork như hình 4.20 cho thấy cánh tay Robot đã bị giật rất mạnh lúc bắt đầu vẽ.



Hình 4.20: Đồ thị kết quả vẽ 2D của mô hình SolidWork

- Kết quả hình vẽ 2D đầu ra của mô hình động lực học như hình 4.21 cho thấy cánh tay Robot chuyển động nhẹ về vị trí bắt đầu vẽ ít bị giật hơn mô hình SolidWork.



Hình 4.21: Đồ thị kết quả vẽ 2D của mô hình động lực học

Nhận xét: Mô hình động lực học cánh tay Robot 3 bậc tự do đã được xây dựng gần giống với thực tế khi cả 2 mô hình động lực học cánh tay Robot và mô hình xây dựng trên SolidWork đều cho ra đồ thị bám với giá trị đặt khi sử dụng chung 1 thông số bộ điều khiển PID. Tuy nhiên, đồ thị đầu ra của mô hình xây dựng trên SolidWork bị dao động mạnh lúc đầu do thông số bộ điều khiển PID được dùng ở đây là bộ thông số được lựa chọn tối ưu dựa trên mô hình động lực học chưa phải là bộ thông số PID tối ưu nhất cho mô hình SolidWork, vì thế nên đồ thị đầu ra của 2 mô hình sẽ có sự khác biệt. Nguyên nhân dẫn đến sự khác biệt này là khi xây dựng mô hình động lực học cho cánh tay Robot 3 bậc tự do thì có một vài thông số của động cơ chúng ta phải lựa chọn những tham số gần đúng, mô hình động lực học bỏ qua ma sát, độ trễ, bên cạnh đó mô hình trên SolidWork thì bao gồm cả ma sát, momen quán tính có thể không đối xứng. Những nguyên nhân đó dẫn đến sự khác nhau giữa 2 mô hình.

Đề xuất hướng đi tiếp theo: Để đảm bảo việc mô phỏng trong đồ án này tạo ra kết quả tốt nhất thì chúng ta nên sử dụng bộ điều khiển tối ưu nhất của từng mô hình để thực hiện mô phỏng, tránh tạo ra những trở ngại khi kết hợp bộ điều khiển PID với bộ điều khiển ILC ở phần sau.

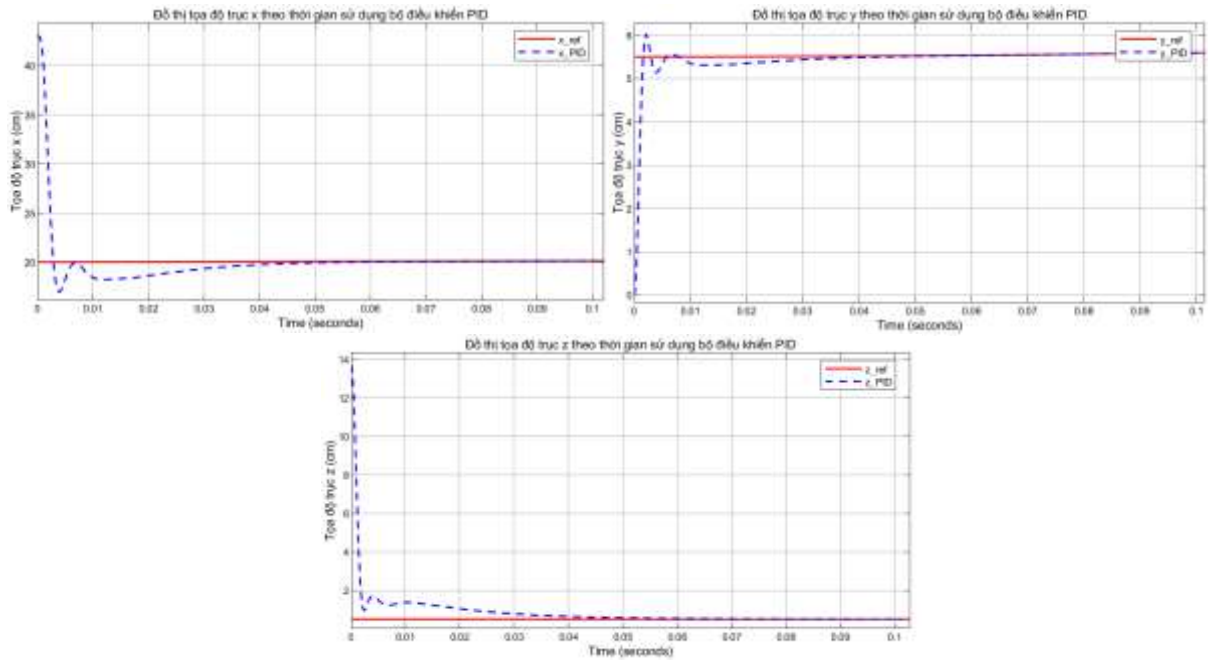
4.2.5. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID

Bằng cách sử dụng phương pháp tinh chỉnh bộ điều khiển PID ở chương 2 thì chúng ta lựa chọn được thông số bộ điều khiển PID cho mô hình Động lực học là:

$$K_p = 664.23; K_i = 36020.25; K_d = 1.985$$

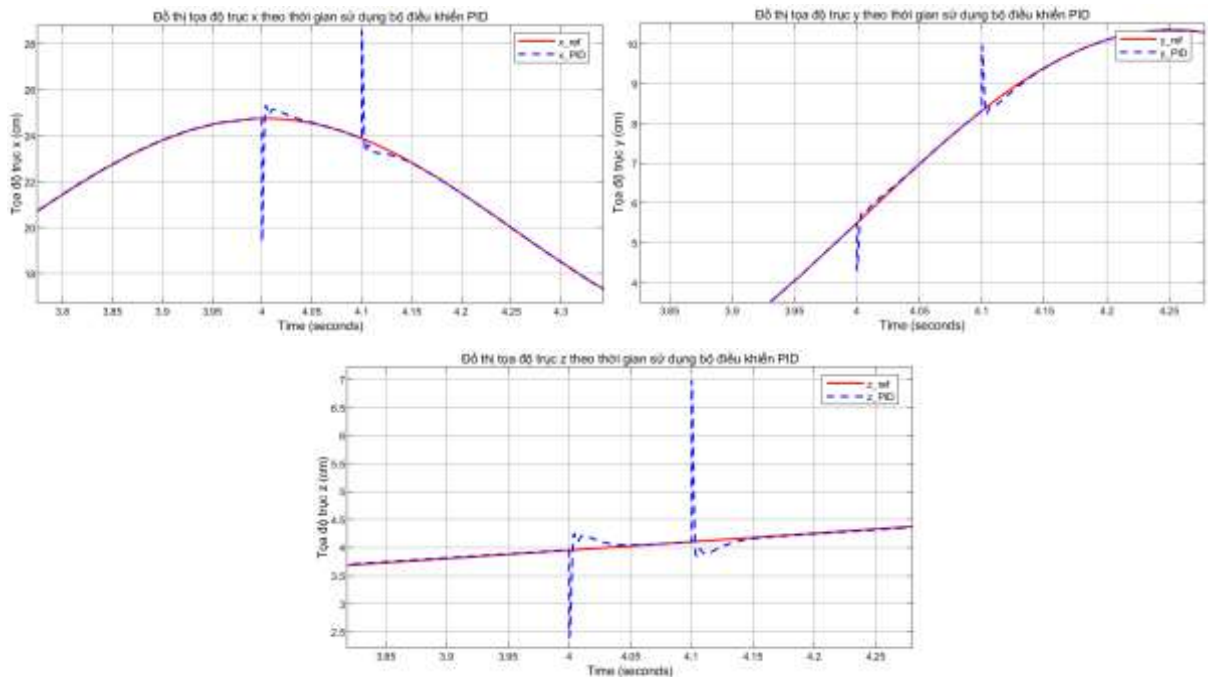
4.2.5.1 Kết quả mô phỏng in 3D khi không có nhiễu và khi có nhiễu tác động

- Kết quả đồ thị đầu ra của các trục tọa độ x y z khi không có nhiễu như hình 4.22.



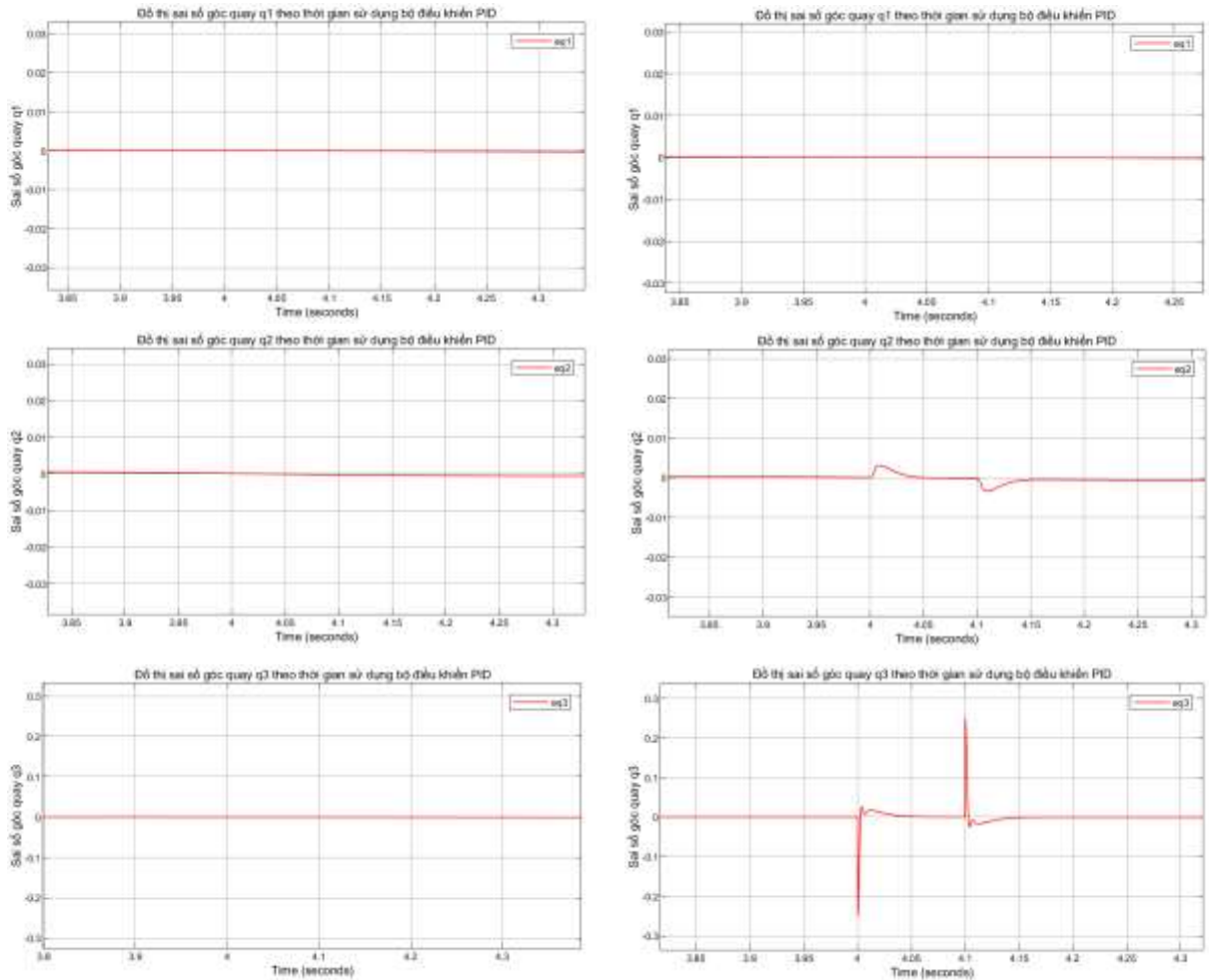
Hình 4.22: Đồ thị tọa độ $x y z$ đầu ra cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu

- Khi có nhiễu tác động vào góc quay q_3 như trong hình 4.23, tại thời điểm nhiễu tác động thì đồ thị tọa độ $x y z$ đều bị ảnh hưởng. Bộ điều khiển PID đã xử lý nhiễu khá tốt, bám nhanh về vị trí cũ và ổn định ngay, tuy nhiên nếu nhiễu này cứ tiếp diễn qua các lần in hoặc vẽ tiếp theo thì bộ PID không thể khử nhiễu hoàn toàn được.



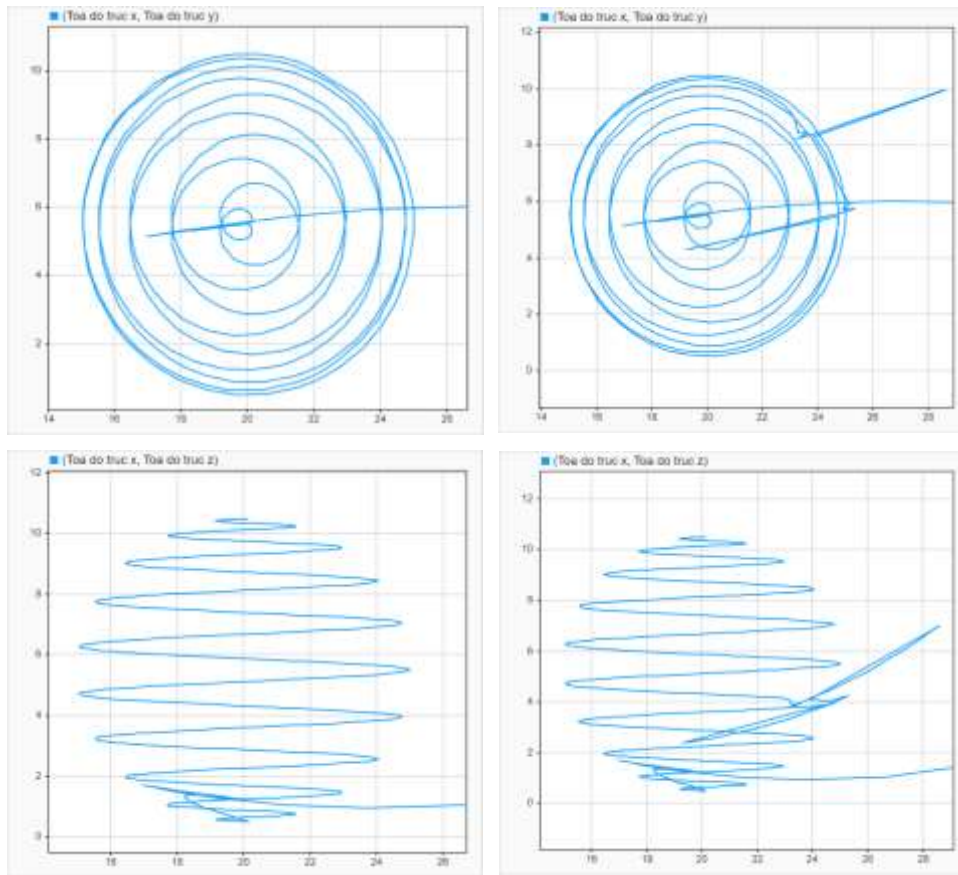
Hình 4.23: Đồ thị tọa độ $x y z$ đầu ra cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu

- Sai lệch góc quay bị ảnh hưởng nhiều nhất là tại góc quay q_3 do nhiễu tác động trực tiếp vào góc quay q_3 , còn sai số 2 góc quay còn lại ít bị ảnh hưởng hơn do nhiễu chỉ tác động vào khâu chấp hành như trong hình 4.24.



Hình 4.24: Đồ thị sai lệch góc quay cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu

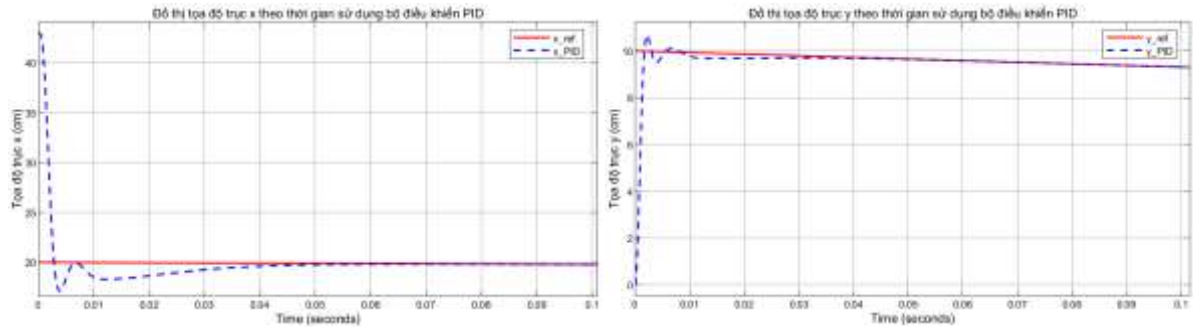
- Khi có nhiễu (hình 4.25 bên phải) tác động vào 1 khớp thì hình in 3D sau khi in ra sẽ bị lỗi tại thời điểm có nhiễu so với lúc không có nhiễu. Cần cải thiện nhiễu qua các lần in tiếp theo nếu còn nhiễu như này.



Hình 4.25: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học khi có nhiễu và khi không có nhiễu

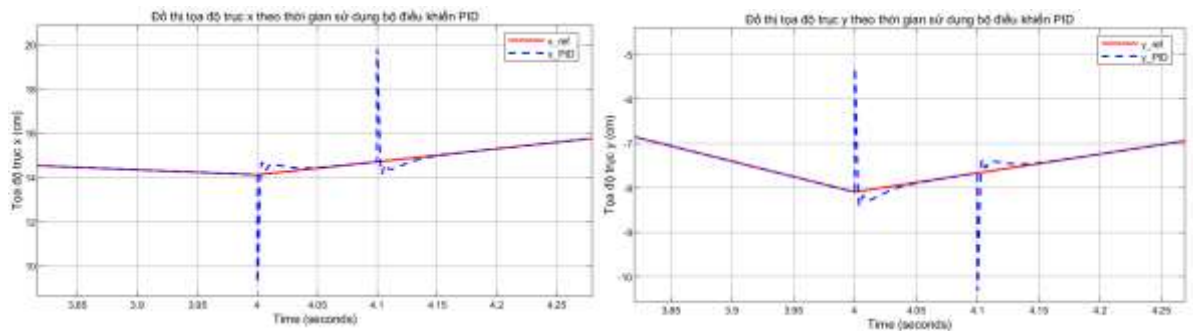
4.2.5.2 Kết quả mô phỏng vẽ 2D khi không có nhiễu và khi có nhiễu tác động

- Kết quả đồ thị đầu ra của các trục tọa độ x y khi không có nhiễu như hình 4.26.



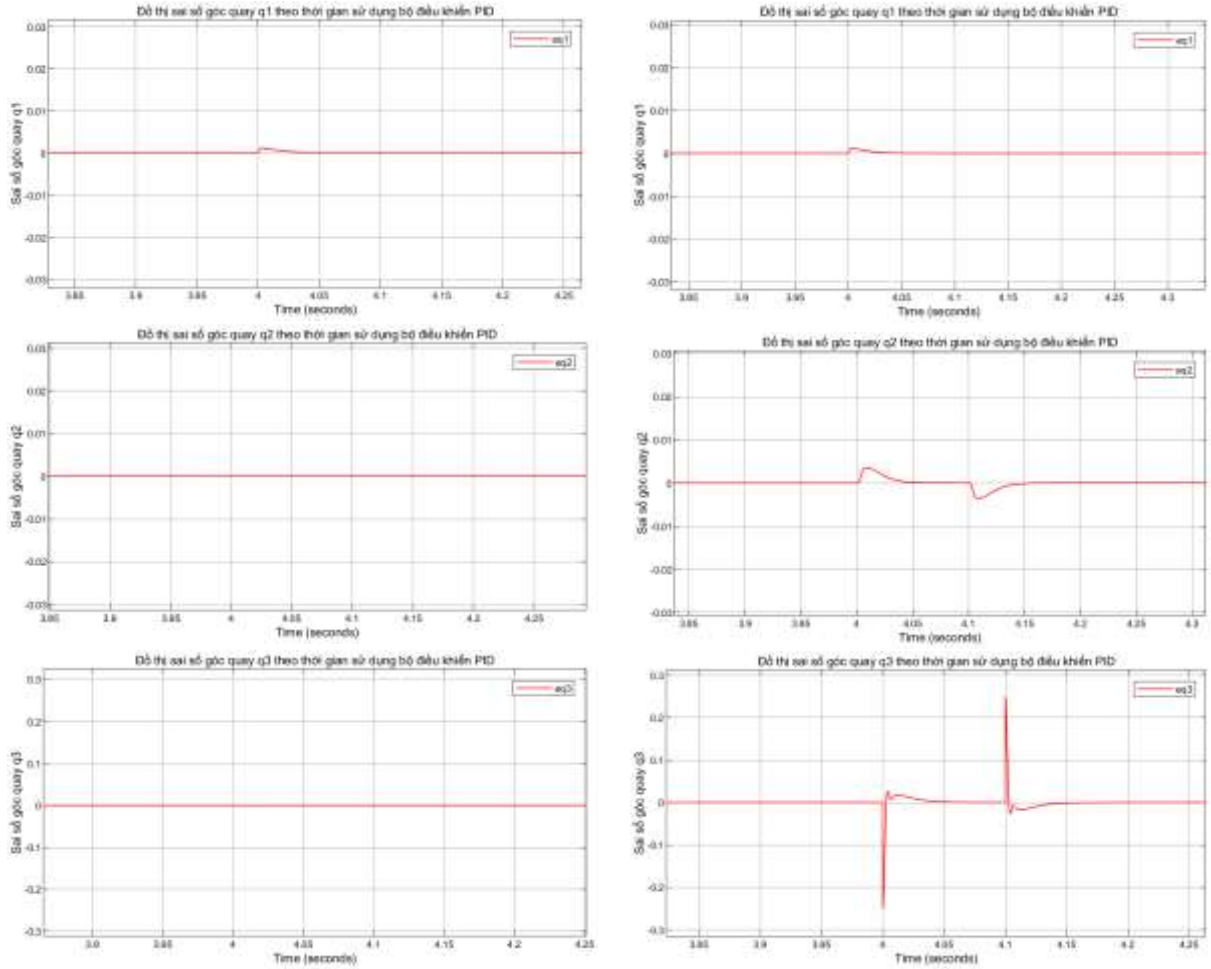
Hình 4.26: Đồ thị tọa độ x y đầu ra cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu

- Khi có nhiễu tác động vào góc quay q_3 như trong hình 4.27, tại thời điểm nhiễu tác động thì đồ thị tọa độ x y đều bị ảnh hưởng. Bộ điều khiển PID đã xử lý nhiễu khá tốt, bám nhanh về vị trí cũ và ổn định ngay, tuy nhiên nếu nhiễu này cứ tiếp diễn qua các lần in hoặc vẽ tiếp theo thì bộ PID không thể khử nhiễu hoàn toàn được.



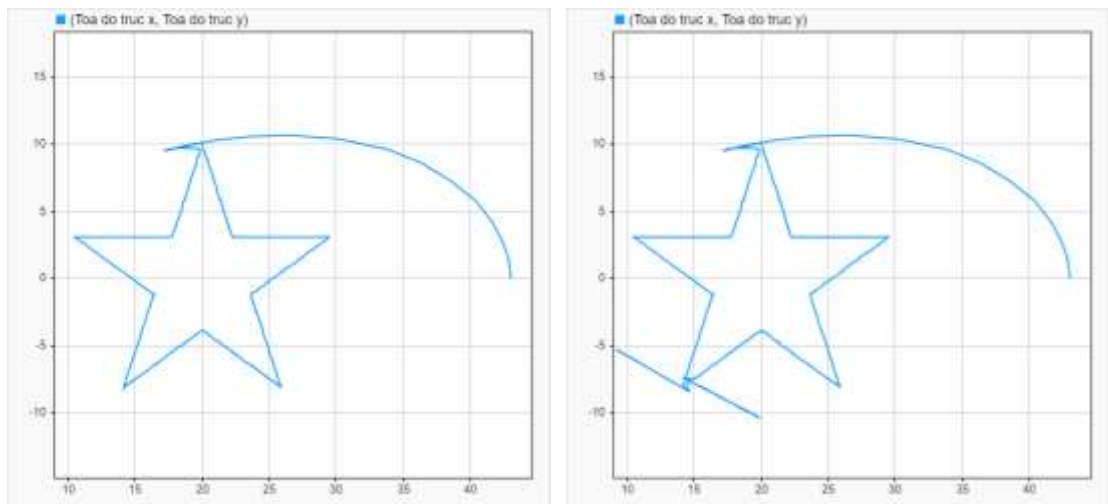
Hình 4.27: Đồ thị tọa độ x y đầu ra cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu

- Sai lệch góc quay (hình 4.28) bị ảnh hưởng nhiều nhất là tại góc quay q_3 do nhiễu tác động trực tiếp vào góc quay q_3 , còn sai số 2 góc quay còn lại ít bị ảnh hưởng hơn do nhiễu chỉ tác động vào khâu chấp hành.



Hình 4.28: Đồ thị sai số góc quay cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu

- Khi có nhiễu (hình 4.29 bên phải) tác động vào 1 khớp thì hình vẽ 2D sau khi vẽ ra sẽ bị lỗi tại thời điểm có nhiễu so với lúc không có nhiễu. Cần cải thiện nhiễu qua các lần in tiếp theo nếu còn nhiễu như này.



Hình 4.29: Đồ thị kết quả vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu

4.2.5.3 Đánh giá chất lượng hệ thống khi sử dụng bộ điều khiển PID

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb}
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.
- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0383	0.0069	0.0345%	15.24%
Trục y	0.0258	0.0018	0.0322%	9.58%
Trục z	0.0808	0.0065	0.1174%	241.8%

Bảng 4.1: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình động lực học khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0383	0.0037	0.0189%	14.56%
Trục y	0.0191	0.002	0.3474%	7.06%

Bảng 4.2: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình động lực học khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình động lực học cánh tay Robot 3 bậc tự do thực hiện in 3D và vẽ 2D thể hiện trong Bảng 4.1 và Bảng 4.2, có thể nhận thấy rằng thời gian xác lập t_s và phần trăm sai số xác lập trung bình $\varepsilon\%$ đều đáp ứng tốt các tiêu chí đã đề ra. Tuy nhiên, phần trăm độ quá điều chỉnh $POT(\%)$ của trục Z vẫn còn tương đối cao và cần được cải thiện.
- Mặt khác, các kết quả này được thu thập trong điều kiện mô phỏng không có nhiễu. Khi hệ thống chịu tác động nhiễu tại khâu 3, gây ra sai lệch góc tại khớp 3, bộ điều khiển PID vẫn cho thấy khả năng đáp ứng nhanh và duy trì tính ổn định cho hệ thống. Tuy vậy, bộ điều khiển PID không thể hoàn toàn loại bỏ ảnh hưởng của nhiễu, dẫn đến gia tăng sai số. Nguyên nhân là do PID không có khả năng ước lượng hay bù nhiễu. Do đó, để cải thiện khả năng khử nhiễu và nâng cao độ chính xác, cần xem xét kết hợp thêm bộ điều khiển ILC vào hệ thống.

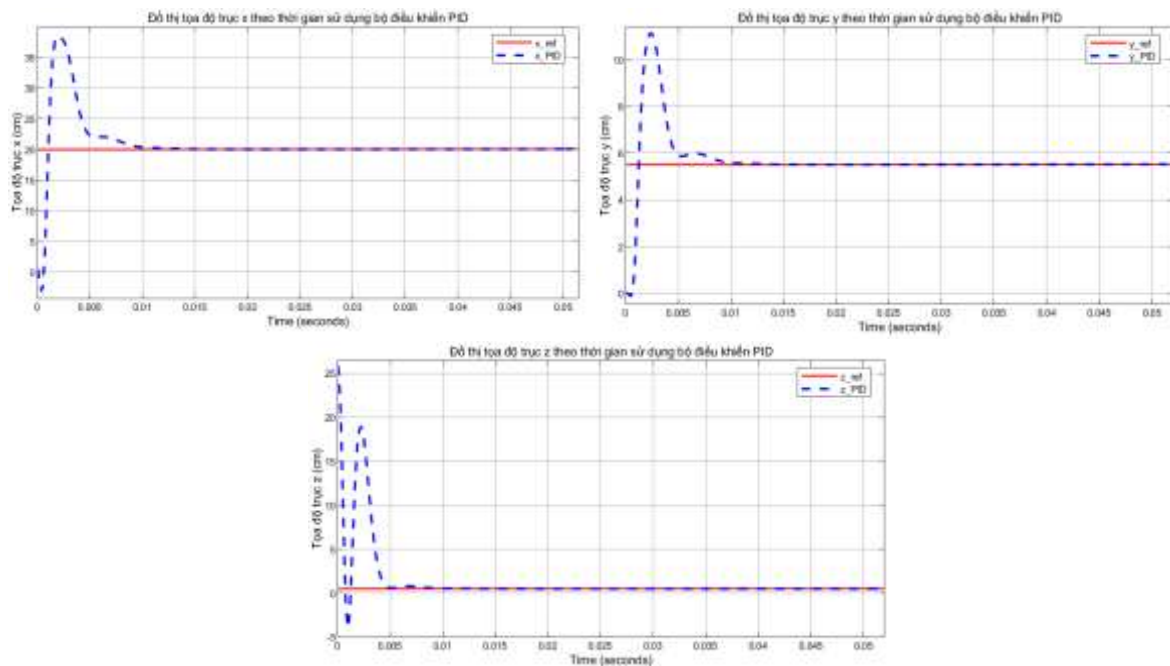
4.2.6. Kết quả mô phỏng điều khiển mô hình xây dựng trên SolidWork khi sử dụng thông số bộ điều khiển PID

Bằng cách sử dụng phương pháp tinh chỉnh bộ điều khiển PID ở chương 2 thì chúng ta lựa chọn được thông số bộ điều khiển PID cho mô hình xây dựng trên SolidWork là:

$$K_p = 1.9152; K_i = 1010.485$$

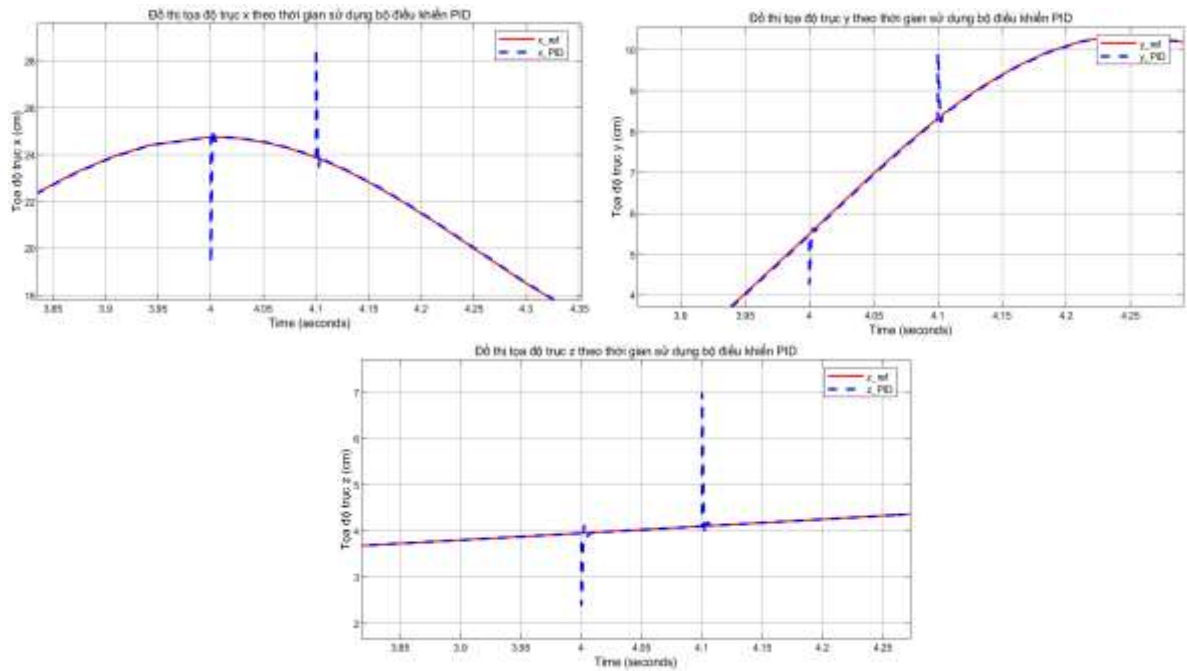
4.2.6.1 Kết quả mô phỏng in 3D khi không có nhiễu và khi có nhiễu tác động

- Kết quả đồ thị đầu ra của các trục tọa độ x y z khi không có nhiễu như hình 4.30.



