

ĐẠI HỌC BÁCH KHOA HÀ NỘI



ĐỒ ÁN TỐT NGHIỆP

Điều khiển chống rung cho cầu trục sử dụng bộ Input Shaping cải tiến

HỒ VĂN QUANG

quang.hv181711@sis.hust.edu.vn

Ngành Kỹ thuật điều khiển và tự động hóa

Giảng viên hướng dẫn:

TS. Đỗ Trọng Hiếu

Chữ ký của GVHD

Khoa:

Tự động hóa

Trường :

Điện- Điện tử

HÀ NỘI, 8/2023

NHIỆM VỤ
ĐỒ ÁN TỐT NGHIỆP

Họ và tên sinh viên: Hồ Văn Quang

Khóa:K63 Trường: Điện- Điện tử

Ngành: KT ĐK &TĐH

1. Tên đề tài: *Điều khiển chống rung cho cầu trục sử dụng bộ Input Shaping cải tiến*

2. Nội dung đề tài:

- Tìm hiểu về hệ thống cầu trục và bài toán chống rung cho hệ thống cầu trục.
- Tìm hiểu về bộ điều khiển ADRC và bộ lọc Input Shaping phục vụ cho điều khiển và chống rung cầu trục.
- Xây dựng mô hình toán học hệ thống cầu trục từ đó ứng dụng bộ điều khiển ADRC kết hợp với bộ lọc Input Shaping ZVD được cải tiến nhờ ứng dụng mạng Neural. Sử dụng thuật toán tối ưu hóa bầy đàn Particle Swarm Optimization (PSO) tiến hành tạo mẫu và huấn luyện cho bộ Input shaping.
- Tìm hiểu mô hình cầu trục tại phòng thí nghiệm và thiết kế điều khiển, kiểm nghiệm tính đúng đắn của bộ điều khiển và bộ Input Shaping cải tiến.

3. Thời gian giao đề tài:04/2023

4. Thời gian hoàn thành:08/2023

Ngày..... tháng năm 2023

CÁN BỘ HƯỚNG DẪN

Lời cảm ơn

Trong quá trình thực hiện đề tài "Điều khiển chống rung cho cầu trục sử dụng bộ Input Shaping cải tiến", em muốn gửi lời cảm ơn chân thành đến thầy giáo TS. Đỗ Trọng Hiếu, người đã tận tình hướng dẫn và đồng hành cùng em trong suốt quá trình nghiên cứu. Nhờ những chỉ dẫn và kiến thức quý báu từ thầy, em đã vận dụng thành công những kiến thức đã học để hoàn thành đề tài này.

Em cũng muốn bày tỏ lòng biết ơn đến các thầy cô trong khoa Tự động hóa, những người đã chia sẻ những kiến thức bổ ích và truyền đạt cho em những kỹ năng cần thiết để thực hiện đề tài này. Sự hỗ trợ từ bạn bè cũng không thể thiếu trong quá trình hoàn thành đồ án tốt nghiệp.

Em xin chân thành cảm ơn

Tóm tắt nội dung đồ án

Trong nội dung phần báo cáo này sẽ trình bày những nội dung lý thuyết cũng như các kết quả mô phỏng và thực nghiệm khi thực hiện nghiên cứu đề tài “Điều khiển chống rung cho cầu trục sử dụng input shaping cải tiến”. Nội dung báo cáo gồm 6 chương như sau:

- Chương 1: Giới thiệu chung về hệ thống cầu trục, sau đó đặt ra yêu cầu điều khiển chống rung lắc cho cầu trục, đưa ra giải pháp tối ưu điều khiển.
- Chương 2: Xây dựng mô hình toán học cho hệ thống cầu trục con lắc đơn, thiết kế bộ điều khiển ADRC điều khiển vị trí và chiều dài dây, giới thiệu thiết kế bộ Input Shaping truyền thống.
- Chương 3: Xây dựng mạng Neural dựa trên cơ sở thuật toán Input Shaping. Sau đó triển khai và kiểm chứng trên mô hình Matlab sử dụng bộ điều khiển đã xây dựng ở chương 2 và so sánh các bộ Input Shaping truyền thống.
- Chương 4: Giới thiệu tổng quan về hệ thống cầu trục thực nghiệm được dùng trong đồ án này. Thông qua nhận dạng hàm truyền và tính toán bộ điều khiển ADRC và bộ lọc Input Shaping, kiểm chứng kết quả trên mô hình thực nghiệm.
- Chương 5: Đưa ra kết luận đánh giá độ hiệu quả của bộ Input Shaping cải tiến. Đưa ra những thiếu sót khi thực hiện đồ án và phương án cải thiện.