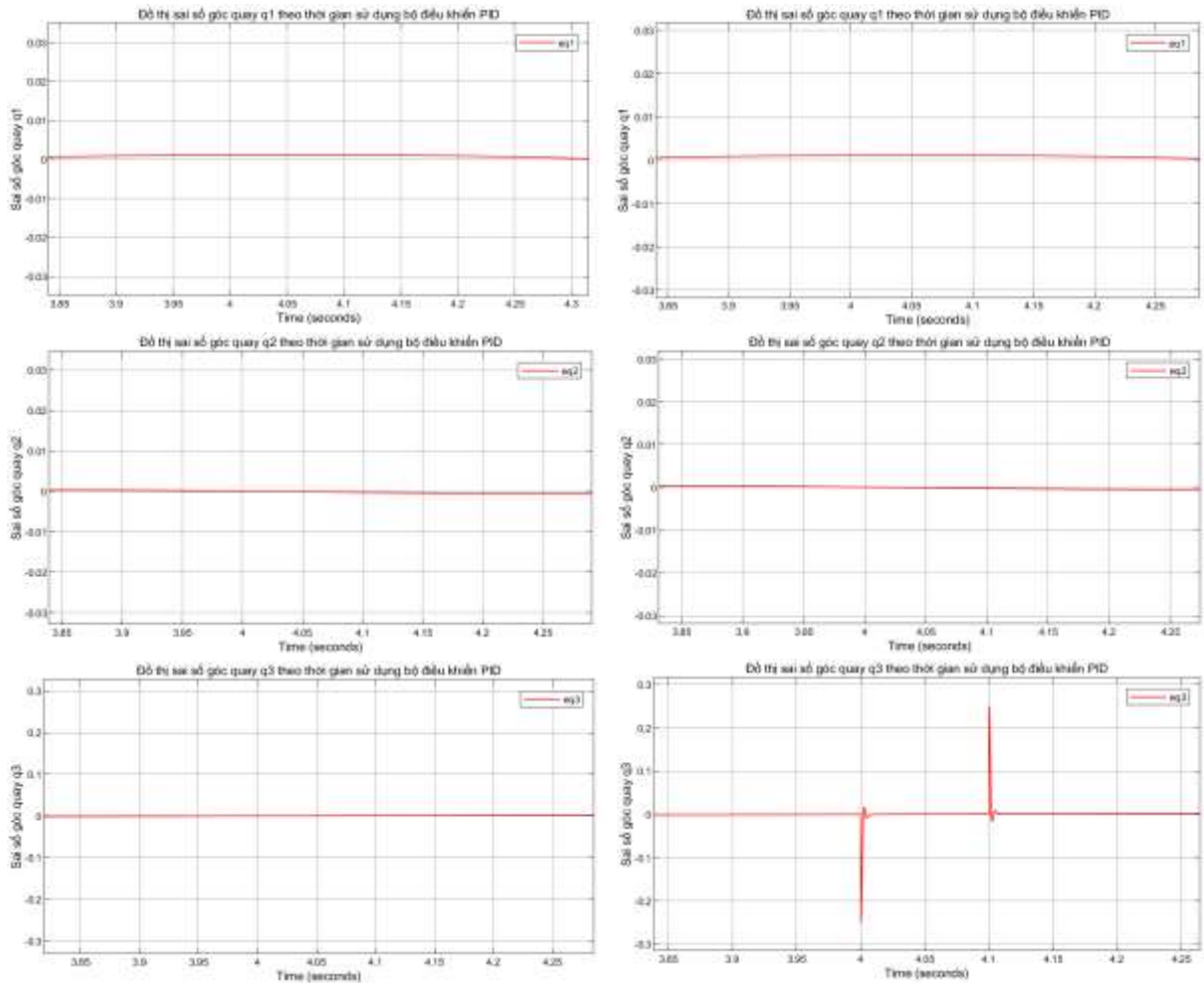
Hình 4.30: Đồ thị tọa độ x y z đầu ra cho hình in 3D của mô hình SolidWork sử dụng bộ điều khiển PID khi không có nhiễu

- Cũng giống như mô hình động lực học, thì khi có nhiễu tác động vào góc quay q_3 như trong hình 4.31, tại thời điểm nhiễu tác động thì đồ thị tọa độ x y z đều bị ảnh hưởng. Bộ điều khiển PID đã xử lý nhiễu khá tốt, bám nhanh về vị trí cũ và ổn định ngay, tuy nhiên nếu nhiễu này cứ tiếp diễn qua các lần in hoặc vẽ tiếp theo thì bộ PID không thể khử nhiễu hoàn toàn được.



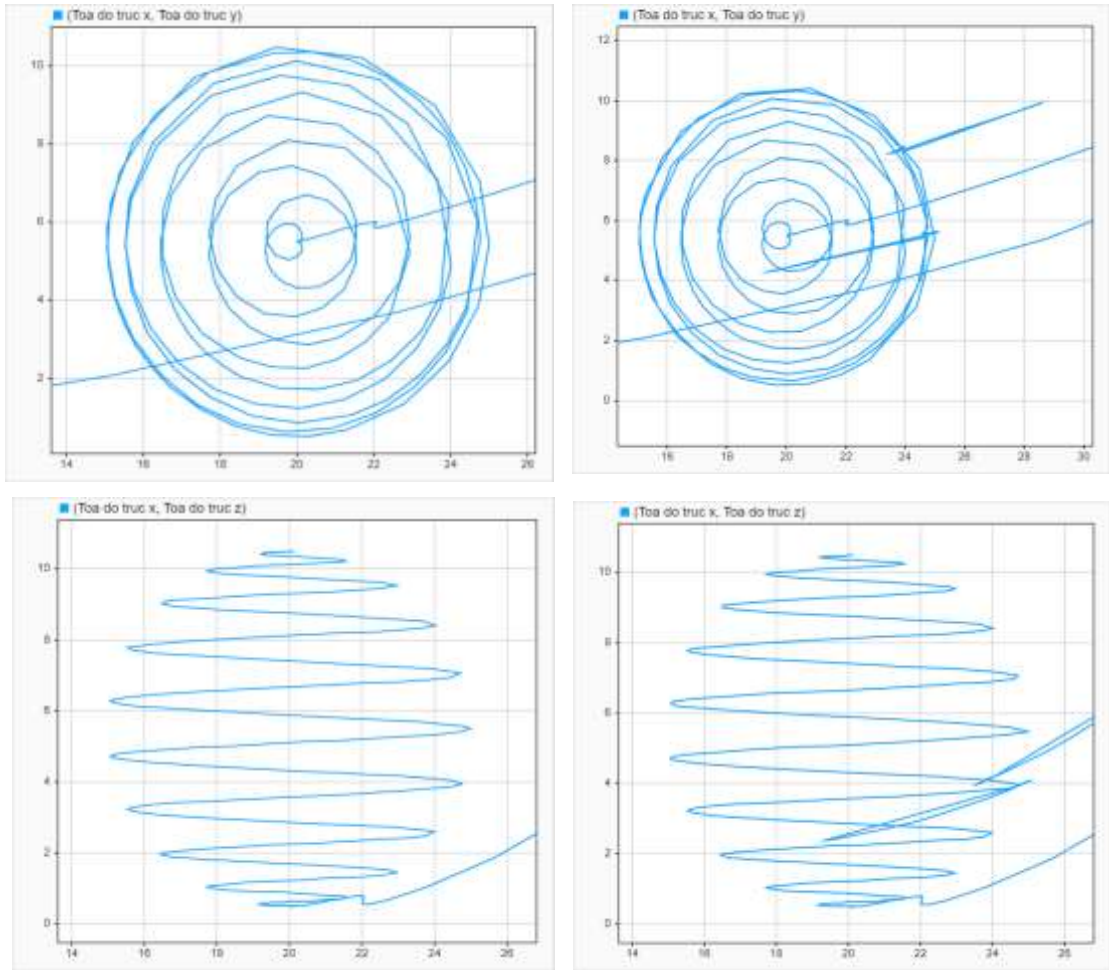
Hình 4.31: Đồ thị tọa độ x y z đưa ra cho hình in 3D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu

- Cũng như mô hình động lực học, thì sai lệch góc quay bị ảnh hưởng nhiều nhất là tại góc quay q_3 do nhiễu tác động trực tiếp vào góc quay q_3 , còn sai số 2 góc quay còn lại ít bị ảnh hưởng hơn do nhiễu chỉ tác động vào khâu chấp hành như trong hình 4.32.



Hình 4.32: Đồ thị sai lệch góc quay cho hình in 3D của mô hình SolidWork sử dụng bộ điều khiển PID khi không có nhiễu và khi có nhiễu

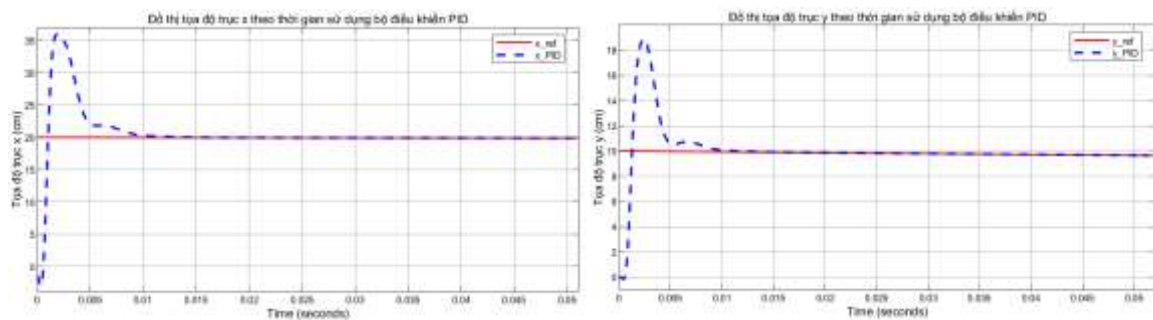
- Đối với đồ thị không có nhiễu (hình 4.33 bên trái), đường đi của khâu chấp hành khá đều, cho thấy sự điều khiển mượt mà và ổn định. Còn khi có nhiễu (hình 4.33 bên phải), quỹ đạo đường đi của khâu chấp hành bị méo, lệch ra ngoài, làm hỏng độ chính xác của hình in 3D.



Hình 4.33: Đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork khi không có nhiễu và khi có nhiễu

4.2.6.2 Kết quả mô phỏng vẽ 2D khi không có nhiễu tác động

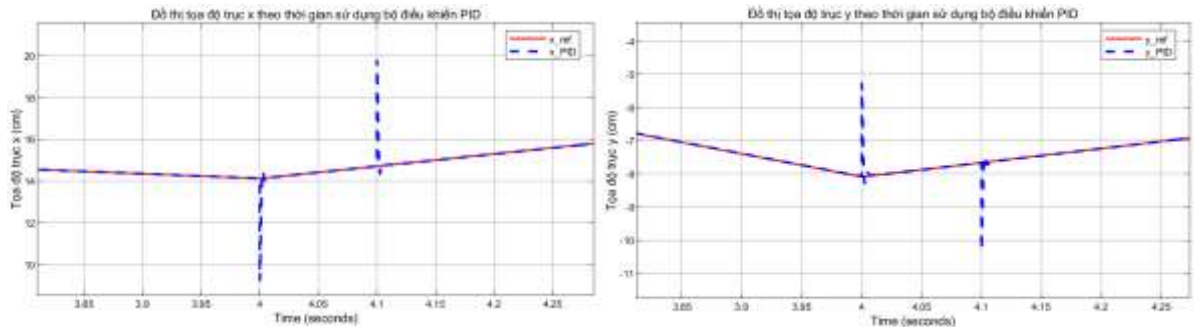
- Kết quả đồ thị đầu ra của các trục tọa độ x y khi không có nhiễu như hình 4.34.



Hình 4.34: Đồ thị tọa độ đầu ra cho hình vẽ 2D của mô hình SolidWork sử dụng bộ điều khiển PID khi không có nhiễu

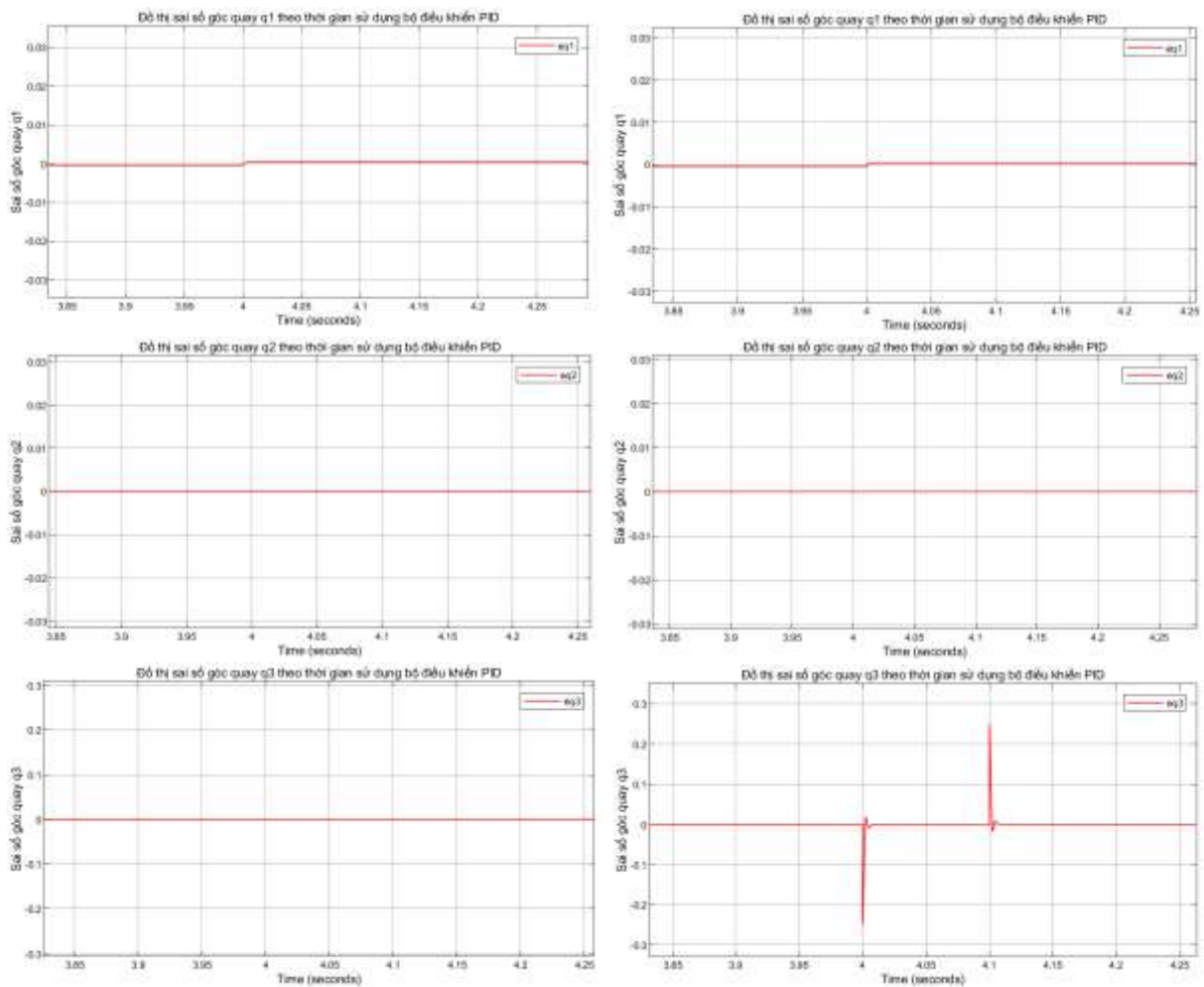
- Cũng giống như mô hình động lực học, thì khi có nhiễu tác động vào góc quay q_3 như hình 4.35, tại thời điểm nhiễu tác động thì đồ thị tọa độ x y z đều bị ảnh hưởng. Bộ điều khiển PID đã xử lý nhiễu khá tốt, bám nhanh về vị trí cũ và ổn định ngay,

tuy nhiên nếu nhiễu này cứ tiếp diễn qua các lần in hoặc vẽ tiếp theo thì bộ PID không thể khử nhiễu hoàn toàn được.



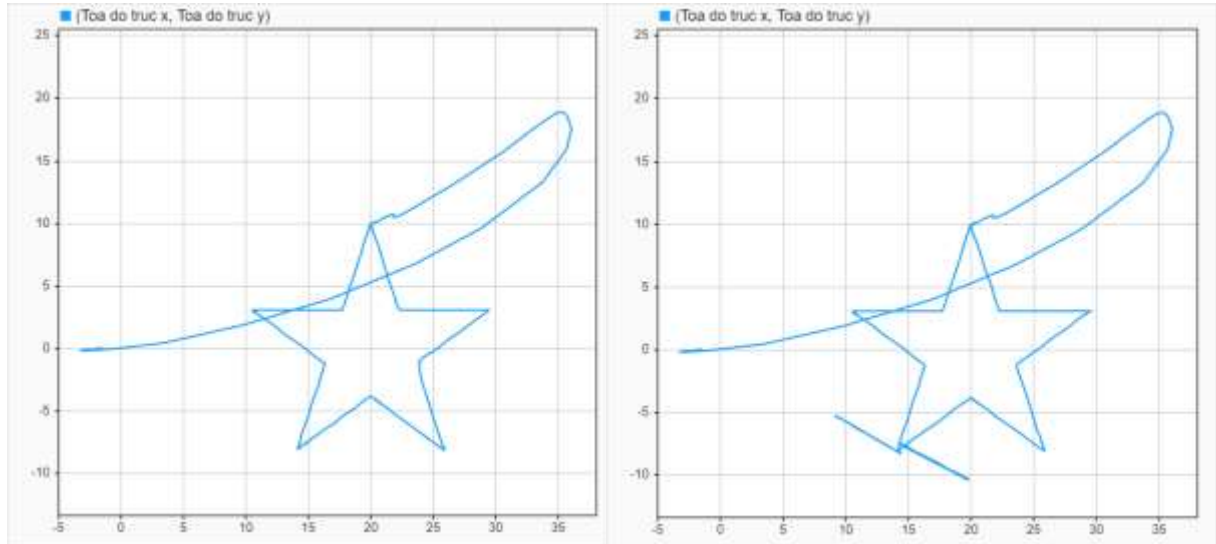
Hình 4.35: Đồ thị tọa độ x y đầu ra cho hình vẽ 2D của mô hình động lực học sử dụng bộ điều khiển PID khi có nhiễu

- Cũng như mô hình động lực học, thì sai lệch góc quay bị ảnh hưởng nhiều nhất là tại góc quay q_3 do nhiễu tác động trực tiếp vào góc quay q_3 , còn sai số 2 góc quay còn lại ít bị ảnh hưởng hơn do nhiễu chỉ tác động vào khâu chấp hành như hình 4.36.



Hình 4.36: Đồ thị sai số góc quay khi vẽ 2D của mô hình SolidWork sử dụng bộ điều khiển PID khi có nhiễu

- Khi không có nhiễu (hình 4.37 bên trái), hình 2D được vẽ ra rất mượt và ổn định, cho thấy bộ điều khiển đáp ứng tốt giá trị đặt. Còn khi có nhiễu (hình 4.37 bên phải) vào thì hình ảnh 2D bị lệch 2 đường khá dài, ảnh hưởng đến chất lượng hình vẽ, cho thấy bộ điều khiển xử lý nhiễu chưa tốt lắm.



Hình 4.37: Đồ thị kết quả vẽ 2D của mô hình SolidWork khi không có nhiễu và khi có nhiễu

4.2.6.3 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.
- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.01	0.018	0.0898%	92.05%
Trục y	0.01	0.0146	0.2825%	102.2%
Trục z	0.0141	0.0011	0.0227%	3662%

Bảng 4.3: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình SolidWork khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
--	--------------	----------	---------------------	-----------

Trục x	0.01	0.0112	0.0584%	80%
Trục y	0.012	0.0032	0.0869%	88.7%

Bảng 4.4: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID của mô hình SolidWork khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do xây dựng trên SolidWork thực hiện in 3D và vẽ 2D được thể hiện trong Bảng 4.3 và Bảng 4.4, có thể nhận thấy rằng thời gian xác lập t_s và phần trăm sai số xác lập trung bình $\varepsilon\%$ đều đáp ứng tốt các tiêu chí đã đề ra. Tuy nhiên, phần trăm độ quá điều chỉnh $POT(\%)$ của trục Z vẫn còn khá cao và cần được cải thiện.
- Mặt khác, các kết quả này được thu thập trong điều kiện mô phỏng không có nhiễu. Khi hệ thống chịu tác động nhiễu tại khâu 3, gây ra sai lệch góc tại khớp 3, bộ điều khiển PID vẫn cho thấy khả năng đáp ứng nhanh và duy trì tính ổn định cho hệ thống. Tuy vậy, bộ điều khiển PID không thể hoàn toàn loại bỏ ảnh hưởng của nhiễu, dẫn đến gia tăng sai số. Nguyên nhân là do PID không có khả năng ước lượng hay bù nhiễu. Do đó, để cải thiện khả năng khử nhiễu và nâng cao độ chính xác, cần xem xét kết hợp thêm bộ điều khiển ILC vào hệ thống.
- So sánh giữa bộ điều khiển PID khi sử dụng mô hình solidwork và sử dụng mô hình động lực học của đối tượng, ta nhận thấy, chất lượng điều khiển tương tự nhau. Như vậy, việc xây dựng mô hình động lực học và tính chọn thông số động cơ bằng phương pháp nhận dạng là phù hợp.

4.3. Kết quả mô phỏng hệ thống điều khiển cánh tay robot 3 DOF sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC

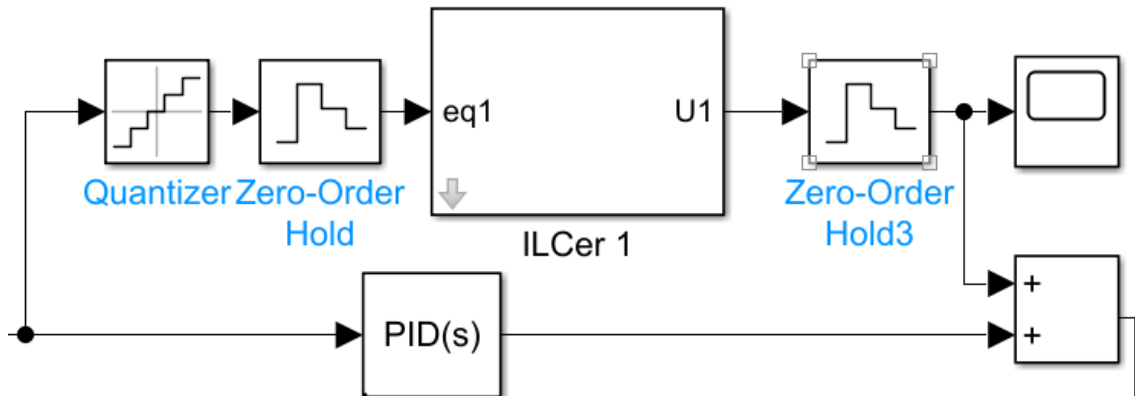
4.3.1. Sơ đồ mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC

Ở phần trước, khi sử dụng bộ điều khiển PID để điều khiển cánh tay Robot thì chúng ta đã mô phỏng trong miền liên tục. Đến phần này, khi tích hợp thêm bộ điều khiển ILC vào để cải thiện khả năng khử nhiễu thì bộ điều khiển ILC cần được rời rạc hóa hay nói cách khác là bộ điều khiển ILC phải mô phỏng trong miền rời rạc, nên tín hiệu khi đưa vào cho bộ điều khiển ILC xử lý phải được rời rạc hóa trước. Vì cần độ chính xác cao và hiệu quả khi áp dụng từ rời rạc qua liên tục thì ta nên chọn thời gian lấy mẫu khoảng 0.001 giây.

Chúng ta sẽ sử dụng 2 khối Quantizer kết hợp với khối ZOH để lấy mẫu và lượng tử hóa tín hiệu đầu vào để đưa tín hiệu đầu vào từ miền liên tục sang miền rời rạc rồi đưa vào bộ điều khiển ILC. Tiếp theo, sau khi bộ điều khiển ILC xử lý tín hiệu xong và

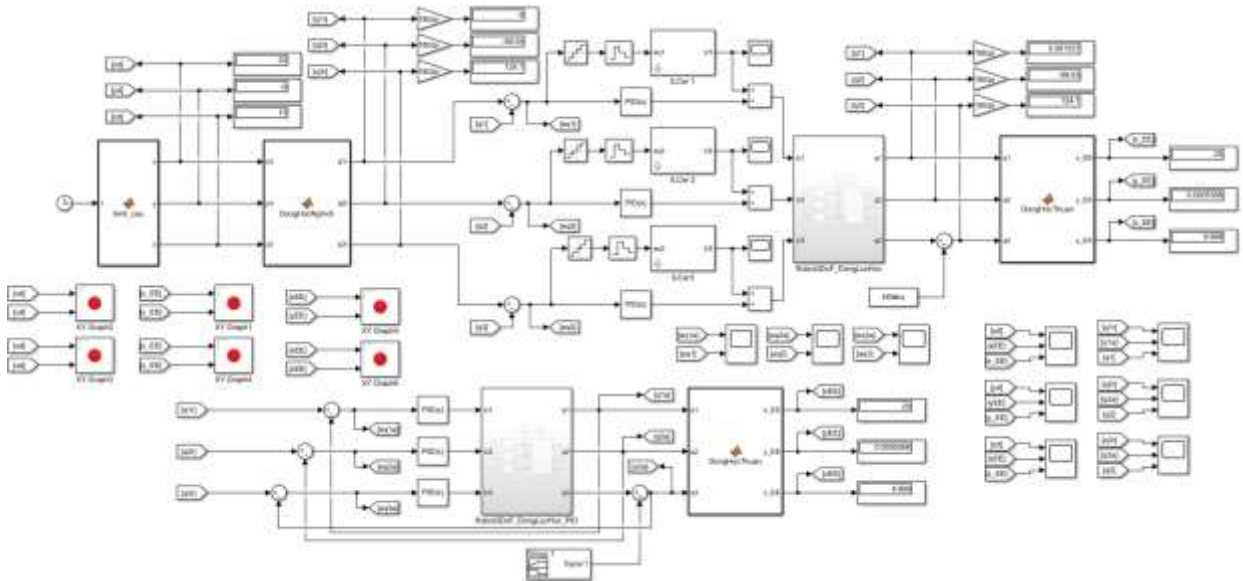
đưa ra tín hiệu điều khiển dạng rời rạc thì chúng ta sử dụng 1 khối ZOH để giữ tín hiệu rời rạc và biến nó thành tín hiệu khả dụng trong miền liên tục. Từ đó, tín hiệu điều khiển từ ILC sẽ kết hợp tín hiệu điều khiển PID ở trong miền liên tục để điều khiển cánh tay Robot 3 bậc tự do như hình 4.38.

Với giá trị lượng tử hóa là 6×10^{-5} và chu kì lấy mẫu $T_s = 0.001$.



Hình 4.38: Rời rạc hóa tín hiệu điều khiển và biến tín hiệu điều khiển đầu ra về miền liên tục cho bộ điều khiển ILC

- Áp dụng bộ điều khiển ILC vào sơ đồ mô phỏng như trong hình 4.39.



Hình 4.39: Sơ đồ mô phỏng điều khiển mô hình Động lực học sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC

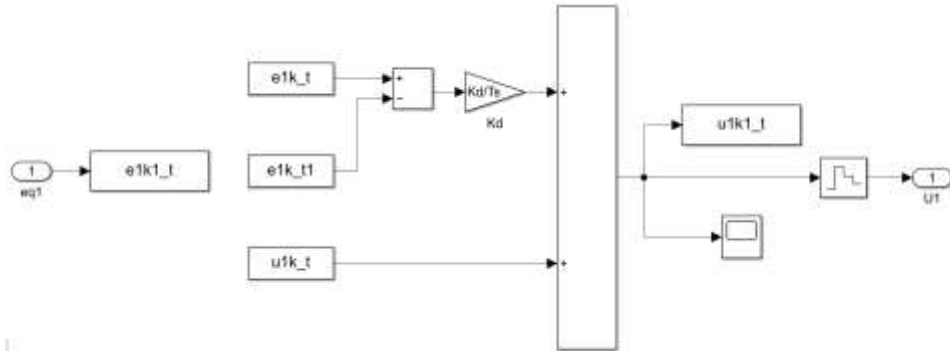
4.3.2. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D

Dựa vào công thức học lặp của bộ điều khiển ILC kiểu D, ta tiến hành mô phỏng bộ điều khiển ILC kiểu D trên Matlab Simulink như sau:

- Công thức học lặp của bộ điều khiển ILC kiểu D ở miền rời rạc:

$$u_{k+1}(t) = u_k(t) + K_d \frac{e_k(t) - e_k(t-1)}{T_s}$$

- Mô phỏng bộ điều khiển ILC kiểu D trên Matlab Simulink:

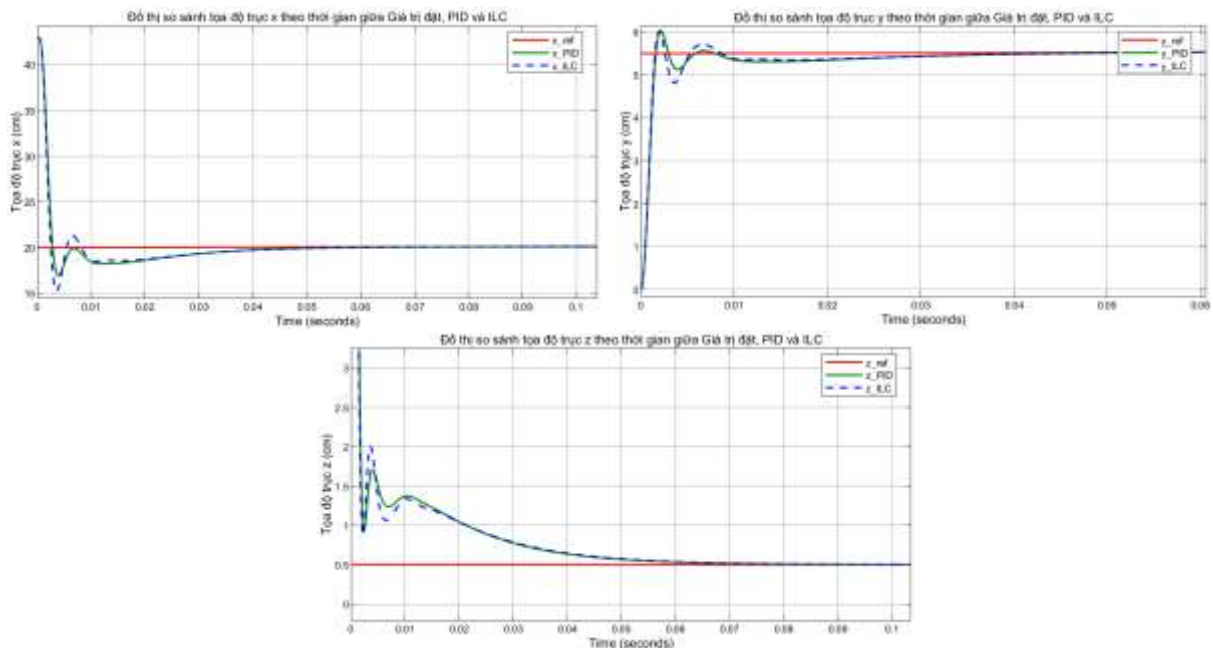


Hình 4.40: Sơ đồ mô phỏng bộ điều khiển ILC kiểu D

Sau nhiều lần tinh chỉnh hệ số K_d thì ta lựa chọn được thông số của bộ điều khiển ILC kiểu D với hệ số K_d và số lần học n là $K_d = 0.03$ và $n = 10$.

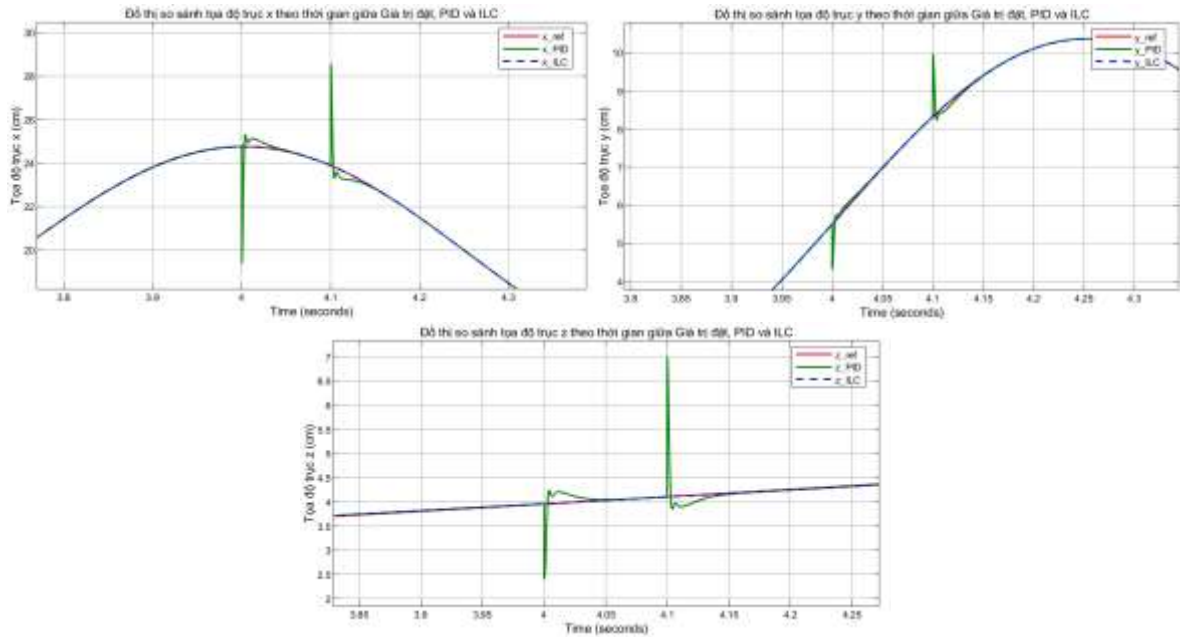
4.3.2.2 Kết quả mô phỏng in 3D khi có nhiễu tác động

- Kết quả đồ thị so sánh các trục x y z trong hình 4.41 cho thấy độ quá điều chỉnh ban đầu tăng do thành phần đạo hàm D làm tăng độ nhạy với độ quá điều chỉnh ban đầu.



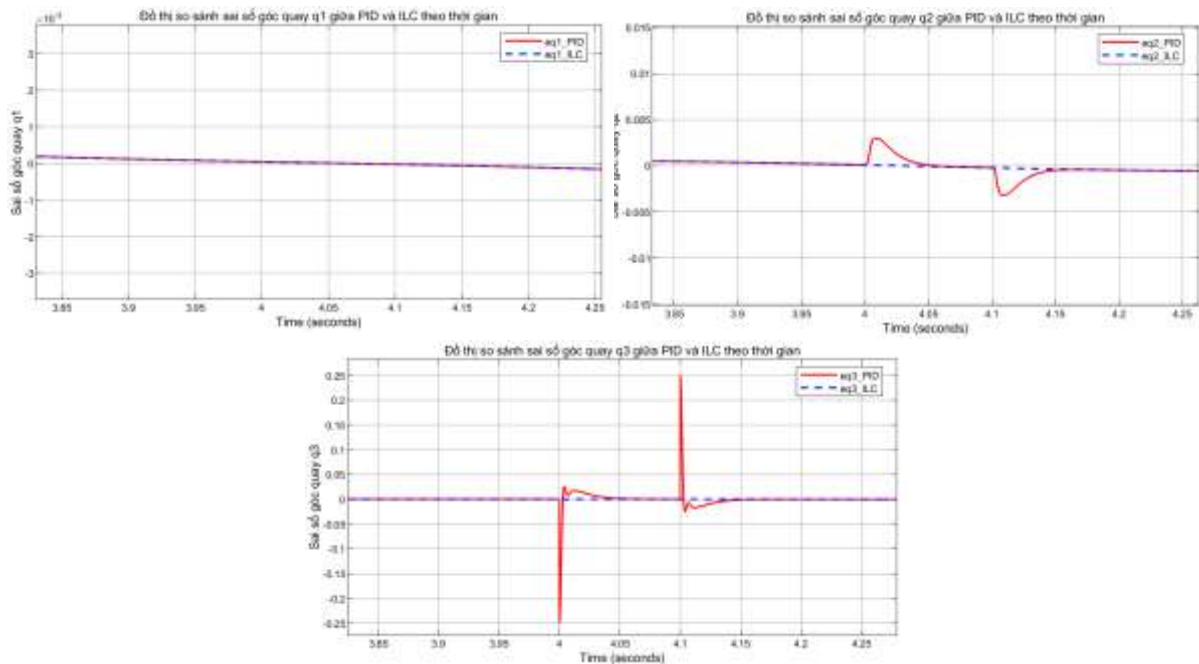
Hình 4.41: Đồ thị so sánh tọa độ x y z khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học

- Kết quả xử lý nhiễu của bộ ILC kiểu D cho thấy nhiễu được xử lý hoàn toàn sau 2 lần học như trong hình 4.42.



Hình 4.42: Đồ thị so sánh khả năng xử lý nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu D

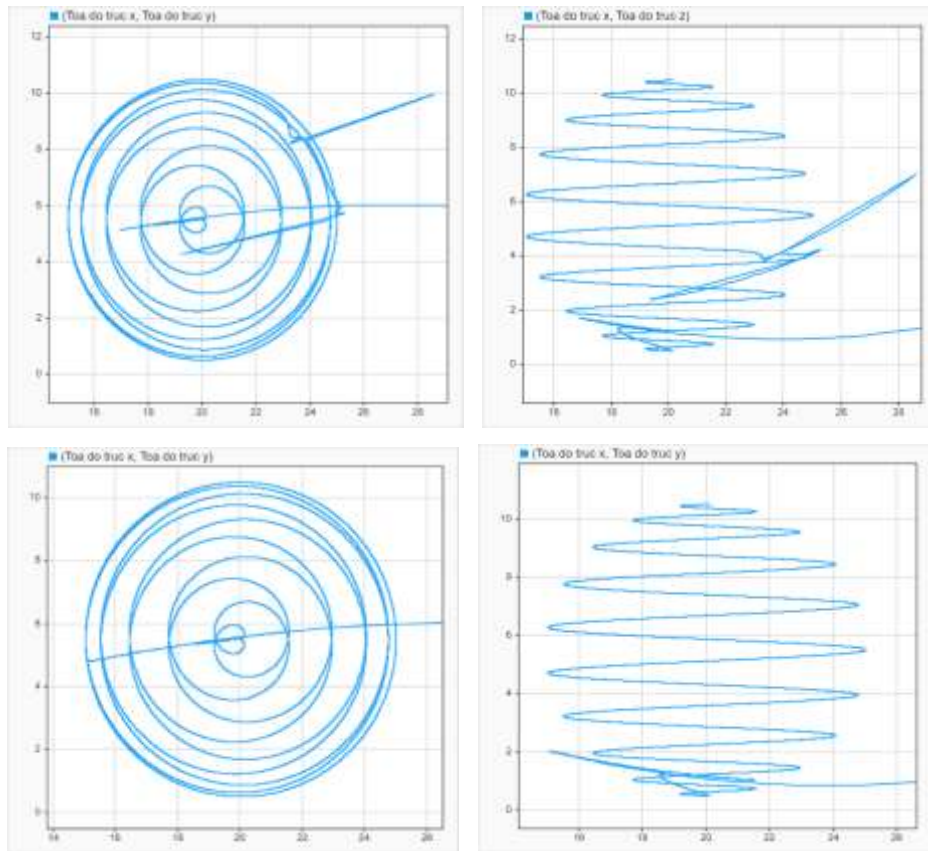
- Kết quả so sánh đồ thị sai lệch góc quay khi xử lý nhiễu giữa hai phương pháp điều khiển được thể hiện trong hình 4.43, cho thấy bộ điều khiển có ILC xử lý nhiễu khá tốt.



Hình 4.43: Đồ thị so sánh sai số góc quay khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học

- Kết quả so sánh hình in 3D trên hai mặt phẳng Oxy và Oxz giữa hai phương pháp PID (hình 4.44 ở trên) và PID kết hợp ILC kiểu D (hình 4.44 ở dưới) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu

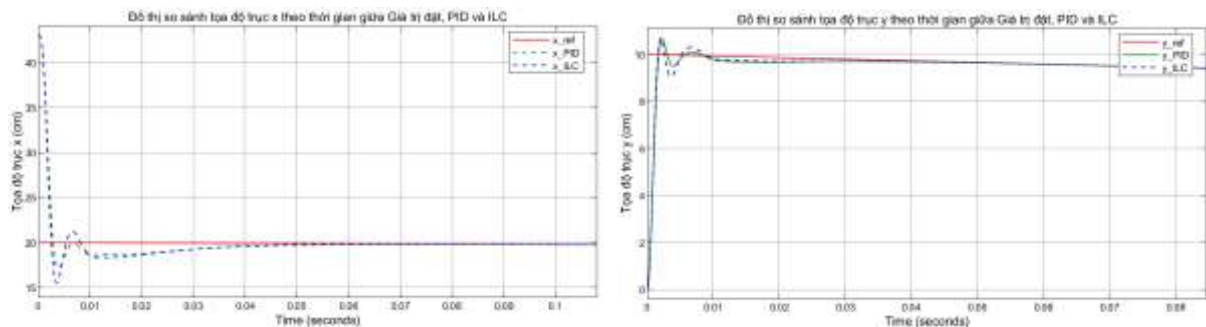
D đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình in.



Hình 4.44: Đồ thị so sánh theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu

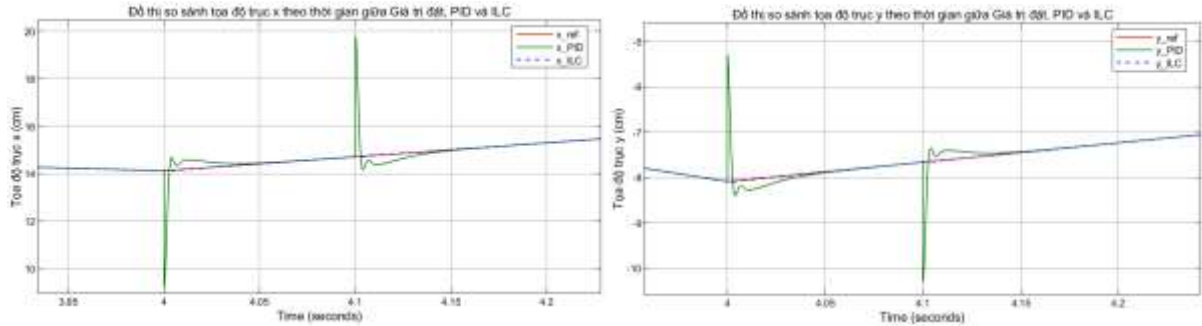
4.3.2.3 Kết quả mô phỏng vẽ 2D khi có nhiễu tác động

- Kết quả đồ thị so sánh các trục x y trong hình 4.45 cho thấy độ quá điều chỉnh ban đầu tăng do thành phần đạo hàm D làm tăng độ nhạy với độ quá điều chỉnh ban đầu.



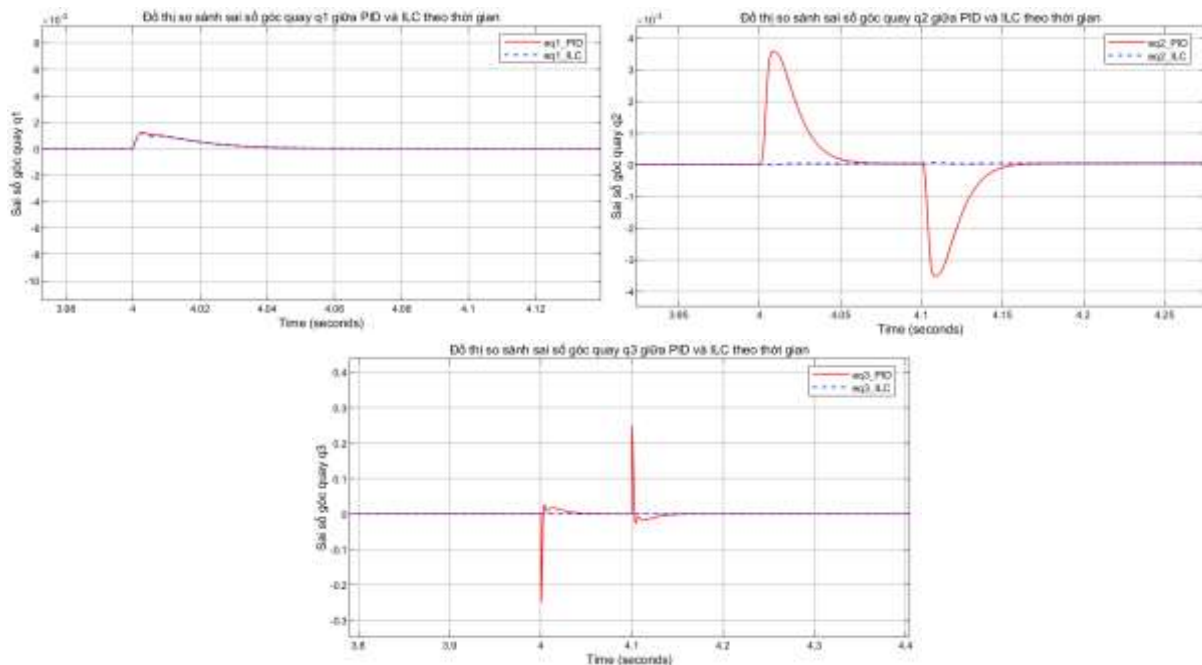
Hình 4.45: Đồ thị so sánh tọa độ x y z khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học

- Kết quả xử lý nhiễu của bộ ILC kiểu D cho thấy nhiễu được xử lý hoàn toàn sau 2 lần học như trong hình 4.46.



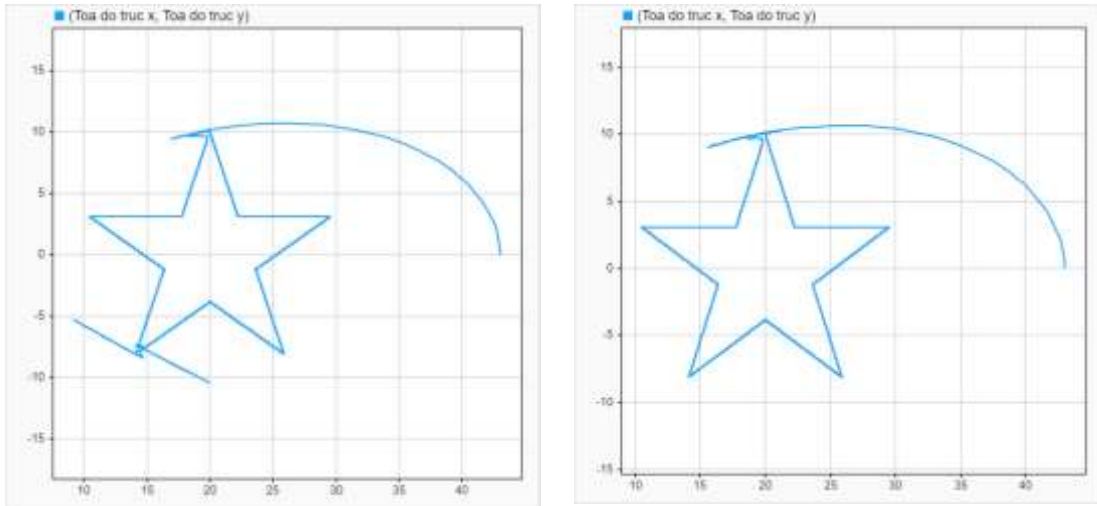
Hình 4.46: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu D

- Kết quả so sánh đồ thị sai lệch góc quay khi xử lý nhiễu giữa hai phương pháp điều khiển được thể hiện trong hình 4.43, cho thấy bộ điều khiển có ILC xử lý nhiễu khá tốt.



Hình 4.47: Đồ thị so sánh sai lệch góc quay khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học

- Kết quả so sánh hình vẽ 2D trên hai mặt phẳng Oxy giữa hai phương pháp PID (hình 4.48 bên trái) và PID kết hợp ILC kiểu D (hình 4.48 bên phải) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình vẽ.



Hình 4.48: Đồ thị so sánh kết quả vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu của mô hình động lực học

4.3.2.4 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.
- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0378	0.0059	0.0293%	23.7%
Trục y	0.0245	0.0014	0.0258%	9.4%
Trục z	0.0796	0.0062	0.1113%	300%

Bảng 4.5: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình động lực học khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0375	0.0029	0.0144%	22.36%
Trục y	0.016	0.0018	0.334%	9.4%

Bảng 4.6: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình động lực học khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình động lực học cánh tay Robot 3 bậc tự do thực hiện in 3D và vẽ 2D bằng phương pháp sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D được thể hiện trong Bảng 4.5 và Bảng 4.6, có thể nhận thấy thời gian xác lập t_s và phần trăm sai số xác lập trung bình $\varepsilon\%$ đã giảm so với khi chỉ sử dụng bộ điều khiển PID, chỉ có độ quá điều chỉnh POT (%) của trục x và z là có xu hướng tăng nhẹ.
- Đối với nhiều tác động, kết quả cho thấy khi sử dụng kết hợp bộ điều khiển PID và ILC, nhiễu đã được loại bỏ gần như hoàn toàn chỉ sau 2 lần học. Điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả, vượt trội so với việc chỉ dùng bộ điều khiển PID đơn thuần.

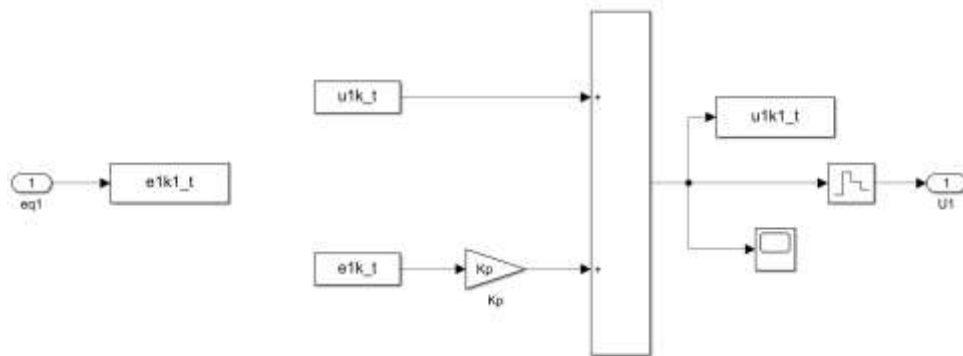
4.3.3. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P

Dựa vào công thức học lặp của bộ điều khiển ILC kiểu P, ta tiến hành mô phỏng bộ điều khiển ILC kiểu P trên Matlab Simulink như sau:

- Công thức học lặp của bộ điều khiển ILC kiểu P ở miền rời rạc:

$$u_{k+1}(t) = u_k(t) + K_p e_k(t)$$

- Mô phỏng bộ điều khiển ILC kiểu P trên Matlab Simulink:

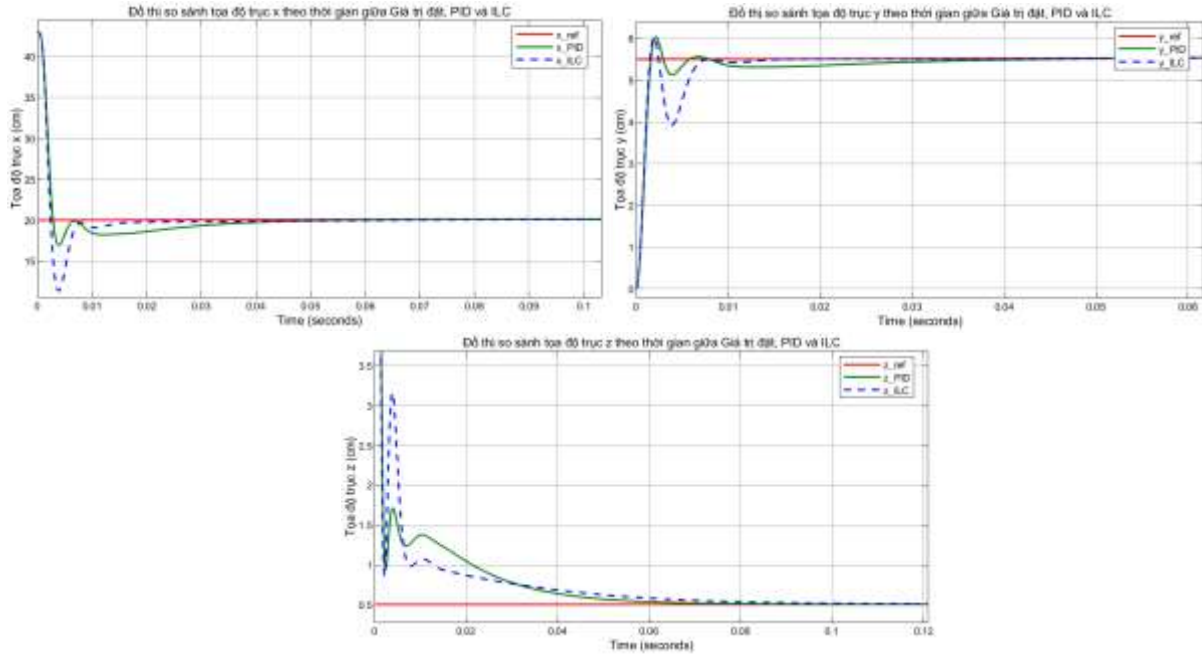


Hình 4.49: Sơ đồ mô phỏng bộ điều khiển ILC kiểu P

Sau nhiều lần tinh chỉnh hệ số K_p thì ta lựa chọn được thông số của bộ điều khiển ILC kiểu P với hệ số K_p và số lần học n là $K_p = 30$ và $n = 10$.

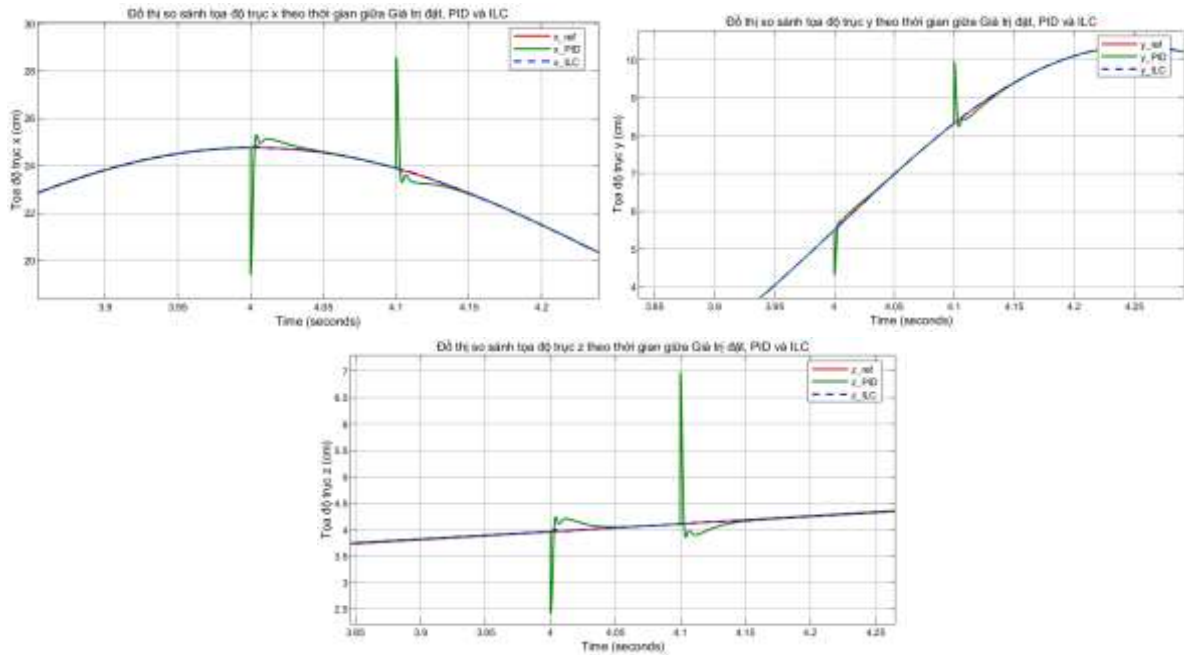
4.3.3.2 Kết quả mô phỏng in 3D khi có nhiễu tác động

- Kết quả đồ thị so sánh các trục x y z trong hình 4.50 cho thấy thành phần P giúp giảm sai lệch nhanh chóng nhưng lại làm tăng độ nhạy với độ quá điều chỉnh ban đầu.