MỤC LỤC

| | |
|--|-----------|
| CHƯƠNG 1. GIỚI THIỆU CHUNG | 1 |
| 1.1 Hệ thống cầu trục | 1 |
| 1.1.1 Giới thiệu chung về hệ thống cầu trục | 1 |
| 1.1.2 Ứng dụng..... | 3 |
| 1.2 Vấn đề chống rung cho cầu trục | 3 |
| 1.2.1 Yêu cầu điều khiển..... | 3 |
| 1.2.2 Đề xuất thuật toán điều khiển | 3 |
| 1.2.3 Nội dung đồ án..... | 4 |
| CHƯƠNG 2. XÂY DỰNG MÔ HÌNH MÔ PHỎNG | 5 |
| 2.1 Mô hình hóa hệ thống cầu trục | 5 |
| 2.1.1 Mô hình toán học hệ thống cầu trục..... | 5 |
| 2.1.2 Kiểm chứng mô hình toán học | 7 |
| 2.1.3 Triển khai và kiểm chứng mô hình trên Matlab Simulink..... | 7 |
| 2.2 Thiết kế bộ điều khiển ADRC cho mô hình mô phỏng | 12 |
| 2.2.1 Bộ điều khiển ADRC | 12 |
| 2.2.2 Thiết kế bộ điều khiển ADRC cho mô hình mô phỏng | 15 |
| 2.3 Giới thiệu bộ lọc Input Shaping..... | 15 |
| 2.3.1 Thuật toán định hình đầu vào (Input shaping) | 16 |
| 2.3.2 Thiết kế bộ lọc Input Shaping | 16 |
| CHƯƠNG 3. TRIỂN KHAI BỘ LỌC INPUT SHAPING CẢI TIẾN TRÊN MÔ HÌNH MÔ PHỎNG..... | 20 |
| 3.1 Mạng neural | 20 |
| 3.1.1 Tổng quan về mạng Neural..... | 20 |
| 3.1.2 Lựa chọn cấu trúc mạng neural..... | 20 |
| 3.1.3 Các bước xây dựng mạng Feedforward Neural Network | 21 |
| 3.2 Thuật toán tối ưu hóa bầy đàn (PSO) | 22 |
| 3.2.1 Tổng quan thuật toán tối ưu hóa bầy đàn..... | 22 |
| 3.2.2 Giải thuật thuật toán PSO..... | 23 |
| 3.3 Thiết kế mạng neural | 25 |
| 3.3.1 Xác định kiến trúc mạng | 25 |
| 3.3.2 Lựa chọn hàm kích hoạt..... | 25 |
| 3.3.3 Chuẩn bị dữ liệu | 31 |

| | | |
|--|---|-----------|
| 3.3.4 | Tiến hành huấn luyện..... | 33 |
| 3.3.5 | Đánh giá sử dụng mô hình..... | 35 |
| 3.4 | Triển khai mô hình mạng Neural trên Matlab Simulink | 35 |
| 3.4.1 | TH1 nâng tải..... | 37 |
| 3.4.2 | TH2 hạ tải..... | 40 |
| 3.4.3 | TH3 nâng hạ tải tránh vật cản | 42 |
| CHƯƠNG 4. TRIỂN KHAI TRÊN MÔ HÌNH THỰC NGHIỆM | | 45 |
| 4.1 | Giới thiệu mô hình cầu trục | 45 |
| 4.1.1 | Mô hình cầu trục | 45 |
| 4.1.2 | Tủ điện | 46 |
| 4.1.3 | Bộ điều khiển NI-myRIO 1900..... | 48 |
| 4.1.4 | Phần mềm LabVIEW | 50 |
| 4.2 | Nhận dạng hàm truyền mô hình..... | 51 |
| 4.3 | Thiết kế bộ điều khiển và bộ lọc cho mô hình thực nghiệm..... | 54 |
| 4.3.1 | Thiết kế bộ điều khiển..... | 54 |
| 4.3.2 | Thiết kế bộ lọc..... | 54 |
| 4.3.3 | Kết quả chạy trên mô hình thực nghiệm | 56 |
| KẾT LUẬN..... | | 63 |
| TÀI LIỆU THAM KHẢO | | 64 |