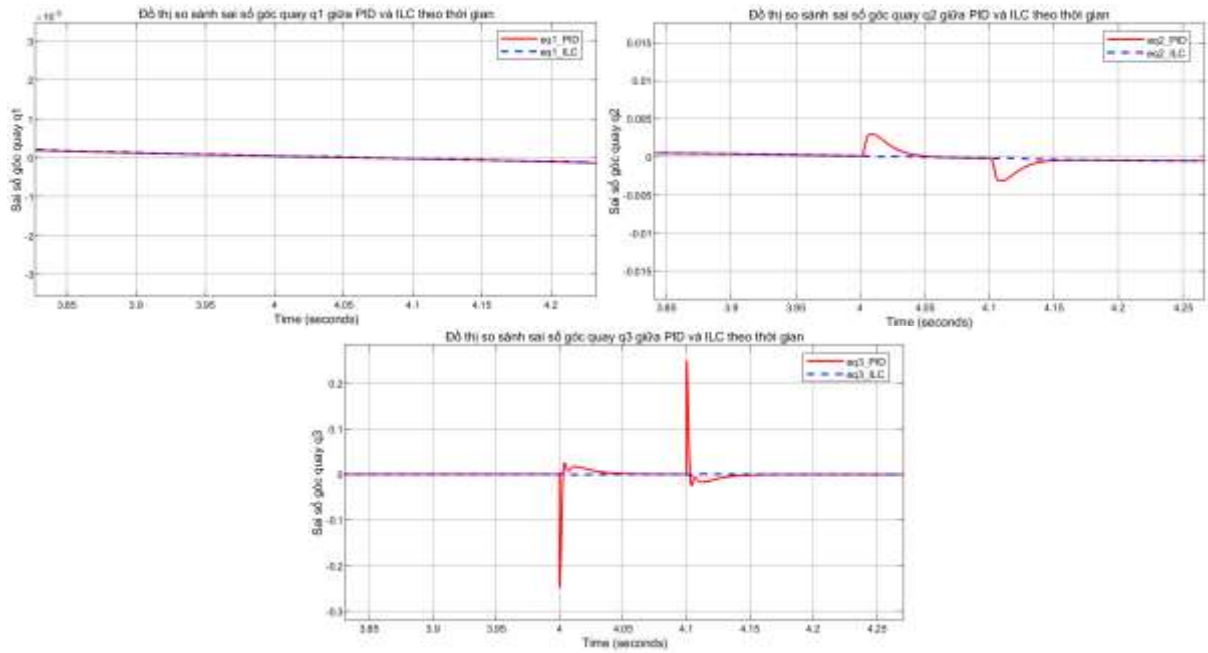
Hình 4.50: Đồ thị so sánh tọa độ x y z khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học

- Kết quả xử lý nhiễu của bộ ILC kiểu P cho thấy nhiễu được xử lý hoàn toàn sau 2 lần học như trong hình 4.51.



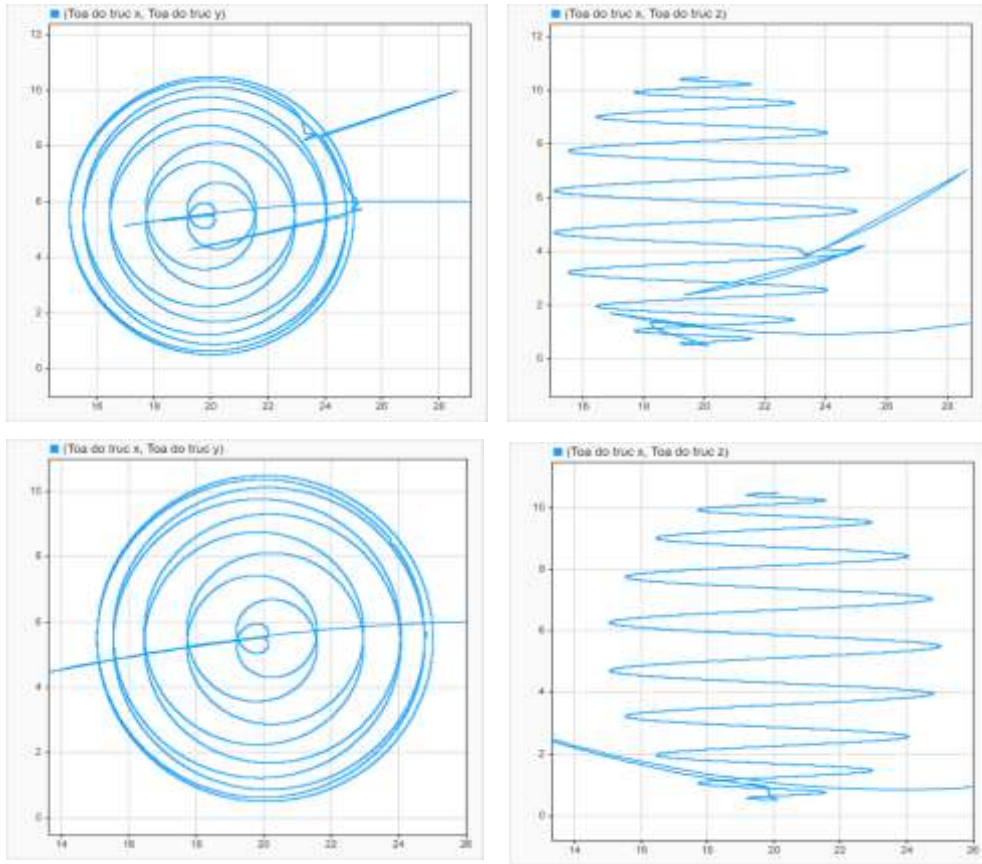
Hình 4.51: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu P

- Kết quả so sánh đồ thị sai lệch góc quay khi xử lý nhiễu giữa hai phương pháp điều khiển được thể hiện trong hình 4.52, cho thấy bộ điều khiển có ILC xử lý nhiễu khá tốt.



Hình 4.52: Đồ thị so sánh sai số góc quay khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học

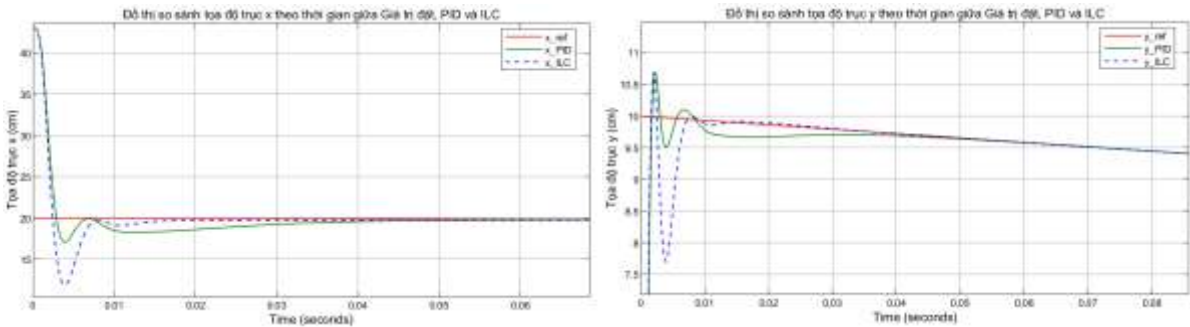
- Kết quả so sánh hình in 3D trên hai mặt phẳng Oxy và Oxz giữa hai phương pháp PID (hình 4.53 ở trên) và PID kết hợp ILC kiểu P (hình 4.53 ở dưới) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình in.



Hình 4.53: Đồ thị so sánh theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu

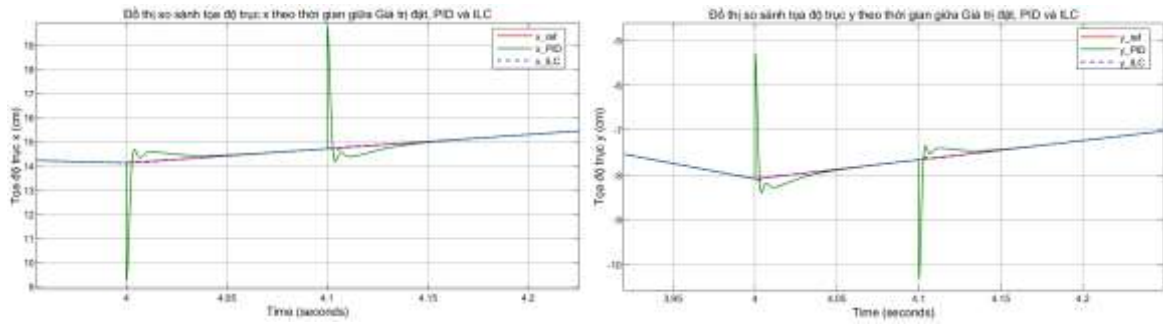
4.3.3.3 Kết quả mô phỏng vẽ 2D khi có nhiễu tác động

- Kết quả đồ thị so sánh các trục x y trong hình 4.54 cho thấy thành phần P giúp giảm sai lệch nhanh chóng nhưng lại làm tăng độ nhạy với độ quá điều chỉnh ban đầu.



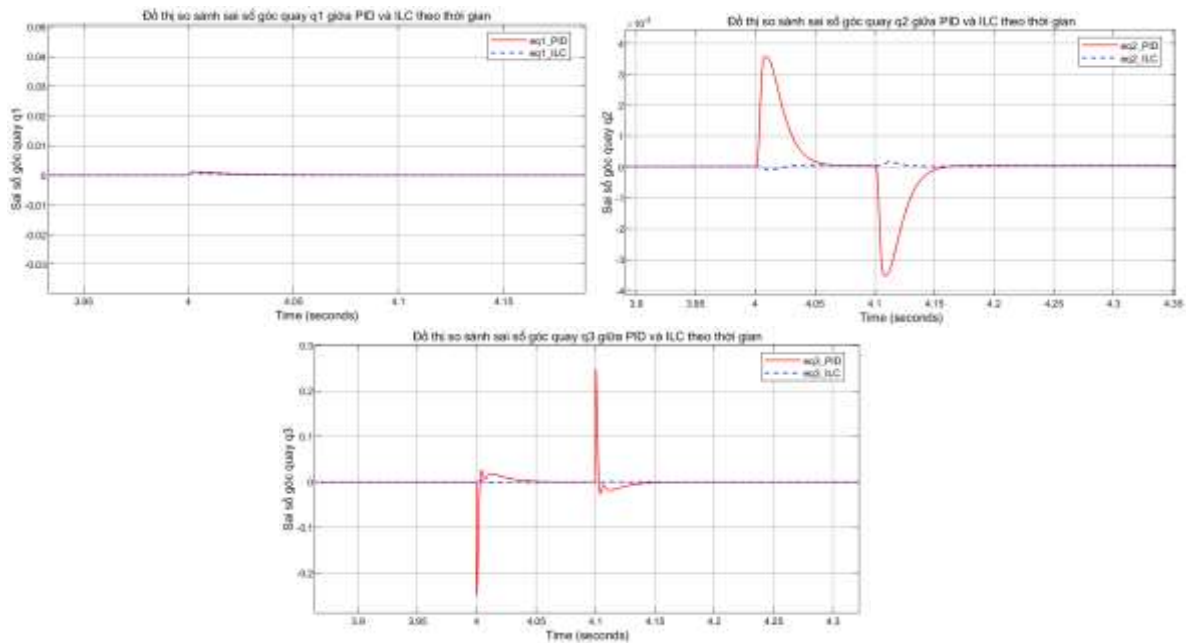
Hình 4.54: Đồ thị so sánh tọa độ x y z khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học

- Kết quả xử lý nhiễu của bộ ILC kiểu P cho thấy nhiễu được xử lý hoàn toàn sau 2 lần học như trong hình 4.55.



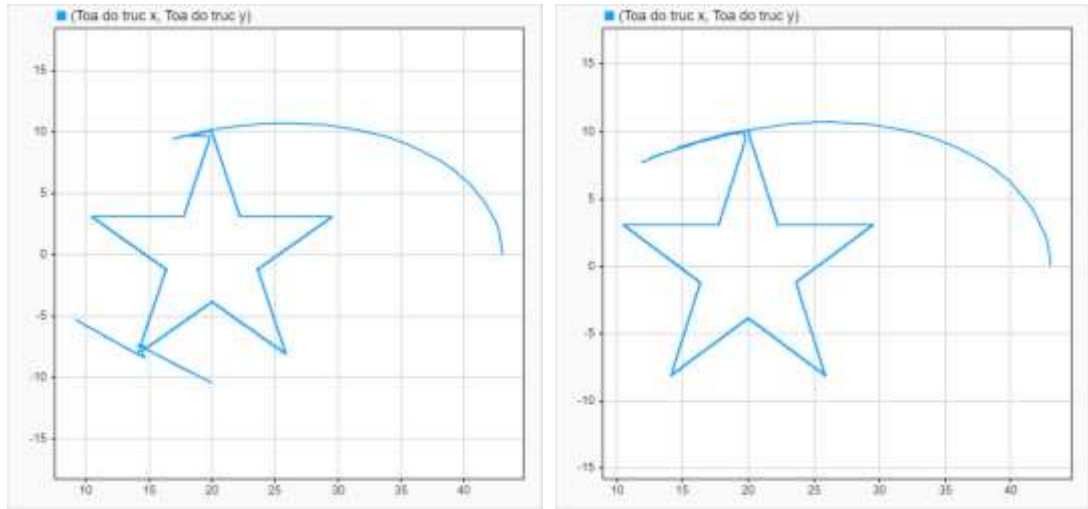
Hình 4.55: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu P

- Kết quả so sánh đồ thị sai lệch góc quay khi xử lý nhiễu giữa hai phương pháp điều khiển được thể hiện trong hình 4.56, cho thấy bộ điều khiển có ILC xử lý nhiễu khá tốt.



Hình 4.56: Đồ thị so sánh sai số góc quay khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học

- Kết quả so sánh hình vẽ 2D trên hai mặt phẳng Oxy giữa hai phương pháp PID (hình 4.57 bên trái) và PID kết hợp ILC kiểu P (hình 4.57 bên phải) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình vẽ.



Hình 4.57: Đồ thị so sánh kết quả vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu của mô hình động lực học

4.3.3.4 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.
- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0137	0.0065	0.0326%	43.35%
Trục y	0.0069	0.0014	0.0249%	28.6%
Trục z	0.1062	0.0062	0.1109%	532.4%

Bảng 4.7: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình động lực học khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0147	0.0037	0.0184%	40%
Trục y	0.01	0.0017	0.312%	22.78%

Bảng 4.8: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình động lực học khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình động lực học cánh tay Robot 3 bậc tự do thực hiện in 3D và vẽ 2D bằng phương pháp sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P được thể hiện trong Bảng 4.7 và Bảng 4.8, có thể nhận thấy thời gian xác lập t_s và phần trăm sai số xác lập trung bình $\varepsilon\%$ đã giảm nhiều so với khi chỉ sử dụng bộ điều khiển PID, độ quá điều chỉnh POT (%) của trục x y z có xu hướng tăng.
- Đối với nhiễu tác động, kết quả cho thấy khi sử dụng kết hợp bộ điều khiển PID và ILC, nhiễu đã được loại bỏ gần như hoàn toàn chỉ sau 2 lần học. Điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả, vượt trội so với việc chỉ dùng bộ điều khiển PID đơn thuần.

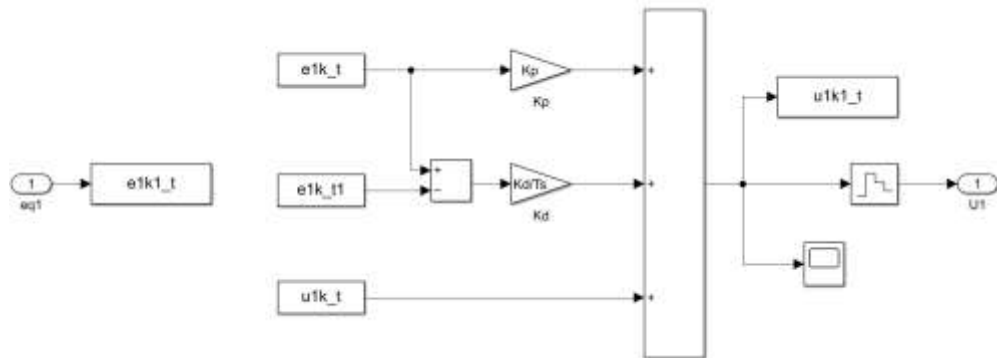
4.3.4. Kết quả mô phỏng điều khiển mô hình Động lực học khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD

Dựa vào công thức học lặp của bộ điều khiển ILC kiểu PD, ta tiến hành mô phỏng bộ điều khiển ILC kiểu PD trên Matlab Simulink như sau:

- Công thức học lặp của bộ điều khiển ILC kiểu PD ở miền rời rạc:

$$u_{k+1}(t) = u_k(t) + K_p e_k(t) + K_d \frac{e_k(t) - e_k(t-1)}{T_s}$$

- Mô phỏng bộ điều khiển ILC kiểu PD trên Matlab Simulink:

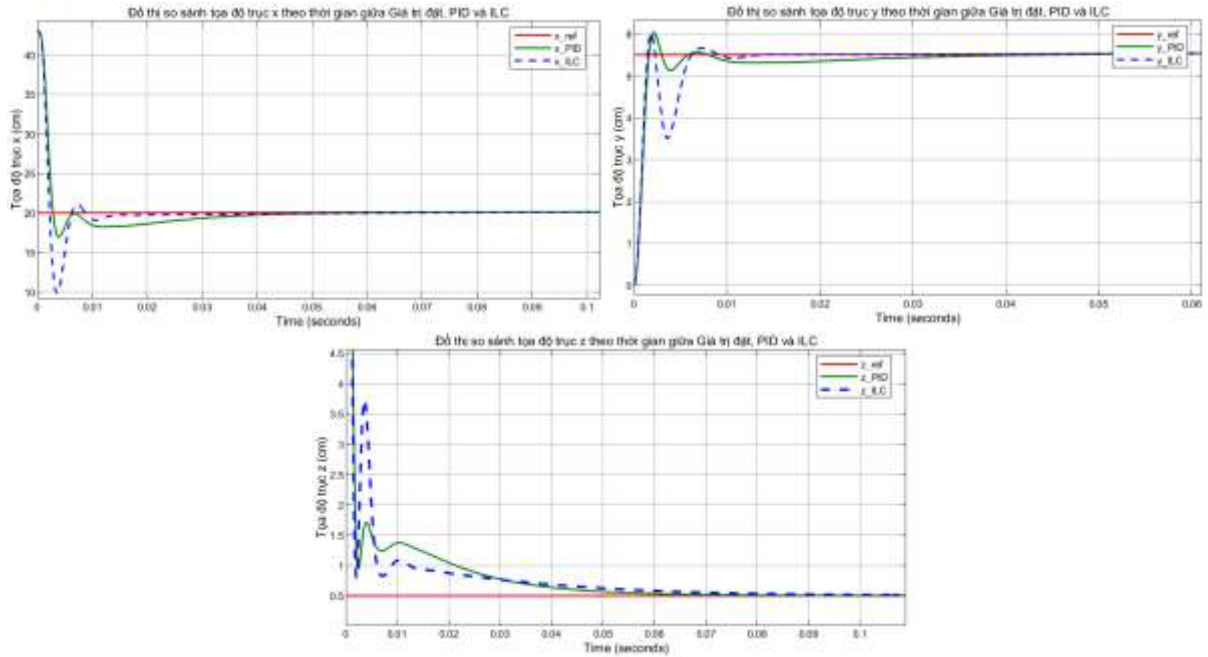


Hình 4.58: Sơ đồ mô phỏng bộ điều khiển PD cho mô hình SolidWork

Sau nhiều lần tinh chỉnh hệ số K_p và K_d thì ta lựa chọn được thông số của bộ điều khiển ILC kiểu PD với hệ số K_p, K_d và số lần học n là $K_p = 20; K_d = 0.02$ và $n = 15$.

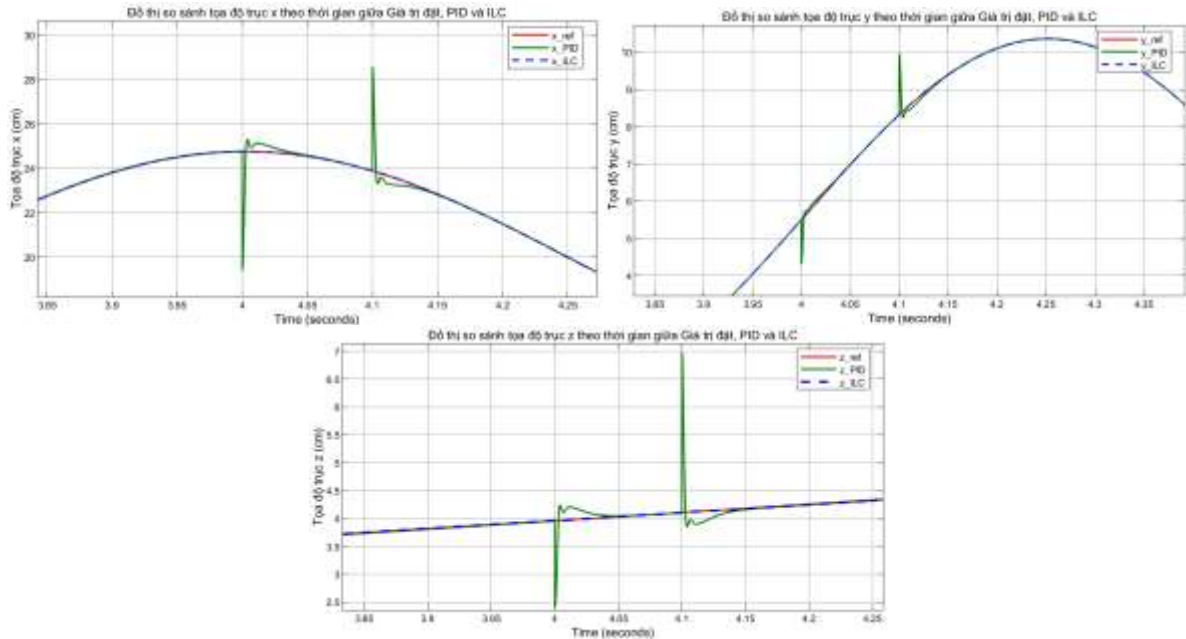
4.3.4.2 Kết quả mô phỏng in 3D khi có nhiễu tác động

- Kết quả đồ thị so sánh các trục x y z trong hình 4.59 cho thấy độ quá điều chỉnh vẫn tăng so với PID do thành phần P giúp giảm sai lệch nhanh chóng nhưng lại làm tăng độ nhạy với độ quá điều chỉnh ban đầu.



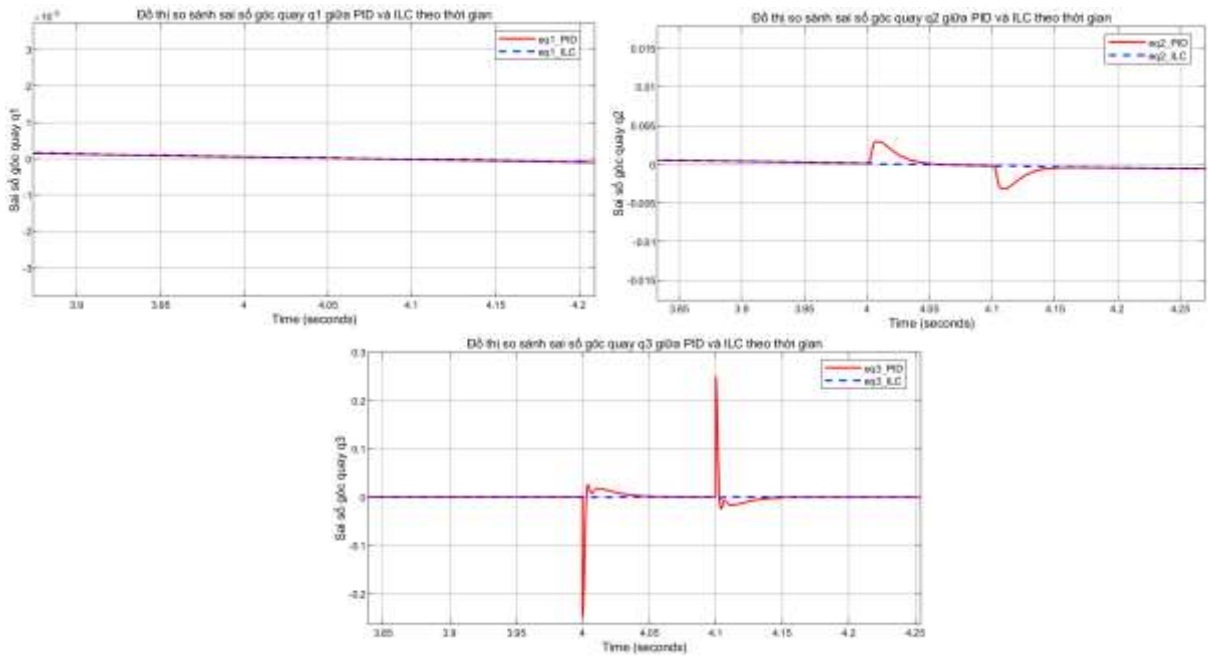
Hình 4.59: Đồ thị so sánh tọa độ x y z khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học

- Kết quả xử lý nhiễu của bộ ILC kiểu PD cho thấy nhiễu được xử lý hoàn toàn sau 2 lần học như trong hình 4.60.



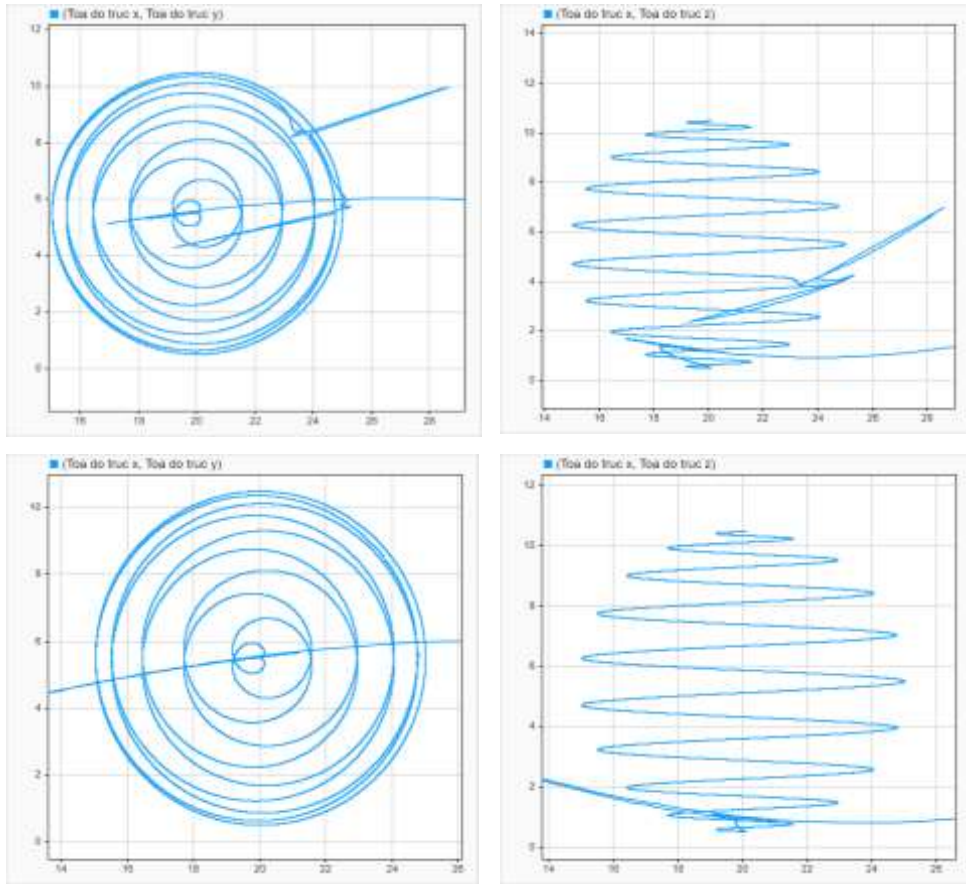
Hình 4.60: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu PD

- Kết quả so sánh đồ thị sai lệch góc quay khi xử lý nhiễu giữa hai phương pháp điều khiển được thể hiện trong hình 4.61, cho thấy bộ điều khiển có ILC xử lý nhiễu khá tốt.



Hình 4.61: Đồ thị so sánh sai lệch góc quay khi in 3D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học

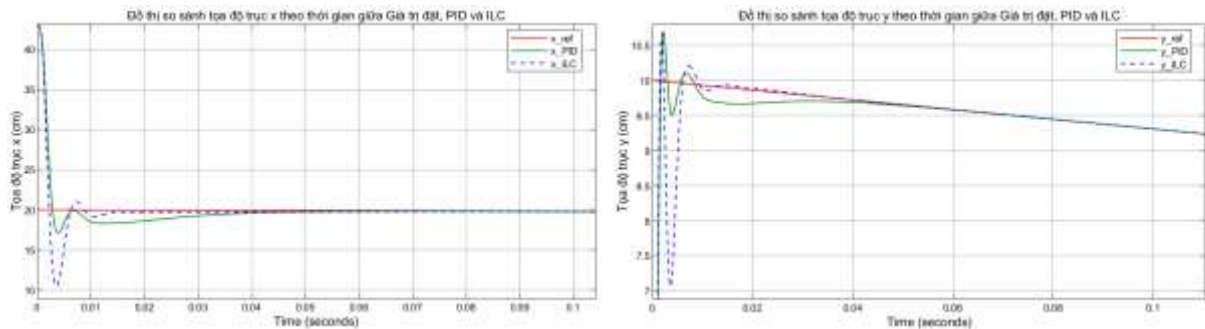
- Kết quả so sánh hình in 3D trên hai mặt phẳng Oxy và Oxz giữa hai phương pháp PID (hình 4.62 ở trên) và PID kết hợp ILC kiểu PD (hình 4.62 ở dưới) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình in.