DANH MỤC HÌNH VẼ

| | |
|--|----|
| Hình 1.1 Hệ thống cầu trục | 1 |
| Hình 1.2 Cấu tạo hệ thống cầu trục..... | 2 |
| Hình 2.1 Mô hình cầu trục 2D | 5 |
| Hình 2.2 Mô hình cầu trục trên Matlab Simulink | 7 |
| Hình 2.3 Giá trị đặt vào trục x theo TH1 | 8 |
| Hình 2.4 Đáp ứng vị trí khi x theo TH1 | 8 |
| Hình 2.5 Đáp ứng chiều dài dây x theo TH1 | 8 |
| Hình 2.6 Đáp ứng góc giao động x theo TH1 | 9 |
| Hình 2.7 Giá trị đặt vào trục z theo TH2 (Tác động $F_z=5N$ theo phương z)..... | 9 |
| Hình 2.8 Đáp ứng vị trí theo TH2 | 10 |
| Hình 2.9 Đáp ứng chiều dài dây theo TH2 | 10 |
| Hình 2.10 Đáp ứng góc lắc theo TH2 | 10 |
| Hình 2.11 Giá trị đặt vào trục x theo TH3 (Tác động lực vào cả hai phương).... | 11 |
| Hình 2.12 Đáp ứng vị trí theo TH3 | 11 |
| Hình 2.13 Giá trị đặt vào trục z theo TH3 | 11 |
| Hình 2.14 Đáp ứng chiều dài dây theo trường hợp 3..... | 12 |
| Hình 2.15 Đáp ứng góc lắc theo TH3 | 12 |
| Hình 2.16 Cấu trúc bộ điều khiển ADRC | 15 |
| Hình 2.17 Cấu trúc điều khiển thuật toán Input Shaping..... | 16 |
| Hình 2.18 Đáp ứng của hệ bậc hai với vector xung..... | 17 |
| Hình 2.19 Vị trí của Vector xung ZV Shaper | 17 |
| Hình 2.20 Đáp ứng của hệ với các xung riêng lẻ và đáp ứng với chuỗi xung..... | 18 |
| Hình 3.1 Lưu đồ thuật toán PSO | 23 |
| Hình 3.2 Kiến trúc mạng neural lựa chọn..... | 25 |
| Hình 3.3 Hàm sigmod | 26 |
| Hình 3.4 Đạo hàm của hàm sigmod | 26 |
| Hình 3.5 Hàm tank | 28 |
| Hình 3.6 Đạo hàm của hàm tanh..... | 28 |
| Hình 3.7 Hàm Swish | 30 |
| Hình 3.8 Mô hình huấn luyện dữ liệu | 32 |
| Hình 3.9 Triển khai mô hình trên Simulink | 36 |
| Hình 3.10 Mô hình mạng Neural trên Matlab Simulink..... | 37 |
| Hình 3.11 Đồ thị tín hiệu điều khiển trục x TH1 | 37 |
| Hình 3.12 Đồ thị đáp ứng vị trí TH1..... | 38 |
| Hình 3.13 Đồ thị tín hiệu điều khiển trục z TH1 | 38 |
| Hình 3.14 Đồ thị đáp ứng chiều dài dây TH1 | 38 |