Hình 4.62: Đồ thị so sánh theo 2 mặt phẳng Oxy và Oxz của mô hình động lực học giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu

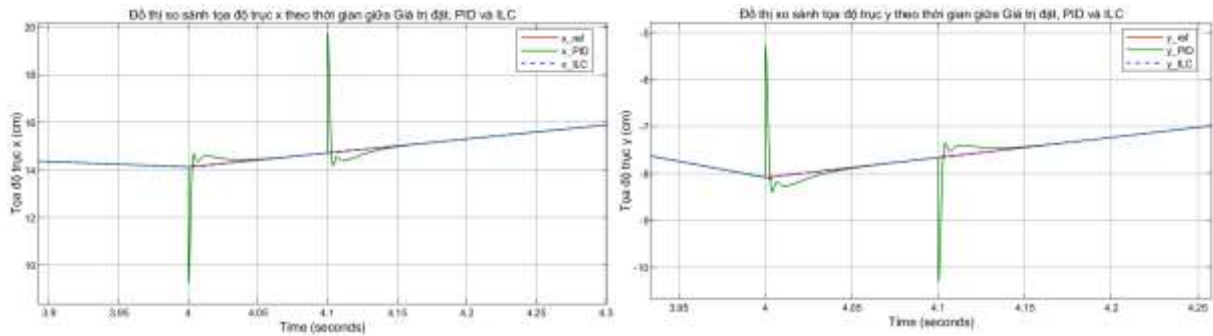
4.3.4.3 Kết quả mô phỏng vẽ 2D khi có nhiễu tác động

- Kết quả đồ thị so sánh các trục x y trong hình 4.63 cho thấy độ quá điều chỉnh vẫn tăng so với PID do thành phần P giúp giảm sai lệch nhanh chóng nhưng lại làm tăng độ nhạy với độ quá điều chỉnh ban đầu.



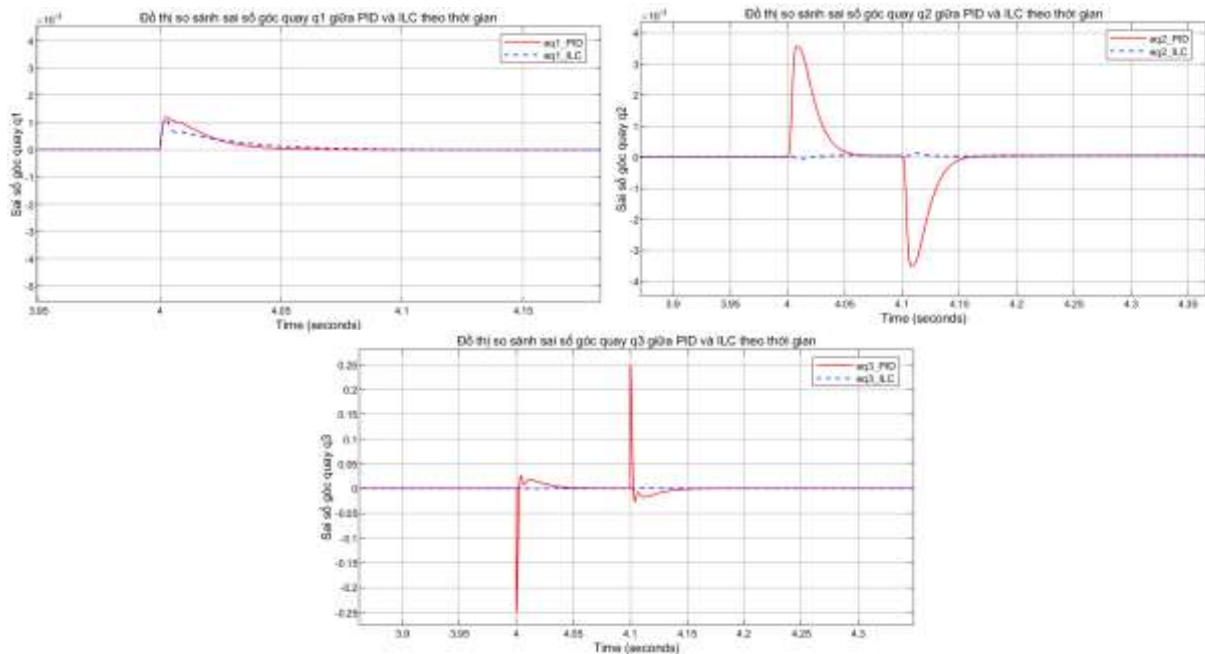
Hình 4.63: Đồ thị so sánh tọa độ x y z khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học

- Kết quả xử lý nhiễu của bộ ILC kiểu PD cho thấy nhiễu được xử lý hoàn toàn sau 2 lần học như trong hình 4.64.



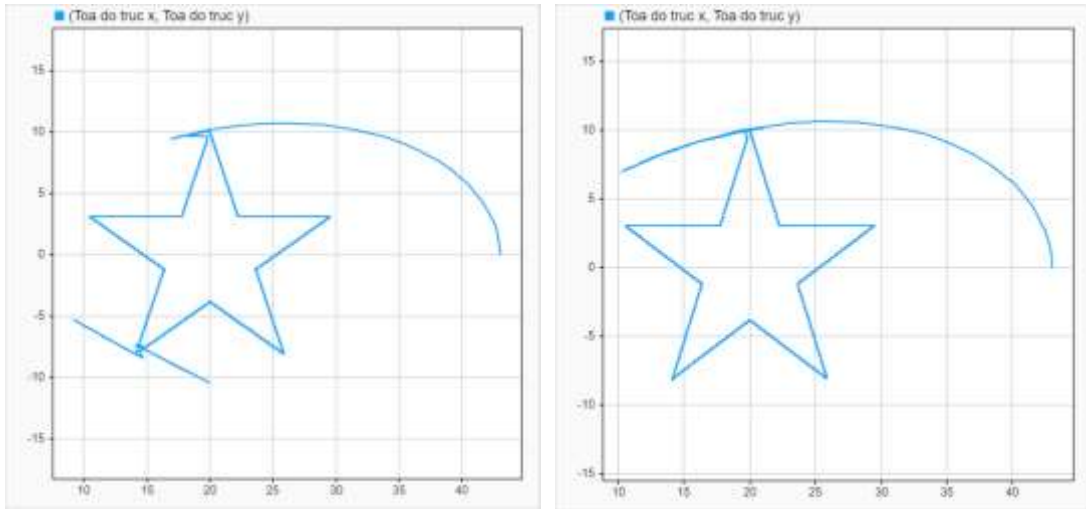
Hình 4.64: Đồ thị so sánh khả năng khử nhiễu của 2 bộ điều khiển PID với PID kết hợp ILC kiểu PD

- Kết quả so sánh đồ thị sai lệch góc quay khi xử lý nhiễu giữa hai phương pháp điều khiển được thể hiện trong hình 4.61, cho thấy bộ điều khiển có ILC xử lý nhiễu khá tốt.



Hình 4.65: Đồ thị so sánh sai số góc quay khi vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học

- Kết quả so sánh hình vẽ 2D trên hai mặt phẳng Oxy giữa hai phương pháp PID (hình 4.66 bên trái) và PID kết hợp ILC kiểu PD (hình 4.66 bên phải) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình vẽ.



Hình 4.66: Đồ thị so sánh kết quả vẽ 2D giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu của mô hình động lực học

4.3.4.4 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.
- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0129	0.0066	0.0329%	51.1%
Trục y	0.0083	0.0014	0.0246%	36.2%
Trục z	0.1075	0.0061	0.1107%	645.5%

Bảng 4.9: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình động lực học khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0133	0.0037	0.0184%	47.5%
Trục y	0.01	0.0015	0.309%	29.1%

Bảng 4.10: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình động lực học khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình động lực học cánh tay Robot 3 bậc tự do thực hiện in 3D và vẽ 2D bằng phương pháp sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD được thể hiện trong Bảng 4.9 và Bảng 4.10, có thể nhận thấy thời gian xác lập t_s và phần trăm sai số xác lập trung bình $\varepsilon\%$ đã giảm so với khi chỉ sử dụng bộ điều khiển PID, chỉ có độ quá điều chỉnh POT (%) của trục Z là vẫn còn xu hướng tăng, cần tinh chỉnh lại hệ số P hợp lý hơn.
- Đối với nhiều tác động, kết quả cho thấy khi sử dụng kết hợp bộ điều khiển PID và ILC, nhiễu đã được loại bỏ gần như hoàn toàn chỉ sau 2 lần học. Điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả, vượt trội so với việc chỉ dùng bộ điều khiển PID đơn thuần.

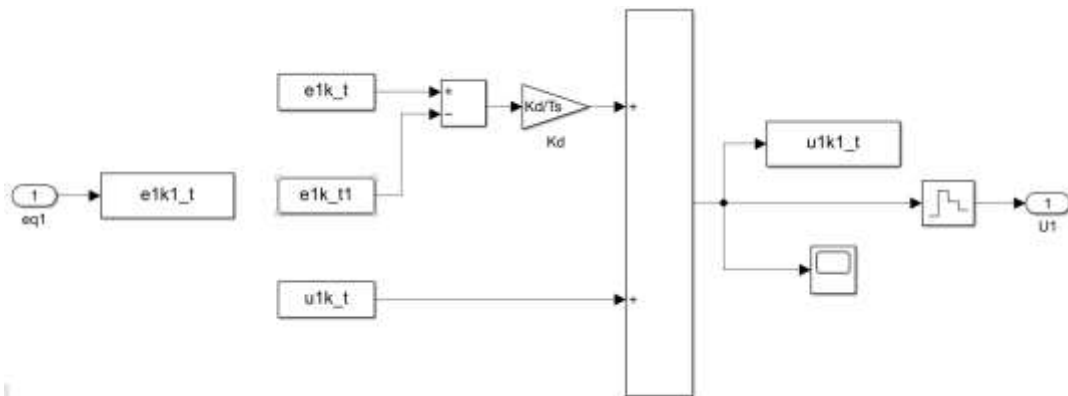
4.3.5. Kết quả mô phỏng điều khiển mô hình SolidWork khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D

Dựa vào công thức học lặp của bộ điều khiển ILC kiểu D, ta tiến hành mô phỏng bộ điều khiển ILC kiểu D trên Matlab Simulink như sau:

- Công thức học lặp của bộ điều khiển ILC kiểu D ở miền rời rạc:

$$u_{k+1}(t) = u_k(t) + K_d \frac{e_k(t) - e_k(t-1)}{T_s}$$

- Mô phỏng bộ điều khiển ILC kiểu D trên Matlab Simulink:

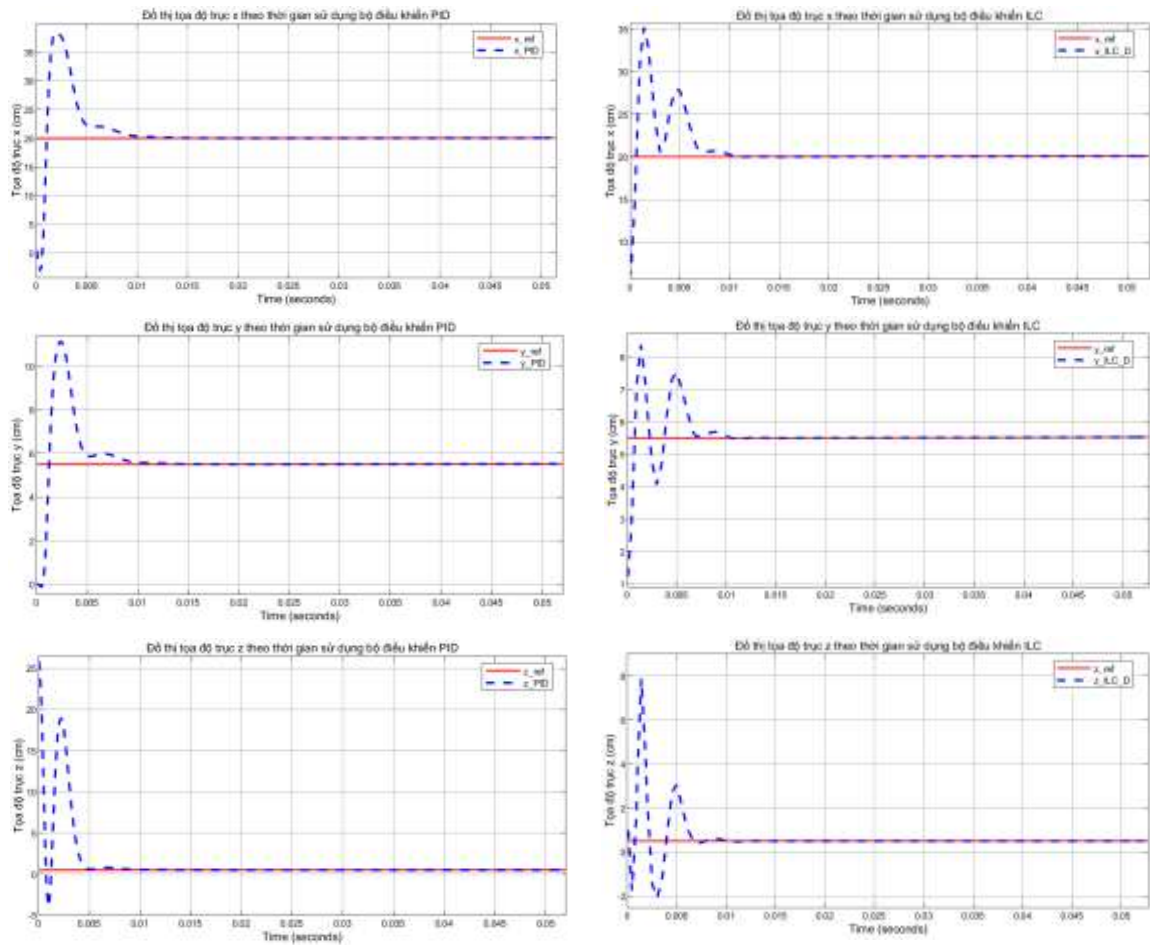


Hình 4.67: Sơ đồ mô phỏng bộ điều khiển ILC kiểu D cho mô hình SolidWork

Sau nhiều lần tinh chỉnh hệ số K_d thì ta lựa chọn được thông số của bộ điều khiển ILC kiểu D với hệ số K_d và số lần học n là $K_d = 1.2 \times 10^{-3}$ và $n = 10$.

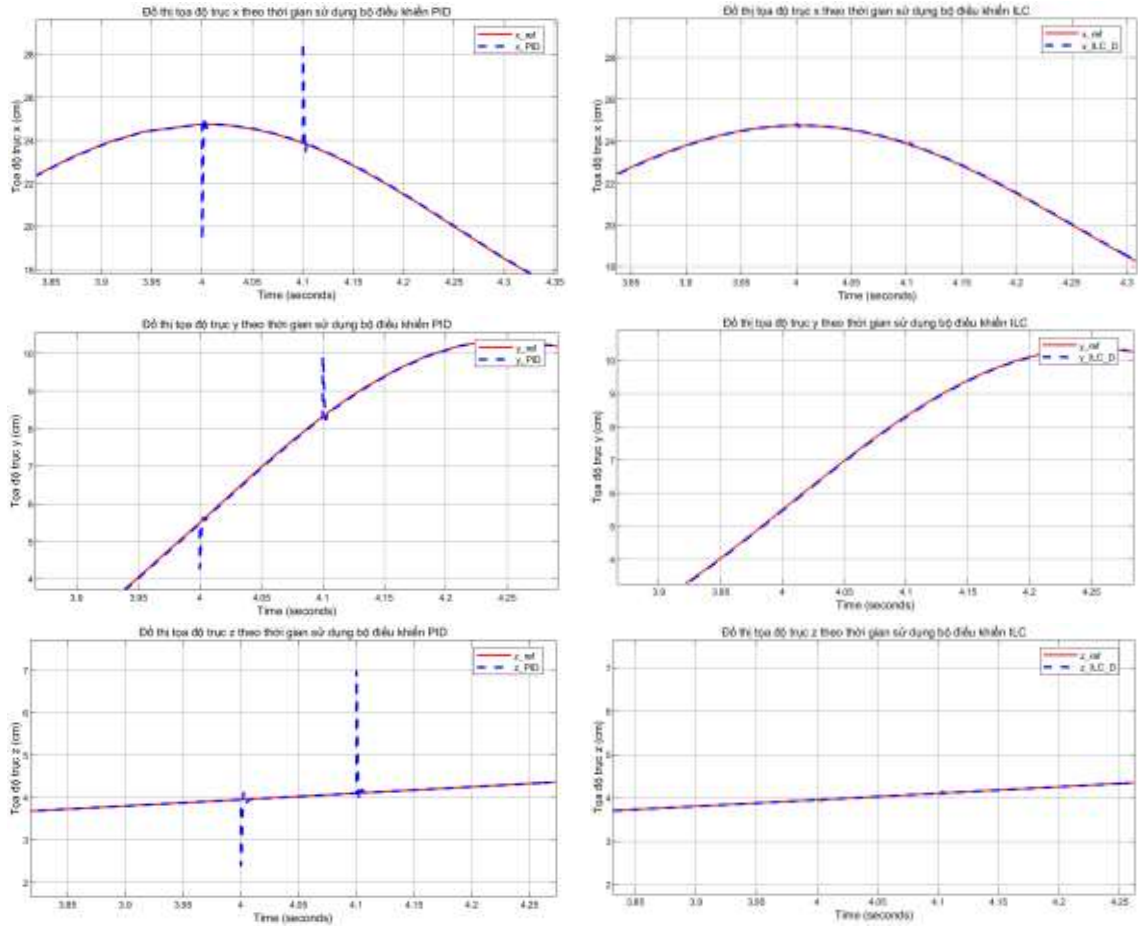
4.3.5.2 Kết quả mô phỏng in 3D khi có nhiễu tác động

- Kết quả so sánh tọa độ các trục x y z giữa hai phương pháp điều khiển PID (hình 4.68 bên trái) và PID kết hợp ILC kiểu D (hình 4.68 bên phải) cho thấy độ quá điều chỉnh đã giảm bớt so với khi chưa dùng ILC.



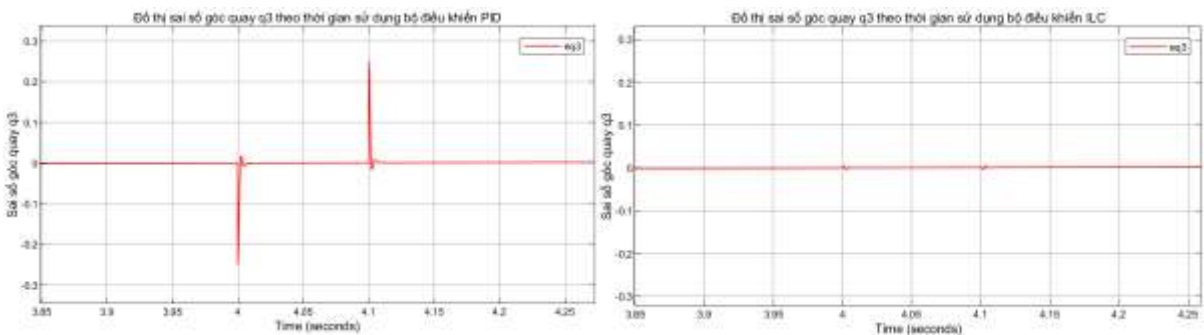
Hình 4.68: Đồ thị so sánh tọa độ x y z đầu ra cho hình in 3D của mô hình SolidWork giữa PID với PID kết hợp ILC kiểu D

- Kết quả đồ thị so sánh khả năng khử nhiễu tại tọa độ ba trục x y z giữa hai phương pháp điều khiển PID (hình 4.69 bên trái) và PID kết hợp ILC kiểu D (hình 4.69 bên phải) cho thấy khi tích hợp thêm ILC vào thì chỉ sau 2 lần học là nhiễu đã được xử lý rất tốt.



Hình 4.69: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D

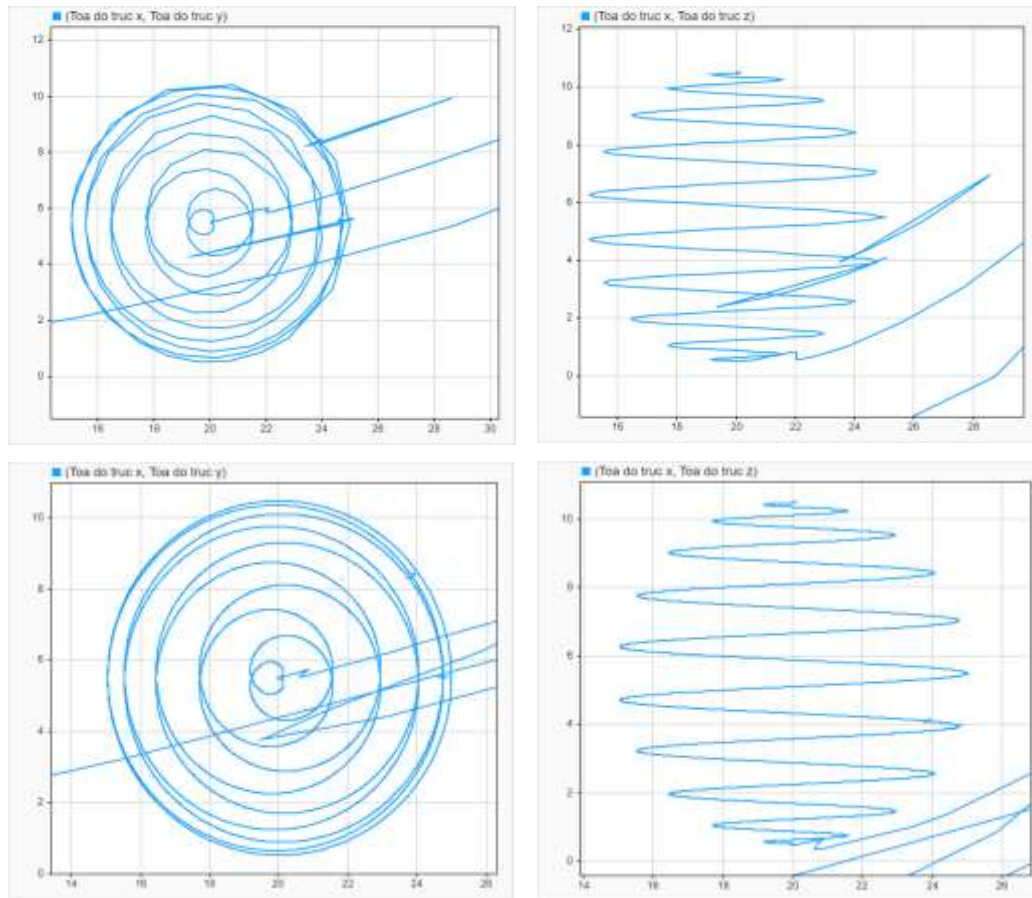
- Kết quả so sánh sai lệch góc quay q_3 khi có nhiễu tác động vào giữa hai phương pháp điều khiển PID (hình 4.70 bên trái) và PID kết hợp ILC kiểu D (hình 4.70 bên phải) cho thấy sai lệch do nhiễu gây ra tại góc quay q_3 đã giảm đáng kể khi tích hợp thêm bộ điều khiển ILC vào.



Hình 4.70: Đồ thị so sánh sai lệch góc quay q_3 khi in 3D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D

- Kết quả so sánh hình in 3D trên hai mặt phẳng Oxy và Oxz giữa hai phương pháp PID (hình 4.71 ở trên) và PID kết hợp ILC kiểu D (hình 4.71 ở dưới) cho thấy đường

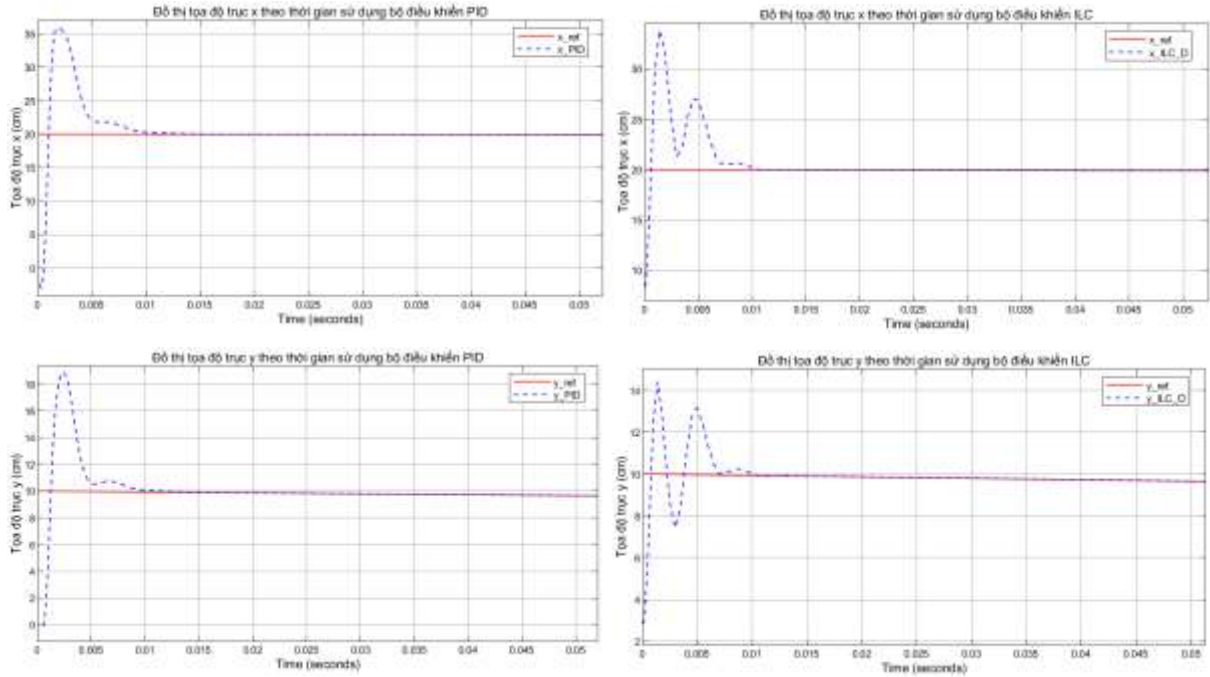
đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình in.



Hình 4.71: Đồ thị so sánh đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D khi có nhiễu

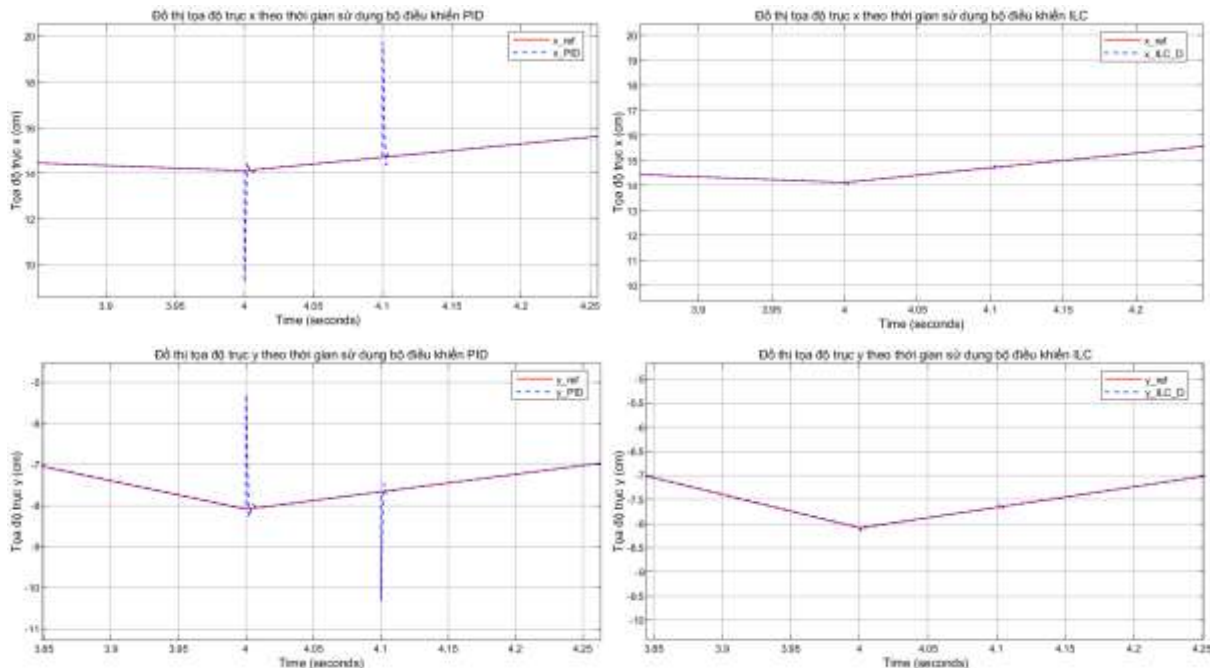
4.3.5.3 Kết quả mô phỏng vẽ 2D khi có nhiễu tác động

- Kết quả so sánh tọa độ các trục x y giữa hai phương pháp điều khiển PID (hình 4.72 bên trái) và PID kết hợp ILC kiểu D (hình 4.72 bên phải) cho thấy độ quá điều chỉnh đã giảm bớt so với khi chưa dùng ILC.



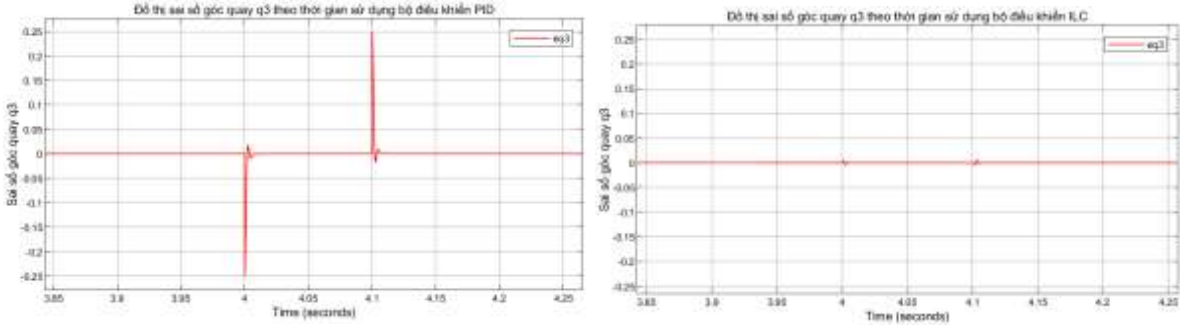
Hình 4.72: Đồ thị tọa độ x y z đầu ra cho hình vẽ 2D của mô hình SolidWork sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D

- Kết quả đồ thị so sánh khả năng khử nhiễu tại tọa độ ba trục x y giữa hai phương pháp điều khiển PID (hình 4.73 bên trái) và PID kết hợp ILC kiểu D (hình 4.73 bên phải) cho thấy khi tích hợp thêm ILC vào thì chỉ sau 2 lần học là nhiễu đã được xử lý rất tốt.



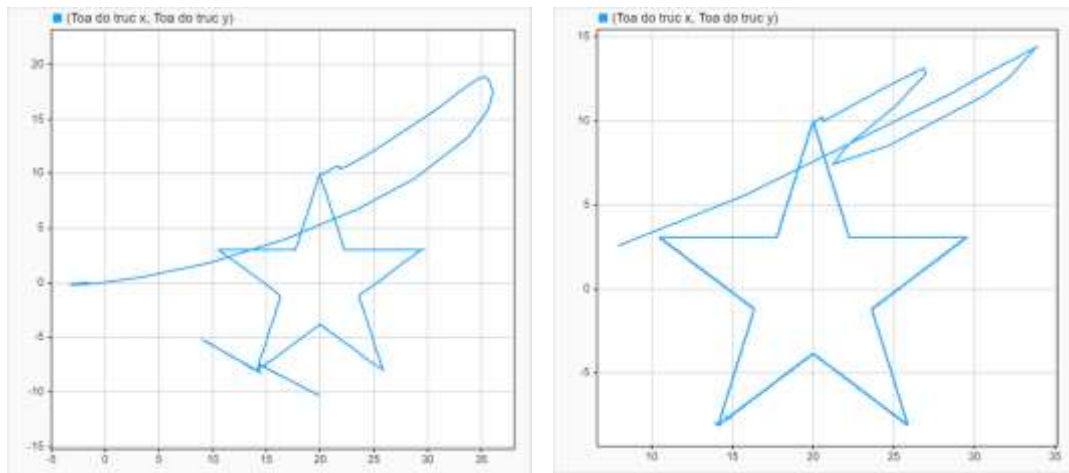
Hình 4.73: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D

- Kết quả so sánh sai lệch góc quay q_3 khi có nhiễu tác động vào giữa hai phương pháp điều khiển PID (hình 4.74 bên trái) và PID kết hợp ILC kiểu D (hình 4.74 bên phải) cho thấy sai lệch do nhiễu gây ra tại góc quay q_3 đã giảm đáng kể khi tích hợp thêm bộ điều khiển ILC vào.



Hình 4.74: Đồ thị so sánh sai lệch góc quay q_3 khi vẽ 2D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D

- Kết quả so sánh hình vẽ 2D trên mặt phẳng Oxy giữa hai phương pháp PID (hình 4.75 bên trái) và PID kết hợp ILC kiểu D (hình 4.75 bên phải) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình vẽ.



Hình 4.75: Đồ thị so sánh hình vẽ 2D đầu ra của mô hình SolidWork giữa bộ điều khiển PID và bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D

4.3.5.4 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.

- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0099	0.0128	0.0639%	75.8%
Trục y	0.0099	0.0127	0.2307%	52%
Trục z	0.012	0.001	0.0186%	1464%

Bảng 4.11: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình SolidWork khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.0099	0.0048	0.0238%	69%
Trục y	0.01	0.0026	0.0712%	44.3%

Bảng 4.12: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu D của mô hình SolidWork khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do xây dựng trên SolidWork thực hiện in 3D và vẽ 2D bằng phương pháp sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu D được thể hiện trong Bảng 4.11 và Bảng 4.12, có thể nhận thấy thời gian xác lập t_s không thay đổi nhiều do bộ PID đã tối ưu tốt tiêu chí này, phần trăm sai số xác lập trung bình $\varepsilon\%$ đã giảm so với khi chỉ sử dụng bộ điều khiển PID, còn độ quá điều chỉnh POT (%) giảm khá nhiều tuy nhưng vẫn còn rất cao so với tiêu chí đề ra.
- Đối với nhiễu tác động, kết quả cho thấy khi sử dụng kết hợp bộ điều khiển PID và ILC, nhiễu đã được loại bỏ gần như hoàn toàn chỉ sau 2 lần học. Điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả, vượt trội so với việc chỉ dùng bộ điều khiển PID.

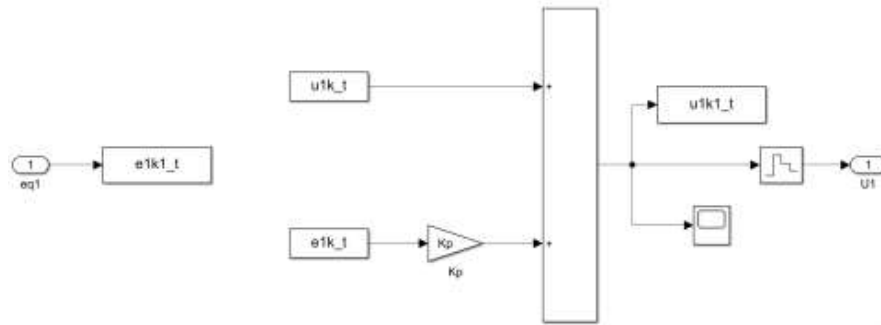
4.3.6. Kết quả mô phỏng điều khiển mô hình SolidWork khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P

Dựa vào công thức học lặp của bộ điều khiển ILC kiểu P, ta tiến hành mô phỏng bộ điều khiển ILC kiểu P trên Matlab Simulink như sau:

- Công thức học lặp của bộ điều khiển ILC kiểu P ở miền rời rạc:

$$u_{k+1}(t) = u_k(t) + K_p e_k(t)$$

- Mô phỏng bộ điều khiển ILC kiểu P trên Matlab Simulink:

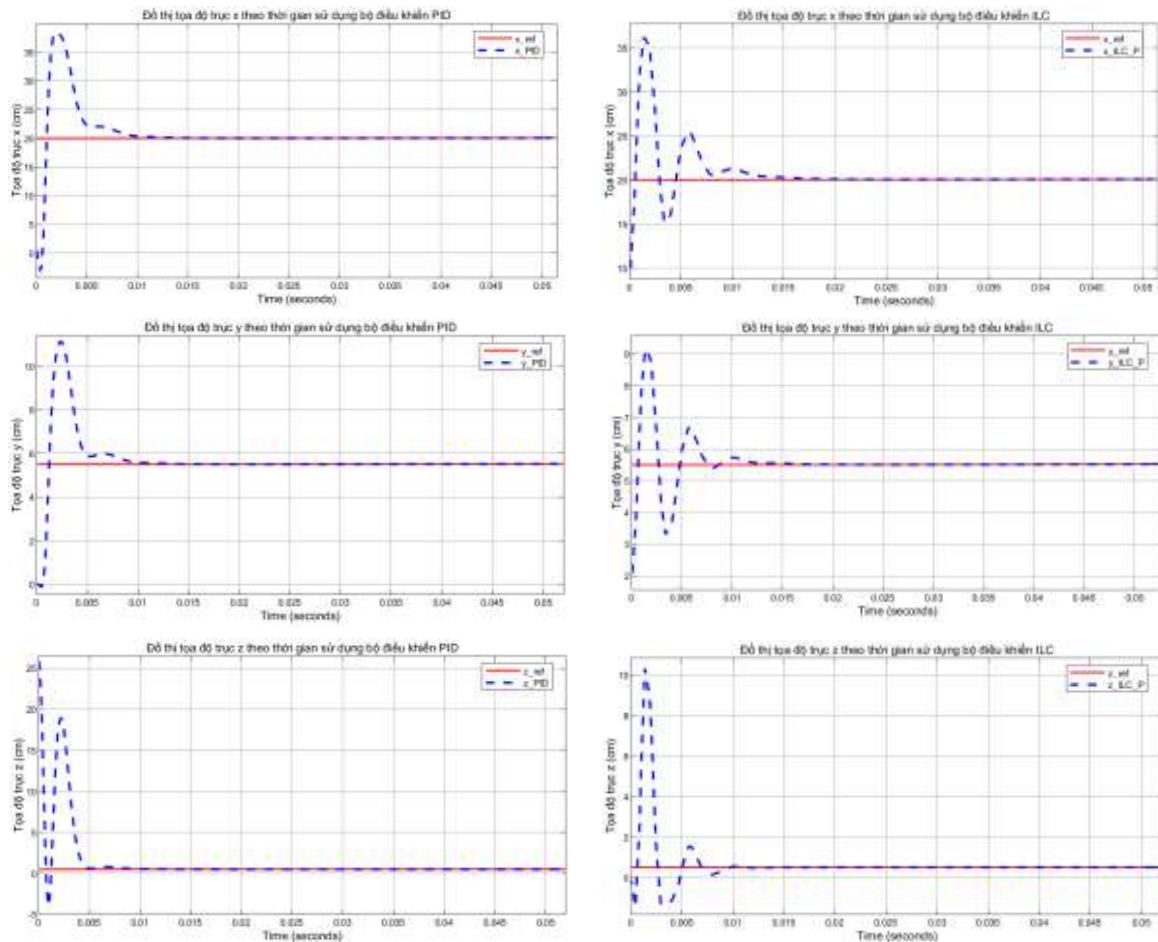


Hình 4.76: Sơ đồ mô phỏng bộ điều khiển ILC kiểu P cho mô hình SolidWork

Sau nhiều lần tinh chỉnh hệ số K_p thì ta lựa chọn được thông số của bộ điều khiển ILC kiểu P với hệ số K_p và số lần học n là $K_p = 0.15$ và $n = 10$.

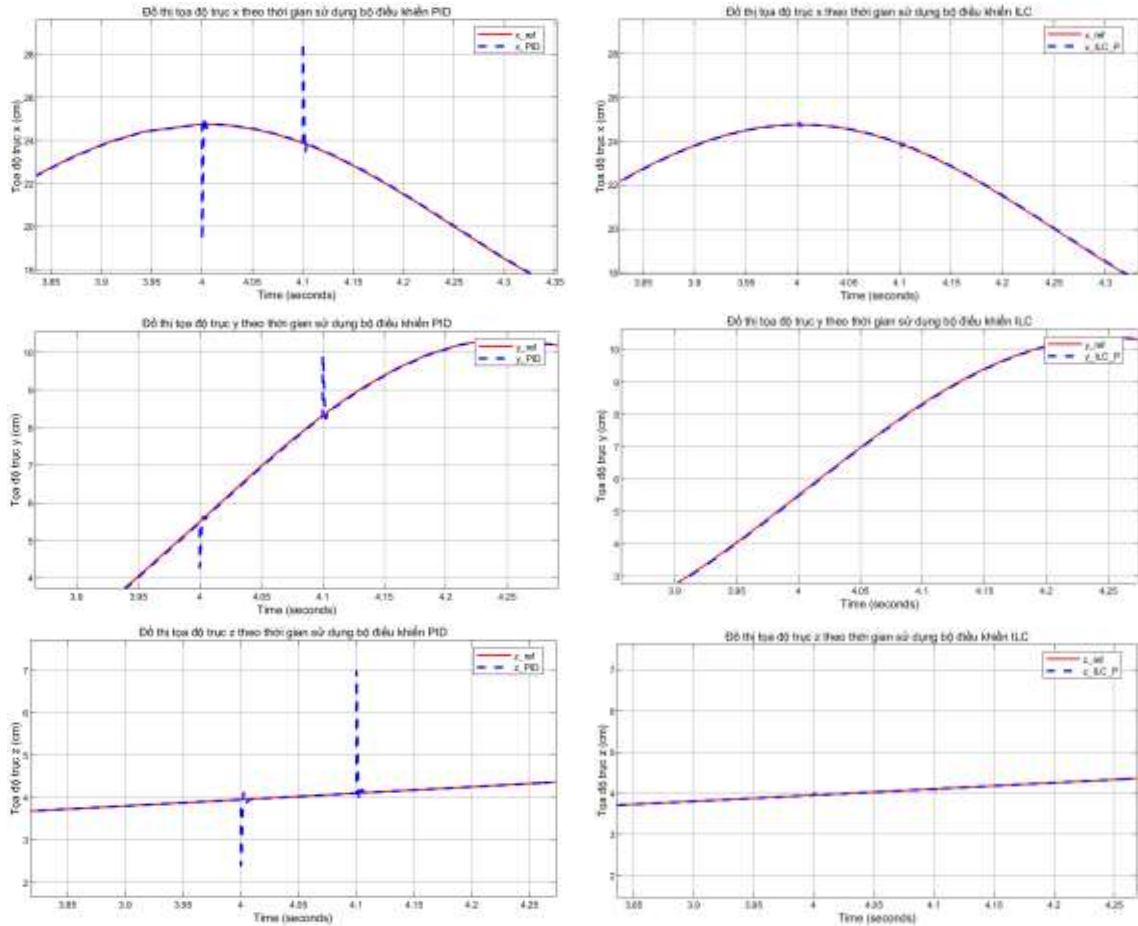
4.3.6.2 Kết quả mô phỏng in 3D khi có nhiễu tác động

- Kết quả so sánh tọa độ các trục x y z giữa hai phương pháp điều khiển PID (hình 4.77 bên trái) và PID kết hợp ILC kiểu P (hình 4.77 bên phải) cho thấy độ quá điều chỉnh đã giảm bớt so với khi chưa dùng ILC. Tuy nhiên, thời gian xác lập tăng nhẹ do bộ điều khiển ILC kiểu P giảm sai số từ từ qua từng vòng học lặp cần nhiều vòng lặp để đạt sai số nhỏ.



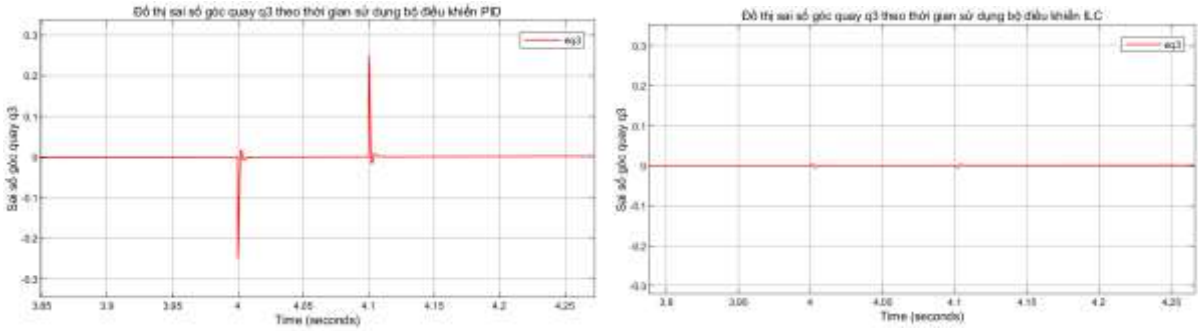
Hình 4.77: Đồ thị so sánh tọa độ x y z đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P

- Kết quả đồ thị so sánh khả năng khử nhiễu tại tọa độ ba trục x y z giữa hai phương pháp điều khiển PID (hình 4.78 bên trái) và PID kết hợp ILC kiểu P (hình 4.78 bên phải) cho thấy khi tích hợp thêm ILC vào thì chỉ sau 2 lần học là nhiễu đã được xử lý rất tốt.



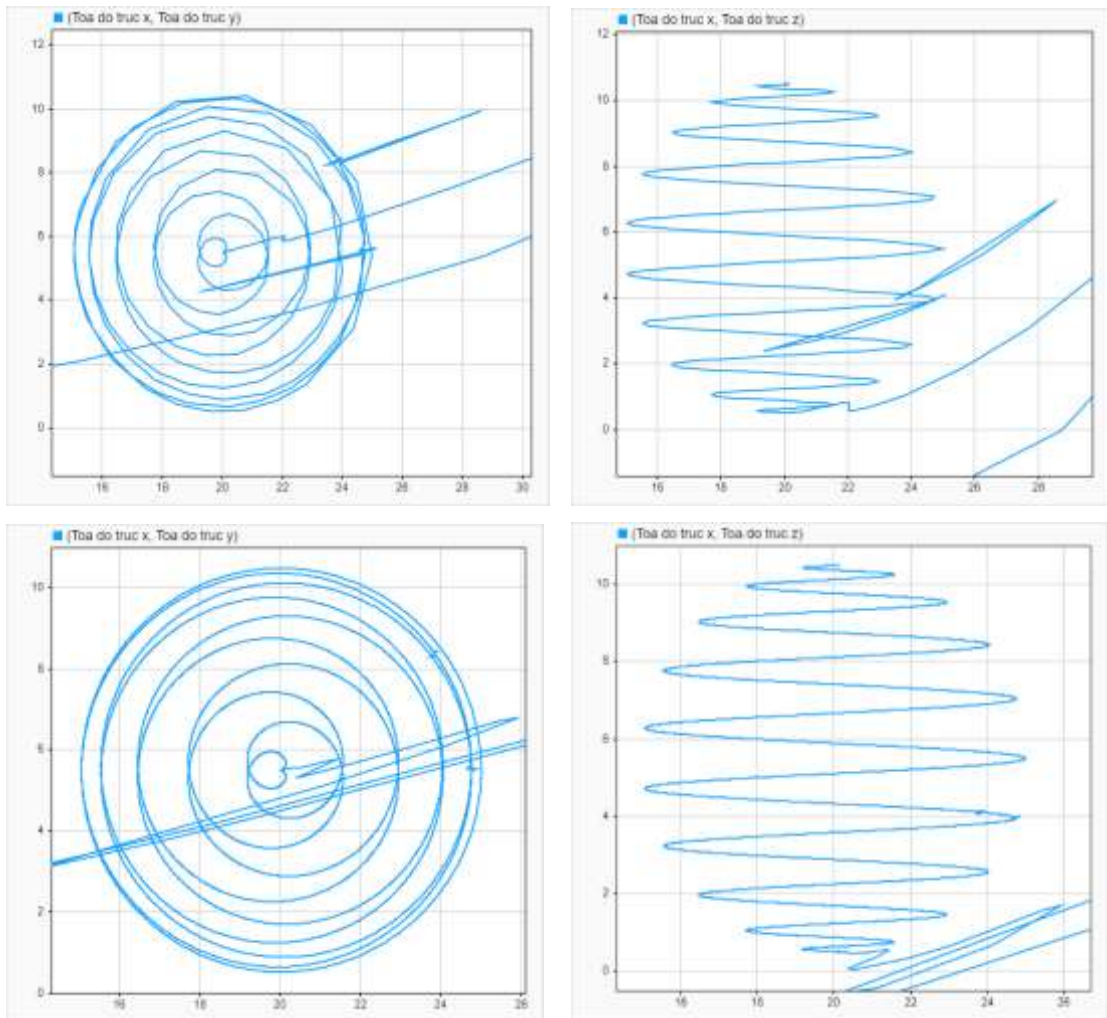
Hình 4.78: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P

- Kết quả so sánh sai lệch góc quay q_3 khi có nhiễu tác động vào giữa hai phương pháp điều khiển PID (hình 4.79 bên trái) và PID kết hợp ILC kiểu P (hình 4.79 bên phải) cho thấy sai lệch do nhiễu gây ra tại góc quay q_3 đã giảm đáng kể khi tích hợp thêm bộ điều khiển ILC vào.



Hình 4.79: Đồ thị so sánh sai lệch góc quay q_3 khi in 3D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P

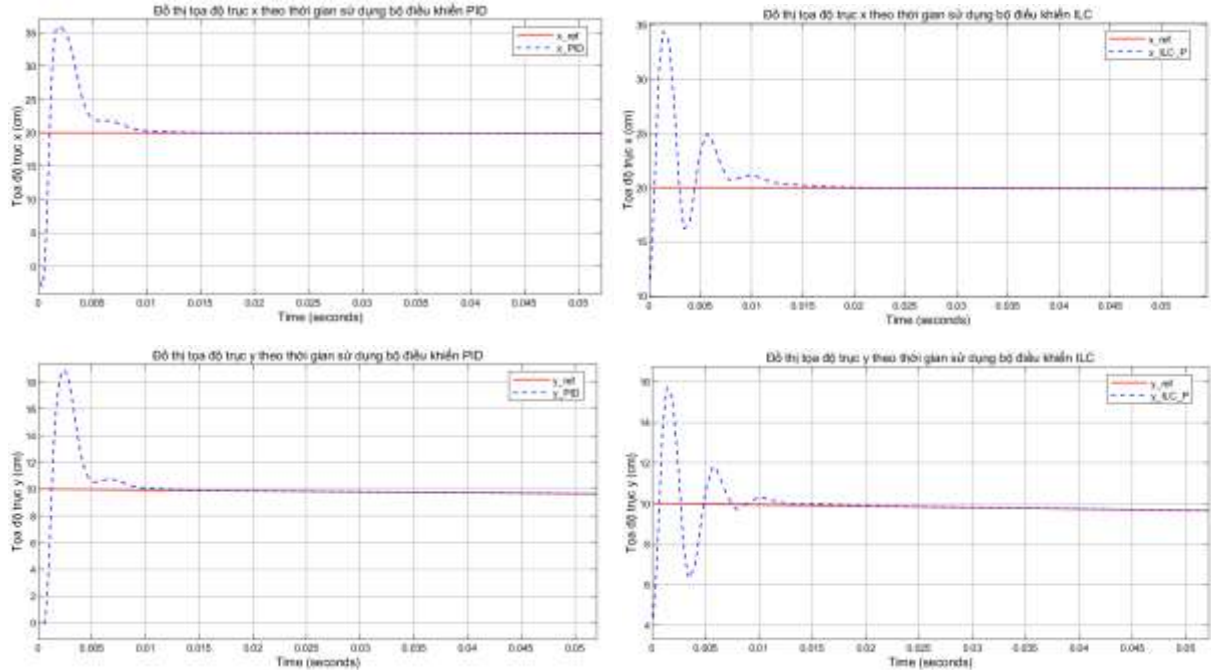
- Kết quả so sánh hình in 3D trên hai mặt phẳng Oxy và Oxz giữa hai phương pháp PID (hình 4.80 ở trên) và PID kết hợp ILC kiểu P (hình 4.80 ở dưới) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình in.



Hình 4.80: Đồ thị so sánh đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P khi có nhiễu

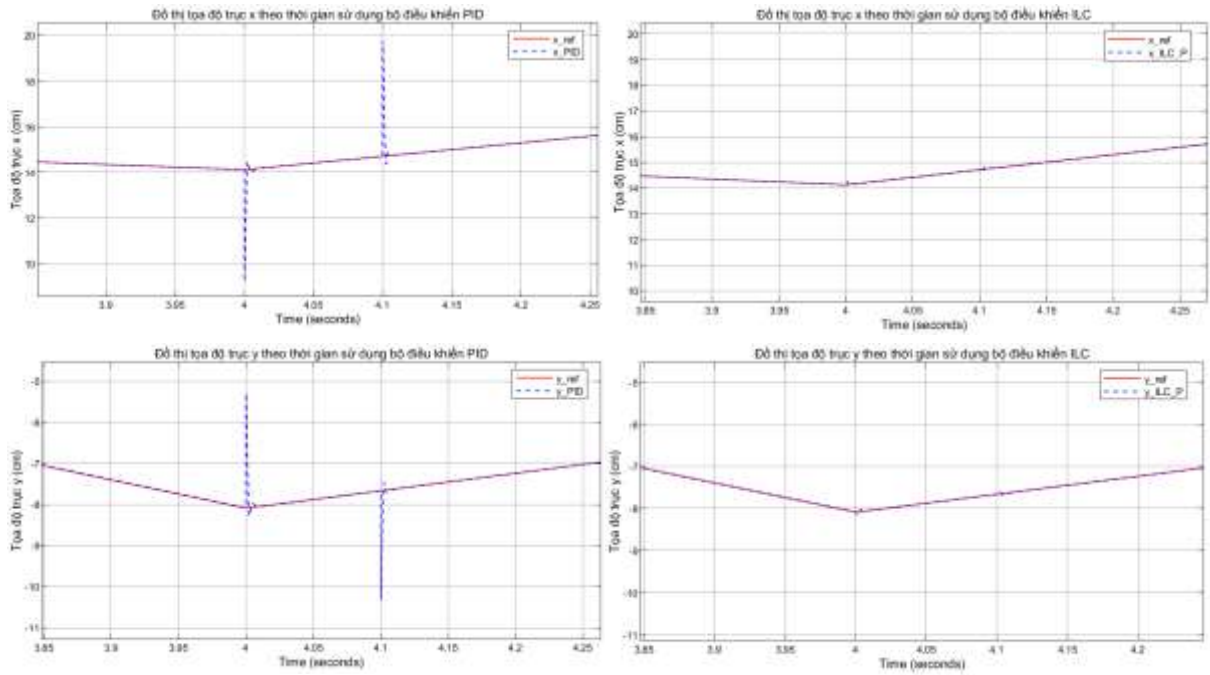
4.3.6.3 Kết quả mô phỏng vẽ 2D khi có nhiễu tác động

- Kết quả so sánh tọa độ các trục x y giữa hai phương pháp điều khiển PID (hình 4.81 bên trái) và PID kết hợp ILC kiểu P (hình 4.81 bên phải) cho thấy độ quá điều chỉnh đã giảm bớt so với khi chưa dùng ILC. Tuy nhiên, thời gian xác lập tăng nhẹ do bộ điều khiển ILC kiểu P giảm sai số từ từ qua từng vòng học lặp cần nhiều vòng lặp để đạt sai số nhỏ.



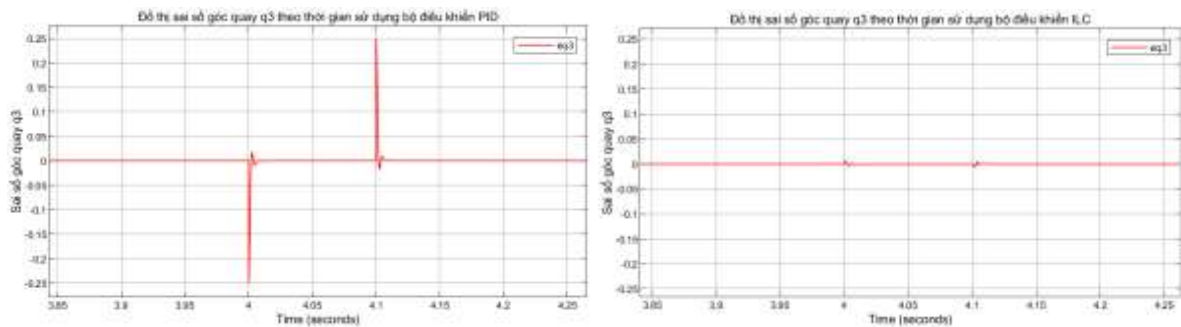
Hình 4.81: Đồ thị so sánh tọa độ x y đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P

- Kết quả đồ thị so sánh khả năng khử nhiễu tại tọa độ 2 trục x y giữa hai phương pháp điều khiển PID (hình 4.82 bên trái) và PID kết hợp ILC kiểu P (hình 4.82 bên phải) cho thấy khi tích hợp thêm ILC vào thì chỉ sau 2 lần học là nhiễu đã được xử lý rất tốt.



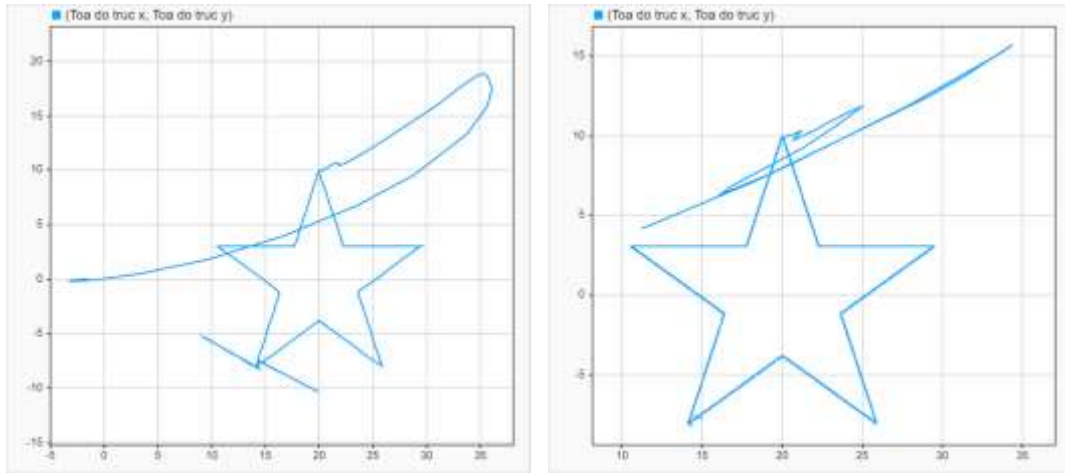
Hình 4.82: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P

- Kết quả so sánh sai lệch góc quay q_3 khi có nhiễu tác động vào giữa hai phương pháp điều khiển PID (hình 4.83 bên trái) và PID kết hợp ILC kiểu P (hình 4.83 bên phải) cho thấy sai lệch do nhiễu gây ra tại góc quay q_3 đã giảm đáng kể khi tích hợp thêm bộ điều khiển ILC vào.



Hình 4.83: Đồ thị so sánh sai lệch góc quay q_3 khi vẽ 2D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu P

- Kết quả so sánh hình vẽ 2D trên mặt phẳng Oxy giữa hai phương pháp PID (hình 4.84 bên trái) và PID kết hợp ILC kiểu P (hình 4.84 bên phải) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình vẽ.



Hình 4.84: Đồ thị so sánh hình vẽ 2D đầu ra của mô hình SolidWork giữa bộ điều khiển PID và bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P

4.3.6.4 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.
- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.013	0.0129	0.0645%	79.9%
Trục y	0.012	0.0127	0.2311%	66.5%
Trục z	0.014	0.00104	0.0189%	1962%

Bảng 4.13: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình SolidWork khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{tb}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.013	0.0049	0.0244%	71.8%
Trục y	0.012	0.0028	0.0756%	57.2%

Bảng 4.14: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu P của mô hình SolidWork khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do xây dựng trên SolidWork thực hiện in 3D và vẽ 2D bằng phương pháp sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu P được thể hiện trong Bảng 4.13 và Bảng 4.14, có thể nhận thấy thời gian xác lập t_s tăng nhẹ, tuy nhiên phần trăm sai số xác lập trung bình $\varepsilon\%$ đã giảm so với khi chỉ sử dụng bộ điều khiển PID, còn độ quá điều chỉnh $POT(\%)$ giảm so với PID tuy nhiên vẫn còn rất cao so với tiêu chí đề ra.
- Đối với nhiễu tác động, kết quả cho thấy khi sử dụng kết hợp bộ điều khiển PID và ILC, nhiễu đã được loại bỏ gần như hoàn toàn chỉ sau 2 lần học. Điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả, vượt trội so với việc chỉ dùng bộ điều khiển PID.

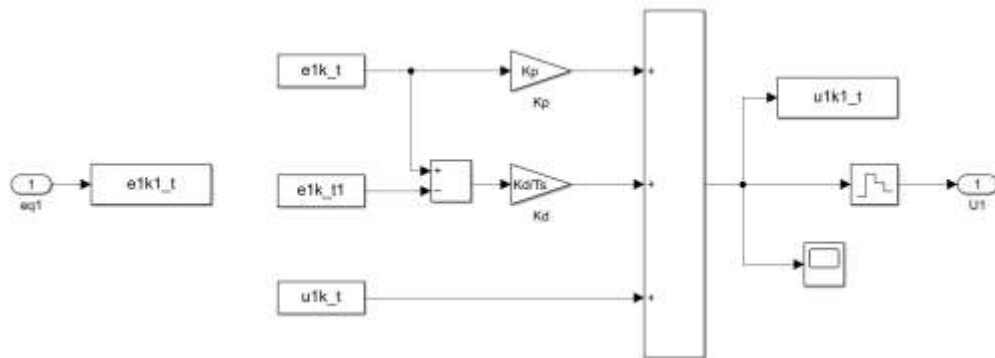
4.3.7. Kết quả mô phỏng điều khiển mô hình SolidWork khi sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD

Dựa vào công thức học lặp của bộ điều khiển ILC kiểu PD, ta tiến hành mô phỏng bộ điều khiển ILC kiểu PD trên Matlab Simulink như sau:

- Công thức học lặp của bộ điều khiển ILC kiểu PD ở miền rời rạc:

$$u_{k+1}(t) = u_k(t) + K_p e_k(t) + K_d \frac{e_k(t) - e_k(t-1)}{T_s}$$

- Mô phỏng bộ điều khiển ILC kiểu PD trên Matlab Simulink:

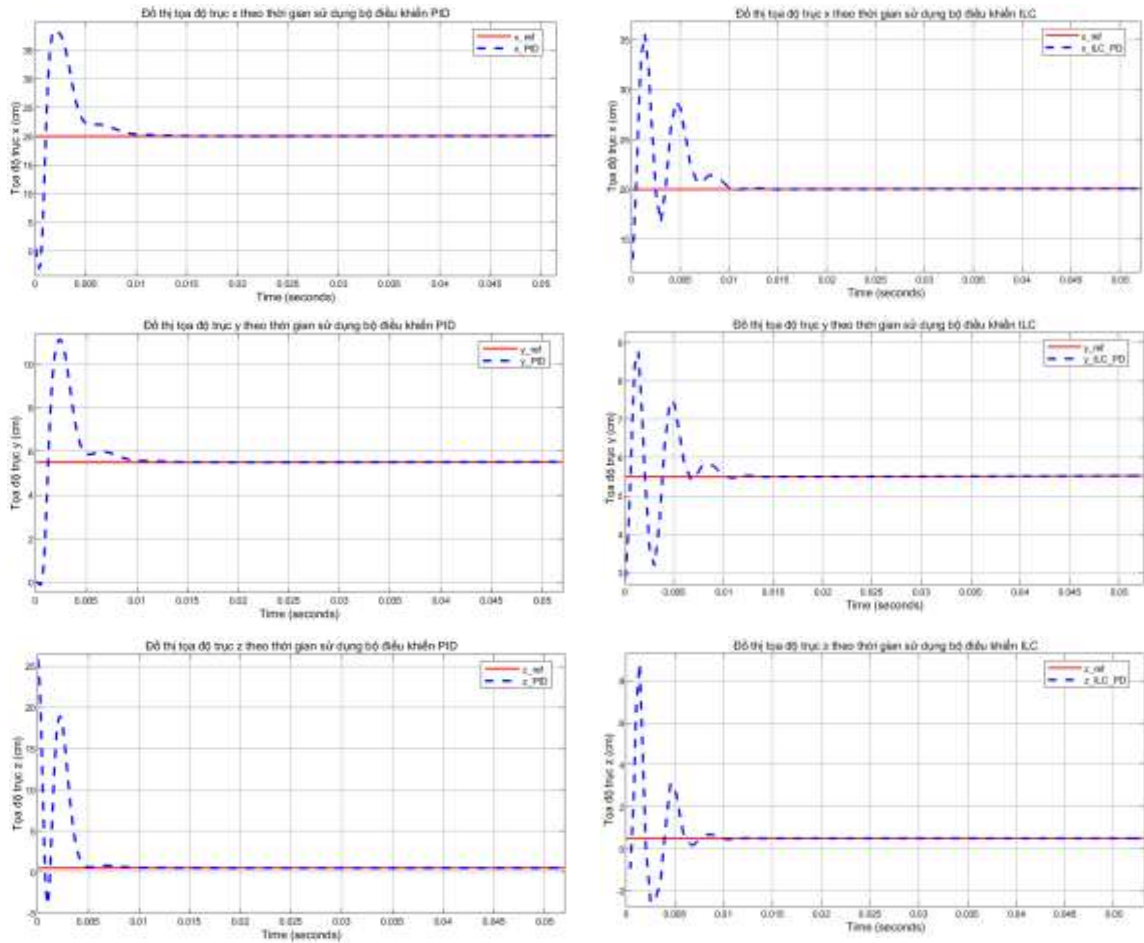


Hình 4.85: Sơ đồ mô phỏng bộ điều khiển ILC kiểu PD cho mô hình SolidWork

Sau nhiều lần tinh chỉnh hệ số K_p và K_d thì ta lựa chọn được thông số của bộ điều khiển ILC kiểu PD với hệ số K_p, K_d và số lần học n là $K_p = 0.02; K_d = 0.0001$ và $n = 15$.

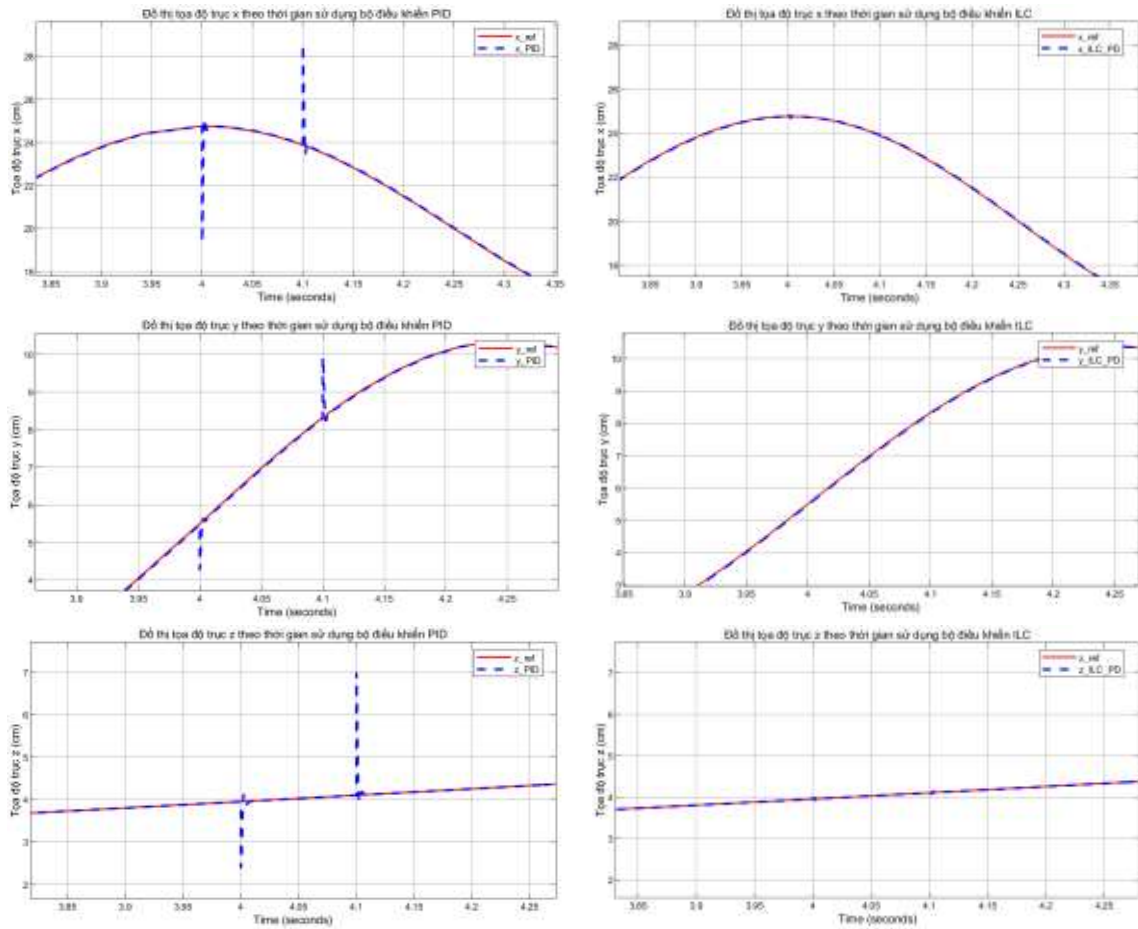
4.3.7.2 Kết quả mô phỏng in 3D khi có nhiễu tác động

- Kết quả so sánh tọa độ các trục x y z giữa hai phương pháp điều khiển PID (hình 4.86 bên trái) và PID kết hợp ILC kiểu PD (hình 4.86 bên phải) cho thấy độ quá điều chỉnh đã giảm bớt so với khi chưa dùng ILC.



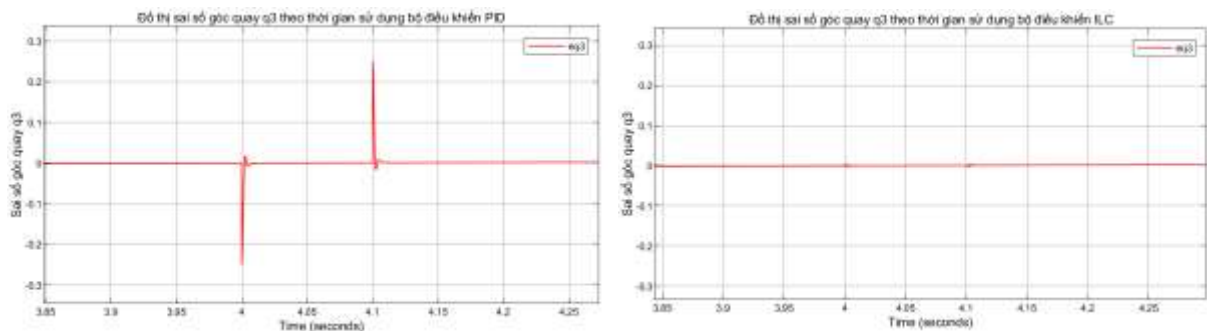
Hình 4.86: Đồ thị so sánh tọa độ $x y z$ đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD

- Kết quả đồ thị so sánh khả năng khử nhiễu tại tọa độ ba trục $x y z$ giữa hai phương pháp điều khiển PID (hình 4.87 bên trái) và PID kết hợp ILC kiểu PD (hình 4.87 bên phải) cho thấy khi tích hợp thêm ILC vào thì chỉ sau 2 lần học là nhiễu đã được xử lý rất tốt.



Hình 4.87: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD

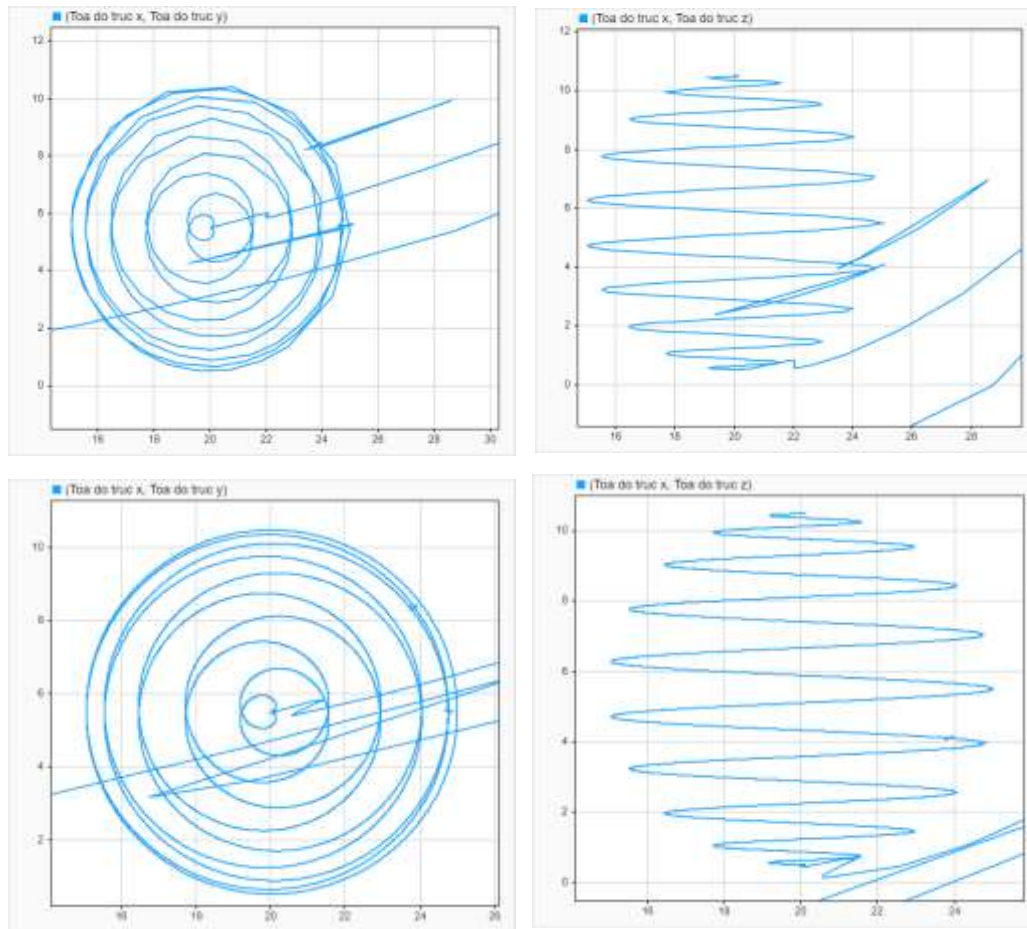
- Kết quả so sánh sai lệch góc quay q_3 khi có nhiễu tác động vào giữa hai phương pháp điều khiển PID (hình 4.79 bên trái) và PID kết hợp ILC kiểu P (hình 4.79 bên phải) cho thấy sai lệch do nhiễu gây ra tại góc quay q_3 đã giảm đáng kể khi tích hợp thêm bộ điều khiển ILC vào.



Hình 4.88: Đồ thị so sánh sai lệch góc quay q_3 khi in 3D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD

- Kết quả so sánh hình in 3D trên hai mặt phẳng Oxy và Oxz giữa hai phương pháp PID (hình 4.89 ở trên) và PID kết hợp ILC kiểu PD (hình 4.89 ở dưới) cho thấy

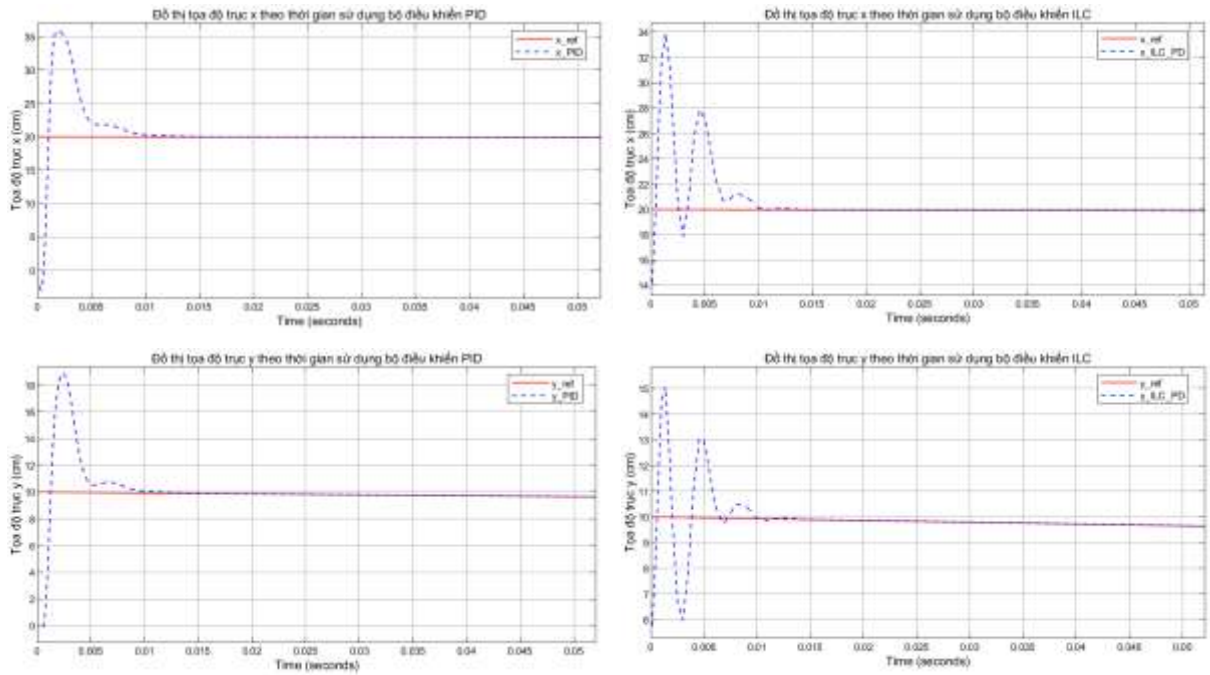
đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình in.



Hình 4.89: Đồ thị so sánh đồ thị theo 2 mặt phẳng Oxy và Oxz của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD khi có nhiễu

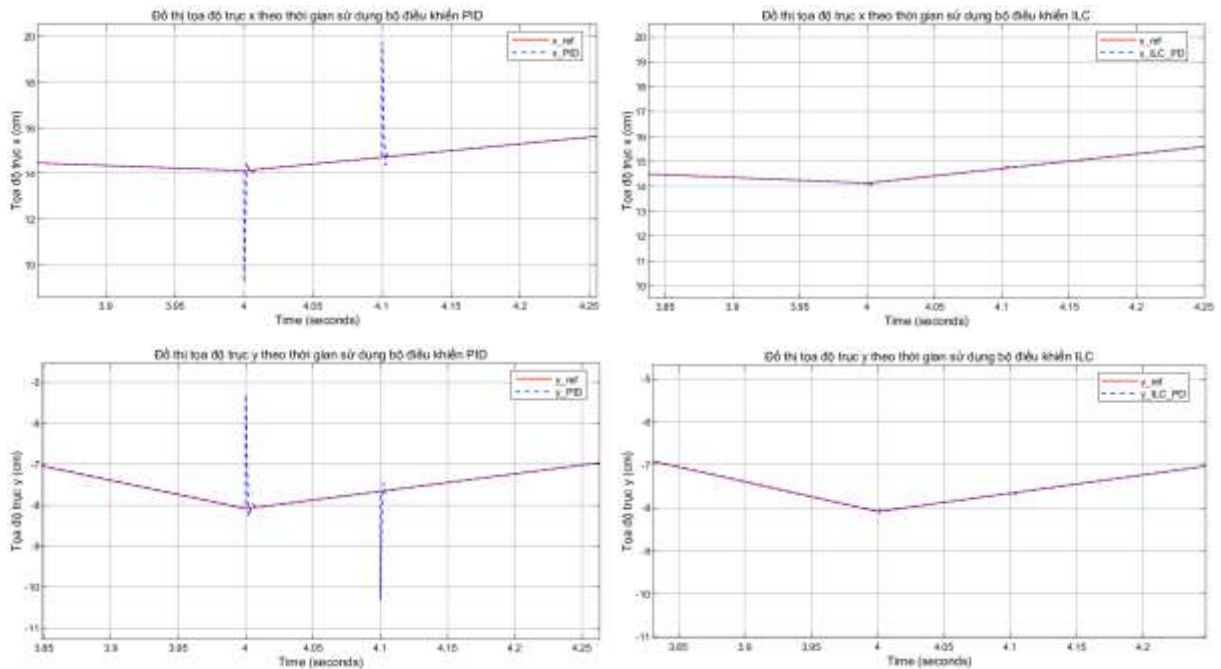
4.3.7.3 Kết quả mô phỏng vẽ 2D khi có nhiễu tác động

- Kết quả so sánh tọa độ các trục x y giữa hai phương pháp điều khiển PID (hình 4.90 bên trái) và PID kết hợp ILC kiểu PD (hình 4.90 bên phải) cho thấy độ quá điều chỉnh đã giảm bớt so với khi chưa dùng ILC.



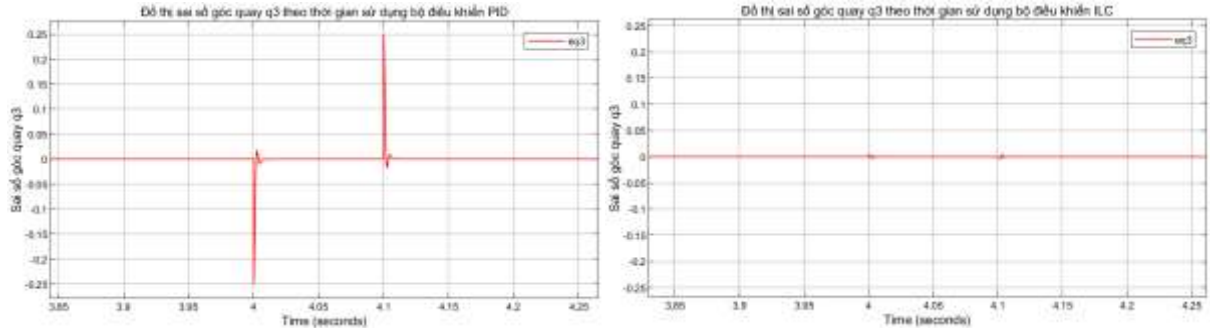
Hình 4.90: Đồ thị so sánh tọa độ x y đầu ra của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD

- Kết quả đồ thị so sánh khả năng khử nhiễu tại tọa độ 2 trục x y giữa hai phương pháp điều khiển PID (hình 4.91 bên trái) và PID kết hợp ILC kiểu PD (hình 4.91 bên phải) cho thấy khi tích hợp thêm ILC vào thì chỉ sau 2 lần học là nhiễu đã được xử lý rất tốt.



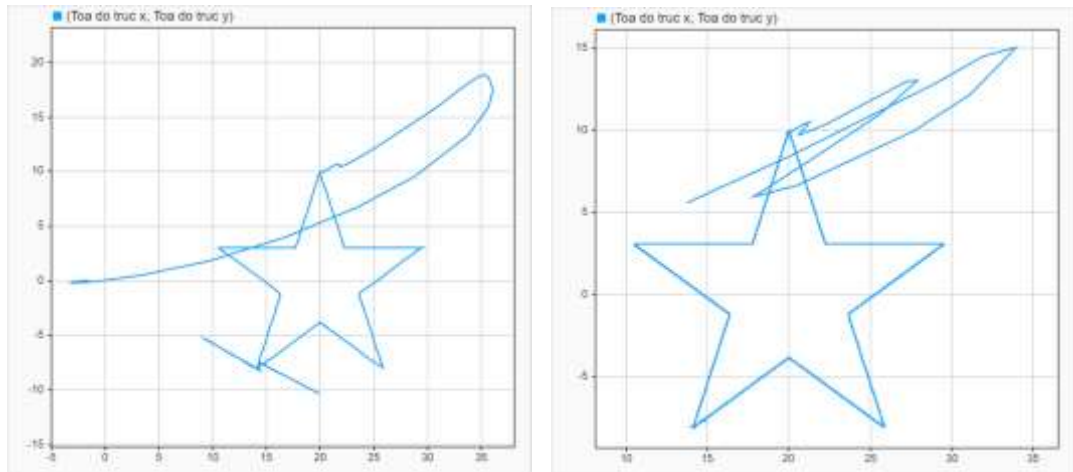
Hình 4.91: Đồ thị so sánh khả năng xử lý nhiễu của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD

- Kết quả so sánh sai lệch góc quay q_3 khi có nhiễu tác động vào giữa hai phương pháp điều khiển PID (hình 4.92 bên trái) và PID kết hợp ILC kiểu PD (hình 4.92 bên phải) cho thấy sai lệch do nhiễu gây ra tại góc quay q_3 đã giảm đáng kể khi tích hợp thêm bộ điều khiển ILC vào.



Hình 4.92: Đồ thị so sánh sai lệch góc quay q_3 khi vẽ 2D của mô hình SolidWork giữa bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu PD

- Kết quả so sánh hình vẽ 2D trên mặt phẳng Oxy giữa hai phương pháp PID (hình 4.93 bên trái) và PID kết hợp ILC kiểu PD (hình 4.93 bên phải) cho thấy đường đi của khâu chấp hành khi dùng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD đã được xử lý nhiễu hoàn toàn, đường đi không còn bị lệch gây ảnh hưởng đến chất lượng đầu ra của hình vẽ.



Hình 4.93: Đồ thị so sánh hình vẽ 2D đầu ra của mô hình SolidWork giữa bộ điều khiển PID và bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD

4.3.7.4 Đánh giá chất lượng hệ thống

Dựa trên các tiêu chí chất lượng mà chúng ta đã đề ra từ phần trước thì ta có các chỉ số chất lượng sau:

- Thời gian xác lập $t_s \leq 0.15$ giây.
- Sai số xác lập trung bình e_{tb} .
- Phần trăm sai số xác lập trung bình $\varepsilon\% \leq 2\%$.

- Phần trăm độ quá điều chỉnh $POT \leq 60\%$.

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y z đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện in 3D như sau:

	t_s (giây)	e_{ib}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.01	0.0128	0.0639%	75.4%
Trục y	0.01	0.0127	0.2306%	58.1%
Trục z	0.014	0.001025	0.0186%	1651.2%

Bảng 4.15: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình SolidWork khi in 3D

Ta xác định được bảng tiêu chí chất lượng cho tọa độ x y đầu ra của hệ thống cánh tay Robot 3 bậc tự do khi thực hiện vẽ 2D như sau:

	t_s (giây)	e_{ib}	$\varepsilon\%$ (%)	POT (%)
Trục x	0.01	0.0047	0.0237%	69.1%
Trục y	0.01	0.0025	0.0703%	50%

Bảng 4.16: Bảng chỉ tiêu đánh giá chất lượng hệ thống sử dụng bộ điều khiển PID kết hợp ILC kiểu PD của mô hình SolidWork khi vẽ 2D

Nhận xét:

- Dựa trên kết quả đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do xây dựng trên SolidWork thực hiện in 3D và vẽ 2D bằng phương pháp sử dụng bộ điều khiển PID kết hợp bộ điều khiển ILC kiểu PD được thể hiện trong Bảng 4.15 và Bảng 4.16, có thể nhận thấy thời gian xác lập t_s không thay đổi nhiều do bộ PID đã tối ưu tốt tiêu chí này, tuy nhiên phần trăm sai số xác lập trung bình $\varepsilon\%$ đã giảm so với khi chỉ sử dụng bộ điều khiển PID, còn độ quá điều chỉnh POT (%) giảm khá nhiều tuy nhưng vẫn còn rất cao so với tiêu chí đề ra.
- Đối với nhiều tác động, kết quả cho thấy khi sử dụng kết hợp bộ điều khiển PID và ILC, nhiễu đã được loại bỏ gần như hoàn toàn chỉ sau 2 lần học. Điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả, vượt trội so với việc chỉ dùng bộ điều khiển PID.

4.4. So sánh, nhận xét kết quả mô phỏng giữa 2 phương pháp điều khiển sử dụng bộ điều khiển PID với bộ điều khiển PID kết hợp ILC

4.4.1. So sánh, nhận xét kết quả mô phỏng mô hình động lực học giữa 2 phương pháp điều khiển

Sau khi thực hiện mô phỏng điều khiển mô hình động lực học cánh tay Robot 3 bậc tự do thực hiện in 3D và vẽ 2D bằng hai phương pháp điều khiển là sử dụng bộ điều khiển PID và sử dụng bộ điều khiển PID kết hợp các bộ điều khiển ILC kiểu D, P, PD

thì ta thu được các thông số đánh giá chất lượng hệ thống. Dựa vào các thông số đó, chúng ta lập bảng so sánh giữa 2 phương pháp điều khiển cho mô hình động lực học như sau:

- Đối với hình in 3D:

		PID	PID + ILC D	PID + ILC P	PID + ILC PD
Trục x	t_s (giây)	0.0383	0.0378	0.0137	0.0129
	e_{ib}	0.0069	0.0059	0.0065	0.0066
	$\varepsilon\%$ (%)	0.0345%	0.0293%	0.0326%	0.0329%
	POT (%)	15.24%	23.7%	43.35%	51.1%
Trục y	t_s (giây)	0.0258	0.0245	0.0069	0.0083
	e_{ib}	0.0018	0.0014	0.0014	0.0014
	$\varepsilon\%$ (%)	0.0322%	0.0258%	0.0249%	0.0246%
	POT (%)	9.58%	9.4%	28.6%	36.2%
Trục z	t_s (giây)	0.0808	0.0796	0.1062	0.1075
	e_{ib}	0.0065	0.0062	0.0062	0.0061
	$\varepsilon\%$ (%)	0.1174%	0.1113%	0.1109%	0.1107%
	POT (%)	241.8%	300%	532.4%	645.5%

Bảng 4.17: Bảng so sánh kết quả mô phỏng mô hình động lực học khi in 3D giữa các phương pháp điều khiển khác nhau

- Đối với hình vẽ 2D:

		PID	PID + ILC D	PID + ILC P	PID + ILC PD
Trục x	t_s (giây)	0.0383	0.0375	0.0147	0.0133
	e_{ib}	0.0037	0.0029	0.0037	0.0037
	$\varepsilon\%$ (%)	0.0189%	0.0144%	0.0184%	0.0184%
	POT (%)	14.56%	22.36%	40%	47.5%
Trục y	t_s (giây)	0.0191	0.016	0.01	0.01
	e_{ib}	0.002	0.0018	0.0017	0.0015
	$\varepsilon\%$ (%)	0.3474%	0.334%	0.312%	0.309%
	POT (%)	7.06%	9.4%	22.78%	29.1%

Bảng 4.18: Bảng so sánh kết quả mô phỏng mô hình động lực học khi vẽ 2D giữa các phương pháp điều khiển khác nhau

Nhận xét: Dựa trên kết quả so sánh đánh giá chất lượng hệ thống mô hình động lực học cánh tay Robot 3 bậc tự do thực hiện in 3D và vẽ 2D giữa các phương pháp sử dụng bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D, P và PD được thể hiện trong Bảng 4.17 và Bảng 4.18, ta có thể thấy:

- Đối với phương pháp sử dụng bộ điều khiển PID: giúp cho hệ thống đạt được thời gian xác lập t_s , sai số xác lập và độ quá điều chỉnh ở mức chấp nhận được, tuy độ quá điều chỉnh ở trục z còn khá cao. Bên cạnh đó, khi xử lý nhiễu thì bộ điều khiển PID không khử nhiễu hoàn toàn mà chỉ giúp hệ đáp ứng lại và duy trì được ổn định thôi, chứ giá trị sai số lúc xảy ra nhiễu vẫn còn khá cao gây ảnh hưởng đến chất lượng hình in và hình vẽ đầu ra.
- Đối với phương pháp PID kết hợp ILC kiểu D: Việc kết hợp PID với bộ điều khiển ILC kiểu D giúp cải thiện đáng kể sai số xác lập trung bình e_{tb} và phần trăm sai số xác lập trung bình $\varepsilon\%$ trên cả ba trục x y z đầu ra. Mức độ dao động $POT(\%)$ được kiểm soát ổn định ở mức trung bình, tuy có tăng nhẹ ở hai trục x và z. Còn về vấn đề khử nhiễu thì ILC kiểu D xử lý tốt được nhiễu chỉ sau 2 lần học, điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả.
- Đối với phương pháp PID kết hợp ILC kiểu P: Bộ điều khiển ILC kiểu P mang lại thời gian xác lập t_s ngắn hơn rất nhiều so với bộ PID và PID kết hợp ILC kiểu D, chứng tỏ khả năng đáp ứng nhanh của hệ thống. Tuy nhiên, nhược điểm lớn của phương pháp này là độ quá điều chỉnh $POT(\%)$ rất cao và không ổn định, đặc biệt tại trục z, làm giảm độ tin cậy và an toàn trong điều khiển. Phần trăm sai số xác lập trung bình $\varepsilon\%$ cũng được cải thiện vượt trội hơn so với ILC kiểu D. Còn về vấn đề khử nhiễu thì ILC kiểu P cũng xử lý tốt được nhiễu chỉ sau 2 lần học như ILC kiểu D, điều này chứng minh rằng việc tích hợp thêm bộ điều khiển ILC giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả.
- Đối với phương pháp PID kết hợp ILC kiểu PD: Đây là phương pháp cho kết quả tổng thể cân bằng tốt nhất giữa các chỉ tiêu: thời gian xác lập t_s nhanh, sai số trung bình nhỏ và sai số tương đối thấp. Tuy nhiên, tương tự như ILC kiểu P, độ quá điều chỉnh tại trục z vẫn tăng mạnh (có thể lên đến gần 700%), gây lo ngại trong các bài toán yêu cầu độ ổn định và chính xác cao. Ở trục x và y, các chỉ tiêu đều tăng so với PID đơn thuần, cho thấy cần thêm các phương pháp mới để giảm độ quá điều chỉnh. Còn về khả năng khử nhiễu cũng giống như hai loại D và P, xử lý nhiễu tốt chỉ sau 2 lần học.

Kết luận: Trong bốn phương pháp điều khiển được so sánh, PID kết hợp ILC kiểu PD và PID kết hợp ILC kiểu D là hai lựa chọn tối ưu hơn PID đơn thuần về độ chính xác và thời gian đáp ứng. Tuy nhiên, vấn đề độ quá điều chỉnh tại trục z vẫn còn tồn tại, đặc biệt khi sử dụng các phương pháp có thành phần tỉ lệ (ILC kiểu P và ILC kiểu PD). Do

đó, để đạt hiệu quả điều khiển toàn diện, cần tiếp tục nghiên cứu giải pháp giới hạn độ quá điều chỉnh.

4.4.2. So sánh, nhận xét kết quả mô phỏng mô hình SolidWork giữa 2 phương pháp điều khiển

Sau khi thực hiện mô phỏng điều khiển mô hình cánh tay Robot 3 bậc tự do xây dựng trên SolidWork thực hiện in 3D và vẽ 2D bằng hai phương pháp điều khiển là sử dụng bộ điều khiển PID và sử dụng bộ điều khiển PID kết hợp các bộ điều khiển ILC kiểu D, P, PD thì ta thu được các thông số đánh giá chất lượng hệ thống. Dựa vào các thông số đó, chúng ta lập bảng so sánh giữa 2 phương pháp điều khiển cho mô hình SolidWork như sau.

- Đối với hình in 3D:

		PID	PID + ILC D	PID + ILC P	PID + ILC PD
Trục x	t_s (giây)	0.01	0.0099	0.013	0.01
	e_{ib}	0.018	0.0128	0.0129	0.0128
	$\varepsilon\%$ (%)	0.0898%	0.0639%	0.0645%	0.0639%
	POT (%)	92.05%	75.8%	79.9%	75.4%
Trục y	t_s (giây)	0.01	0.0099	0.012	0.01
	e_{ib}	0.0146	0.0127	0.0127	0.0127
	$\varepsilon\%$ (%)	0.2825%	0.2307%	0.2311%	0.2306%
	POT (%)	102.2%	52%	66.5%	58.1%
Trục z	t_s (giây)	0.0141	0.012	0.014	0.014
	e_{ib}	0.0011	0.001	0.00104	0.001025
	$\varepsilon\%$ (%)	0.0227%	0.0186%	0.0189%	0.0186%
	POT (%)	3662%	1464%	1962%	1651.2%

Bảng 4.19: Bảng so sánh kết quả mô phỏng mô hình SolidWork khi in 3D giữa các phương pháp điều khiển khác nhau

- Đối với hình vẽ 2D:

		PID	PID + ILC D	PID + ILC P	PID + ILC PD
Trục x	t_s (giây)	0.01	0.0099	0.013	0.01
	e_{ib}	0.0112	0.0048	0.0049	0.0047
	$\varepsilon\%$ (%)	0.0584%	0.0238%	0.0244%	0.0237%
	POT (%)	80%	69%	71.8%	69.1%
Trục y	t_s (giây)	0.012	0.01	0.012	0.01
	e_{ib}	0.0032	0.0026	0.0028	0.0025
	$\varepsilon\%$ (%)	0.0869%	0.0712%	0.0756%	0.0703%

	POT (%)	88.7%	44.3%	57.2%	50%
--	---------	-------	-------	-------	-----

Bảng 4.20: Bảng so sánh kết quả mô phỏng mô hình SolidWork khi vẽ 2D giữa các phương pháp điều khiển khác nhau

Nhận xét: Dựa trên kết quả so sánh đánh giá chất lượng hệ thống mô hình cánh tay Robot 3 bậc tự do xây dựng trên SolidWork thực hiện in 3D và vẽ 2D giữa các phương pháp sử dụng bộ điều khiển PID với bộ điều khiển PID kết hợp ILC kiểu D, P và PD được thể hiện trong Bảng 4.19 và Bảng 4.20, ta có thể thấy:

- Đối với phương pháp sử dụng bộ điều khiển PID: Thời gian xác lập t_s gần như giống nhau và rất nhỏ trên cả ba trục (0.01–0.0141 giây), cho thấy phản ứng nhanh, sai số xác lập trung bình e_{tb} và phần trăm sai số xác lập trung bình $\varepsilon\%$ ở mức cao hơn so với các phương pháp còn lại, cho thấy độ chính xác còn hạn chế, Độ quá điều chỉnh POT (%) đặc biệt lớn ở trục Z (trên 3600% khi in 3D), phản ánh sự dao động quá độ mạnh mẽ và nguy cơ gây mất ổn định trong quá trình điều khiển.
- Đối với phương pháp PID kết hợp ILC kiểu D: Có sự cải thiện rõ rệt về độ chính xác, với e_{tb} và $\varepsilon\%$ giảm mạnh ở cả ba trục. Ví dụ, $\varepsilon\%$ ở trục x giảm từ khoảng 0.0898% xuống còn 0.0639% (in 3D) và từ 0.0584% xuống 0.0238% (vẽ 2D). Độ quá điều chỉnh POT (%) cũng được giảm đáng kể, đặc biệt ở trục Z (giảm còn khoảng 1440–1480%), cho thấy khả năng triệt tiêu dao động tốt hơn PID thuần túy. Đây là phương pháp cân bằng tốt giữa thời gian đáp ứng nhanh và cải thiện tính ổn định cho hệ thống.
- Đối với phương pháp PID kết hợp ILC kiểu P: Phương pháp này giúp giảm nhẹ sai số xác lập trung bình e_{tb} và vẫn giữ được thời gian xác lập t_s ngắn tuy không bằng PID. Tuy nhiên, so với ILC-D, hiệu quả giảm dao động POT (%) không cao bằng, đặc biệt vẫn còn dao động lớn ở trục Z (khoảng 1940–2000%). Mặc dù độ chính xác có cải thiện, nhưng khả năng làm giảm độ quá điều chỉnh không mạnh bằng ILC kiểu D hay ILC kiểu PD.
- Đối với phương pháp PID kết hợp ILC kiểu PD: Đây là phương pháp mang lại sự cải thiện đồng đều cả về sai số và độ quá điều chỉnh ở cả ba trục. Đặc biệt, độ quá điều chỉnh ở trục Z khi in 3D được giảm xuống mức thấp (~1650%), cho thấy hiệu quả trong việc khử dao động ban đầu và nâng cao độ ổn định. Mức sai số trung bình cũng ở mức thấp, tương đương hoặc tốt hơn ILC kiểu D và ILC kiểu P. Tổng thể, đây là phương pháp tối ưu nhất trong bốn phương án xét theo tất cả các tiêu chí.

Kết luận: Từ các bảng số liệu 4.19 và 4.20, cho ta thấy việc kết hợp điều khiển học lặp (ILC) với bộ điều khiển PID rõ ràng cải thiện đáng kể chất lượng điều khiển so với PID thuần túy. Trong đó, PID kết hợp ILC kiểu PD cho kết quả tốt nhất về tổng thể (cân bằng

giữa tốc độ, độ chính xác và mức dao động thấp), PID kết hợp ILC kiểu D cũng là một lựa chọn hiệu quả với khả năng giảm sai số và độ quá điều chỉnh khá tốt. Bên cạnh đó, việc xử lý nhiễu sau 2 lần học của các bộ PID kết hợp ILC khá tốt, cho thấy việc tích hợp bộ điều khiển ILC vào giúp hệ thống nâng cao khả năng chống nhiễu một cách hiệu quả.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Trong đồ án tốt nghiệp lần này, nhóm đã tập trung nghiên cứu và triển khai các phương pháp điều khiển cho cánh tay Robot 3 bậc tự do, với mục tiêu đánh giá hiệu quả điều khiển giữa hai phương pháp phổ biến là PID đơn thuần và PID tích hợp thêm điều khiển học lặp (ILC). Thông qua việc xây dựng mô hình động học, động lực học, thiết kế mô hình 3D trên SolidWork và thực hiện mô phỏng trên Matlab/Simulink, đồ án đã hoàn thành đầy đủ các nội dung đặt ra và đạt được những kết quả:

Cụ thể, nhóm đã tiến hành nghiên cứu tổng quan về robot công nghiệp, nhấn mạnh vào cấu trúc, nguyên lý hoạt động và ứng dụng thực tiễn của cánh tay robot 3 bậc tự do. Dựa trên các cơ sở lý thuyết về động học và động lực học, nhóm đã xây dựng mô hình toán học của robot bằng phương pháp Denavit–Hartenberg để mô tả động học thuận và động học nghịch, đồng thời áp dụng phương trình Lagrange để thiết lập mô hình động lực học phản ánh mối quan hệ giữa momen và chuyển động của robot. Bên cạnh đó, mô hình 3D của cánh tay Robot cũng được thiết kế bằng phần mềm SolidWorks nhằm hỗ trợ trực quan hóa cấu trúc cơ khí, trong khi toàn bộ hệ thống điều khiển và mô phỏng hoạt động của robot được triển khai trên nền tảng Matlab/Simulink, qua đó tạo tiền đề vững chắc cho việc thử nghiệm các phương pháp điều khiển trong các chương tiếp theo.

Tiếp nối chương xây dựng mô hình, nhóm đã tiến hành triển khai hai phương pháp điều khiển chính cho cánh tay robot 3 bậc tự do, bao gồm bộ điều khiển PID đơn thuần và bộ PID tích hợp điều khiển học lặp (ILC), với mục tiêu điều khiển robot bám theo quỹ đạo cho trước một cách chính xác và ổn định.

- Đối với bộ điều khiển PID đơn thuần, nhóm đã xây dựng mô hình điều khiển dựa trên nguyên lý hiệu chỉnh sai số theo ba thành phần: tỉ lệ (P), tích phân (I) và đạo hàm (D). Qua mô phỏng trên môi trường Matlab/Simulink, bộ điều khiển PID đơn thuần cho thấy khả năng đáp ứng nhanh, dễ triển khai. Tuy nhiên, trong các tình huống yêu cầu độ chính xác cao qua nhiều lần lặp hoặc hệ thống thay đổi theo thời gian, bộ điều khiển PID bộc lộ nhược điểm như tăng sai số tích lũy và giảm khả năng thích nghi với nhiễu.
- Ngược lại, phương pháp điều khiển bằng bộ PID tích hợp điều khiển học lặp (ILC) được triển khai nhằm tận dụng đặc tính lặp lại của hệ thống để cải thiện độ chính xác theo thời gian. Bộ điều khiển ILC hoạt động bằng cách học từ sai số ở mỗi chu kỳ lặp và hiệu chỉnh điều khiển ở các chu kỳ tiếp theo, giúp hệ thống dần hội tụ đến quỹ đạo mong muốn. Kết quả mô phỏng cho thấy tích hợp

thêm ILC vào bộ điều khiển PID đặc biệt hiệu quả trong các tác vụ lặp lại, làm giảm đáng kể sai số và đảm bảo đầu cuối của cánh tay robot đạt vị trí chính xác hơn sau mỗi lần lặp.

Việc so sánh kết quả mô phỏng giữa hai phương pháp cho thấy rằng bộ điều khiển PID kết hợp ILC có hiệu quả vượt trội hơn so với việc chỉ sử dụng PID đơn thuần. Từ đó, đề án đã đưa ra đánh giá rõ ràng về hiệu quả thực tiễn của phương pháp điều khiển tích hợp, đồng thời góp phần đề xuất hướng áp dụng phù hợp cho các hệ thống robot thực tế có yêu cầu cao về độ chính xác trong các tác vụ lặp lại.

2. Hướng phát triển

Từ những kết quả đã đạt được trong quá trình nghiên cứu và mô phỏng điều khiển cánh tay robot 3 bậc tự do bằng bộ điều khiển PID và kết hợp PID với ILC, nhóm nhận thấy vẫn còn nhiều tiềm năng để mở rộng và nâng cao chất lượng điều khiển trong các nghiên cứu tiếp theo.

Trước hết, việc triển khai mô hình trên phần cứng thực tế là bước cần thiết nhằm kiểm nghiệm độ chính xác và tính khả thi của bộ điều khiển trong điều kiện môi trường thực, có nhiễu và sai số không mô phỏng được. Việc tích hợp cảm biến phản hồi vị trí, tốc độ thực và đo lường sai số quỹ đạo sẽ giúp hiệu chỉnh mô hình một cách chính xác hơn.

Bên cạnh đó, có thể mở rộng mô hình từ cánh tay robot 3 bậc tự do lên các cấu trúc phức tạp hơn như 4 hoặc 6 bậc tự do, từ đó đánh giá khả năng áp dụng của bộ điều khiển PID kết hợp ILC trong các hệ thống robot công nghiệp đa bậc có tính linh hoạt cao hơn. Ngoài ra, việc cải tiến bộ điều khiển ILC theo hướng thích nghi (Adaptive ILC) hoặc kết hợp thêm với các bộ điều khiển hiện đại như LQR, MPC cũng là hướng đi khả thi nhằm tăng khả năng thích ứng và tối ưu hóa hiệu suất điều khiển trong các môi trường biến động.

Cuối cùng, nhóm cũng định hướng phát triển hệ thống điều khiển thông minh hơn bằng cách kết hợp thuật toán học máy (Machine Learning) để tối ưu hóa thông số điều khiển PID – ILC dựa trên dữ liệu thu thập được từ quá trình vận hành thực tế, góp phần nâng cao độ chính xác và hiệu suất hệ thống một cách tự động.

TÀI LIỆU THAM KHẢO

- [1] ROBOT công nghiệp - TS. Phạm Đăng Phước - NXB Xây dựng Hà Nội (2007)
- [2] Giáo trình KỸ THUẬT ROBOT – PGS. TS. Nguyễn Trường Thịnh – NXB Đại học Quốc gia TP. Hồ Chí Minh
- [3] ĐIỀU KHIỂN HỌC LẬP - Nguyễn Thu Hà, Nguyễn Hoài Nam, Nguyễn Doãn Phước - NXB Bách Khoa Hà Nội
- [4] Robotics Modelling, Planning and Control – Chapter 7 - Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo
- [5] book 2004 - Robot Modeling and Control - Spong, Hutchinson & Vidyasagar
- [6] Nonlinear Control Feedback Linearization Sliding Mode Control
- [7] PID Controllers: Theory, Design, and Tuning - Åström & Hägglund
- [8] <https://slunik.slu.se/kursfiler/TE0010/10095.1213/Reg1TuneReview.pdf>
- [9] Lý thuyết điều khiển tự động – TS. Nguyễn Chí Ngôn, ThS. Nguyễn Hoàng Dũng – Đại Học Cần Thơ (2012)
- [10] [Cánh tay Robot công nghiệp của ZEKEEP mã số ZKBOT-3042B](#)

PHỤ LỤC

Code đầu vào cho cánh tay Robot:

Hình vẽ 2D:

```
function [x, y, z] = ngoi_sao(t)
n = 5;
R = 10;
r = R * sin(pi / (2 * n)) / cos(pi / n);
x_center = 20;
y_center = 0;
z_center = 0;
T = 10;
theta = (0:2*n-1) * pi / n + pi/2;
radii = repmat([R, r], 1, n);
x_vertices = x_center + radii .* cos(theta);
y_vertices = y_center + radii .* sin(theta);
segment_time = T / (2 * n);
index_float = mod(t / segment_time, 2 * n) + 1;
index = floor(index_float);
alpha = index_float - index;
x = (1 - alpha) * x_vertices(index) + alpha * x_vertices(mod(index, 2 * n)
+ 1);
y = (1 - alpha) * y_vertices(index) + alpha * y_vertices(mod(index, 2 * n)
+ 1);
z = z_center;
end
```

Hình in 3D:

```
function [x, y, z] = hình_cau(t)
R = 5;
n_turns = 10;
T_total = 10;
x0 = 20;
y0 = 5.5;
z0 = 5.5;
if t < 0
    t = 0;
elseif t > T_total
    t = T_total;
```

```

end
s = t / T_total;
phi = pi * (1 - s);
theta = 2 * pi * n_turns * s;
x = x0 + R * sin(phi) * cos(theta);
y = y0 + R * sin(phi) * sin(theta);
z = z0 + R * cos(phi);

```

Hình 3D trên Matlab:

```

R = 5; % Bán kính hình cầu
n_turns = 10; % Số vòng xoắn
T_total = 10; % Tổng thời gian
t = linspace(0, T_total, 5000); % Dải thời gian
% Tâm mới của hình cầu
x0 = 20;
y0 = 5.5;
z0 = 5.5;
% Khởi tạo mảng tọa độ
x = zeros(size(t));
y = zeros(size(t));
z = zeros(size(t));
% Tính tọa độ quỹ đạo trên mặt cầu
for i = 1:length(t)
    s = t(i) / T_total;
    phi = pi * s; % Từ cực Bắc đến cực Nam
    theta = 2 * pi * n_turns * s;
    x(i) = x0 + R * sin(phi) * cos(theta);
    y(i) = y0 + R * sin(phi) * sin(theta);
    z(i) = z0 + R * cos(phi);
end
% Vẽ quỹ đạo
plot3(x, y, z, 'b', 'LineWidth', 1.5);
axis equal;
grid on;
xlabel('X'); ylabel('Y'); zlabel('Z');
title('Quỹ đạo xoắn ốc trên mặt cầu với tâm (20, 0, 5)');

```

Động học nghịch:

```

function [q1r, q2r, q3r] = DongHocNghich(xd, yd, zd)
l1 = 13.7;
l2 = 20;

```

```

l3 = 23;
%tinh theta 1
q1r = atan2(yd,xd);
%tinh theta 3
E = xd*cos(q1r) + yd*sin(q1r);
F = E^2 + zd^2 - 2*zd*l1 + l1^2 - l2^2 - l3^2;
c3 = F/(2*l2*l3);
c3 = max(min(c3, 1), -1);
s3 = sqrt(1 - c3^2);
q3r = atan2(s3,c3);
%tinh theta 2
TS_c2 = E*(l2 + l3*c3) + l3*s3*(l1 - zd);
TS_s2 = (l1 - zd)*(l2 + l3*c3) - E*l3*s3;
MS = (l2 + l3*c3)^2 + (l3^2)*(s3^2);
c2 = TS_c2/MS;
s2 = TS_s2/MS;
q2r = atan2(s2,c2);
end

```

Động học thuận:

```

function [x_EE, y_EE, z_EE] = DongHocThuan(q1d, q2d, q3d)
l1 = 13.7;
l2 = 20;
l3 = 23;
x_EE = cos(q1d)*(l2*cos(q2d) + l3*cos(q2d + q3d));
y_EE = sin(q1d)*(l2*cos(q2d) + l3*cos(q2d + q3d));
z_EE = l1 - l2*sin(q2d) - l3*sin(q2d + q3d);
end

```

Phương trình Lagrange động lực học cánh tay Robot 3DOF:

```

function [T1,T2,T3] = fcn(q2,q3,q1d,q2d,q3d,q1dd,q2dd,q3dd)
%Dinh nghĩa gia tri
l1 = 13.7;
l2 = 20;
l3 = 23;
R = 9.25;
g = 9.8;
m1 = 3;
m2 = 2;
m3 = 2;

```

```

m11 = 0.5*m1*R;
m22 = ((m2/3)+m3)*(l2^2) + (1/3)*m3*(l3^2) + m3*l2*l3*cos(q3);
m23 = (1/3)*m3*(l3^2) + 0.5*m3*l2*l3*cos(q3);
m32 = m23;
m33 = (1/3)*m3*(l3^2);
M = [m11 0 0; 0 m22 m23; 0 m32 m33];

c11 = 0;
c12 = 0;
c13 = 0;
c21 = 0;
c22 = -m3*l2*l3*q3d*sin(q3);
c23 = -m3*l2*l3*q2d*sin(q3) - 0.5*m3*l2*l3*q3d*sin(q3);
c31 = 0;
c32 = 0.5*m3*l2*l3*q2d*sin(q3);
c33 = 0;
C = [c11 c12 c13; c21 c22 c23; c31 c32 c33];

g1 = 0;
g2 = (0.5*m2+m3)*g*l2*cos(q2) + 0.5*m3*g*l3*cos(q2+q3);
g3 = 0.5*m3*g*l3*cos(q2+q3);
G = [g1;g2;g3];

dq = [q1d;q2d;q3d];
ddq = [q1dd;q2dd;q3dd];
T = (M*ddq) + (C*dq) + G;
T1 = T(1);
T2 = T(2);
T3 = T(3);
end

```

Code bộ điều khiển ILC:

```

clc;
clear;
Ts = 0.001;
time = (0:Ts:10)';

q3_noise_value = zeros(size(time));
q3_noise_value(time >= 4 & time <= 4.1) = 0.25;
q3_noise = [time, q3_noise_value]; %nhieu

```

```

e1k_t = [time zeros(length(time),1)];
e1k_t1 = [time zeros(length(time),1)]; % e_k(t-1)
u1k_t = [time zeros(length(time),1)];

e2k_t = e1k_t;
e2k_t1 = e1k_t;
u2k_t = u1k_t;

e3k_t = e1k_t;
e3k_t1 = e1k_t;
u3k_t = u1k_t;

N = 1;
n = 15;

for i = 1:n
    if i == 1
        nhieu = q3_noise;
    else
        nhieu = [time, zeros(length(time),1)];
    end
    sim('DATN_Robot3DOF_ILC_PD_DLH');
    % Joint 1
    e1k_t1(2:end,2) = e1k_t(1:end-1,2);
    e1k_t1(end,2) = e1k_t(end-1,2);
    e1k_t = e1k1_t;
    u1k_t = u1k1_t;
    % Joint 2
    e2k_t1(2:end,2) = e2k_t(1:end-1,2);
    e2k_t1(end,2) = e2k_t(end-1,2);
    e2k_t = e2k1_t;
    u2k_t = u2k1_t;
    % Joint 3
    e3k_t1(2:end,2) = e3k_t(1:end-1,2);
    e3k_t1(end,2) = e3k_t(end-1,2);
    e3k_t = e3k1_t;
    u3k_t = u3k1_t;
    N = N + 1;
end

```

Code tính toán các chỉ tiêu đánh giá chất lượng hệ thống:

```

t = tout;
e1 = abs(xd - xr);
e2 = abs(yd - yr);
e3 = abs(zd - zr);

%Thoi gian xac lap (2% sai so xac lap)
A_up_x = 1.02*abs(xd);
A_down_x = 0.98*abs(xd);
n1 = length(xr);
range1 = (abs(xr) >= A_down_x) & (abs(xr) <= A_up_x);
for m = 1:n1
    if all(range1(m:end))
        t_xl_x = t(m);
        break;
    end
end

A_up_y = 1.02*abs(yd);
A_down_y = 0.98*abs(yd);
n2 = length(yr);
range2 = (abs(yr) >= A_down_y) & (abs(yr) <= A_up_y);
for j = 1:n2
    if all(range2(j:end))
        t_xl_y = t(j);
        break;
    end
end

A_up_z = 1.02*abs(zd);
A_down_z = 0.98*abs(zd);
n3 = length(zr);
range3 = (abs(zr) >= A_down_z) & (abs(zr) <= A_up_z);
for l = 1:n3
    if all(range3(l:end))
        t_xl_z = t(l);
        break;
    end
end

%Tinh sai so trung binh
t_tinhtoan_x = t >= t_xl_x;
e_tb_x = mean(e1(t_tinhtoan_x));

```

```

xd_tb = mean(xd(t_tinhthoan_x));
pt_ex_tb = (abs(e_tb_x/xd_tb))*100;

t_tinhthoan_y = t >= t_xl_y;
e_tb_y = mean(e2(t_tinhthoan_y));
yd_tb = mean(yd(t_tinhthoan_y));
pt_ey_tb = (abs(e_tb_y/yd_tb))*100;

t_tinhthoan_z = t >= t_xl_z;
e_tb_z = mean(e3(t_tinhthoan_z));
zd_tb = mean(zd(t_tinhthoan_z));
pt_ez_tb = (abs(e_tb_z/zd_tb))*100;

%Ket qua
fprintf('=== ĐÁNH GIÁ CHẤT LƯỢNG HỆ THỐNG ===\n');
fprintf('Thời gian xác lập x: %.4f giây\n', t_xl_x);
fprintf('Thời gian xác lập y: %.4f giây\n', t_xl_y);
fprintf('Thời gian xác lập z: %.4f giây\n', t_xl_z);
fprintf('Sai số xác lập trung bình x: %.4f\n', e_tb_x);
fprintf('Sai số xác lập trung bình y: %.4f\n', e_tb_y);
fprintf('Sai số xác lập trung bình z: %.6f\n', e_tb_z);
fprintf('Phần trăm sai số xác lập trung bình x: %.4f%%\n', pt_ex_tb);
fprintf('Phần trăm sai số xác lập trung bình y: %.4f%%\n', pt_ey_tb);
fprintf('Phần trăm sai số xác lập trung bình z: %.4f%%\n', pt_ez_tb);

```