| | |
|---|----|
| Hình 3.15 Đồ thị đáp ứng góc lệch TH1 | 39 |
| Hình 3.16 Đồ thị tín hiệu điều khiển vị trí TH2..... | 40 |
| Hình 3.17 Đồ thị đáp ứng vị trí TH2..... | 40 |
| Hình 3.18 Đồ thị tín hiệu điều khiển chiều dài dây TH2 | 40 |
| Hình 3.19 Đồ thị đáp ứng chiều dài dây TH2 | 41 |
| Hình 3.20 Đồ thị đáp ứng góc lắc TH2..... | 41 |
| Hình 3.21 Đồ thị tín hiệu điều khiển trục x | 42 |
| Hình 3.22 Đồ thị đáp ứng vị trí TH3..... | 42 |
| Hình 3.23 Đồ thị tín hiệu điều khiển trục z..... | 43 |
| Hình 3.24 Đồ thị đáp ứng chiều dài dây TH3 | 43 |
| Hình 3.25 Đồ thị đáp ứng góc lắc TH3..... | 44 |
| Hình 4.1 Mô hình cầu trục thực nghiệm | 45 |
| Hình 4.2 Mặt ngoài của tủ điện..... | 46 |
| Hình 4.3 Mặt trong của tủ điện | 47 |
| Hình 4.4 Bộ điều khiển NI-myRIO 1900..... | 48 |
| Hình 4.5 Cấu hình phần cứng của NI-myRIO 1900 | 48 |
| Hình 4.6 Sơ đồ I/O trên cổng mở rộng MXP..... | 49 |
| Hình 4.7 Sơ đồ I/O trên cổng mở rộng MSP | 50 |
| Hình 4.8 Giao diện lập trình trên LabVIEW..... | 51 |
| Hình 4.9 Chương trình nhận dạng hàm truyền động cơ..... | 52 |
| Hình 4.10 Giao diện phần mềm nhận dạng hàm truyền động cơ..... | 52 |
| Hình 4.11 Đồ thị tốc độ động cơ trục x theo điện áp..... | 53 |
| Hình 4.12 Đồ thị tốc độ động cơ trục Z theo điện áp..... | 53 |
| Hình 4.13: Phương pháp tìm kiếm dữ liệu trong thực nghiệm [3] | 55 |
| Hình 4.14 Bộ lọc Input Shaping cải tiến trên phần mềm Labview..... | 57 |
| Hình 4.15 Điện áp điều khiển trục x (TH1 nâng tải) | 57 |
| Hình 4.16 Đáp ứng vị trí trục x TH1..... | 58 |
| Hình 4.17 Điện áp điều khiển trục z TH1 | 58 |
| Hình 4.18 Đáp ứng chiều dài dây trục z TH1 | 58 |
| Hình 4.19 Đáp ứng góc lắc trường hợp nâng tải TH1 | 59 |
| Hình 4.20 Điện áp điều khiển trục x TH2 (Hạ tải) | 60 |
| Hình 4.21 Đáp ứng vị trí trục x TH2..... | 60 |
| Hình 4.22 Điện áp điều khiển trục z TH2 | 60 |
| Hình 4.23 Đáp ứng chiều dài dây trục z TH2 | 61 |
| Hình 4.24 Đáp ứng góc lắc khi hạ tải TH2 | 61 |

DANH MỤC BẢNG

| | |
|---|----|
| Bảng 2.1 Thông số mô hình toán học cầu trục | 7 |
| Bảng 3.1 Dữ liệu huấn luyện được | 33 |
| Bảng 3.2 Trọng số và độ lệch huấn luyện được..... | 35 |
| Bảng 3.3 Đánh giá sử dụng mô hình..... | 35 |
| Bảng 3.4 Thông số bộ lọc Input Shaping..... | 36 |
| Bảng 3.5 Đáp ứng góc lắc ứng với các thuật toán điều khiển trong TH1. | 39 |
| Bảng 3.6 Đáp ứng góc lệch trong TH2 | 41 |
| Bảng 4.1 Dữ liệu huấn luyện thực nghiệm | 56 |
| Bảng 4.2 Trọng số cho bộ Input Shaping cải tiến thực nghiệm..... | 56 |
| Bảng 4.3 Đáp ứng góc lắc TH1..... | 59 |
| Bảng 4.4 Đáp ứng góc lắc TH2..... | 62 |

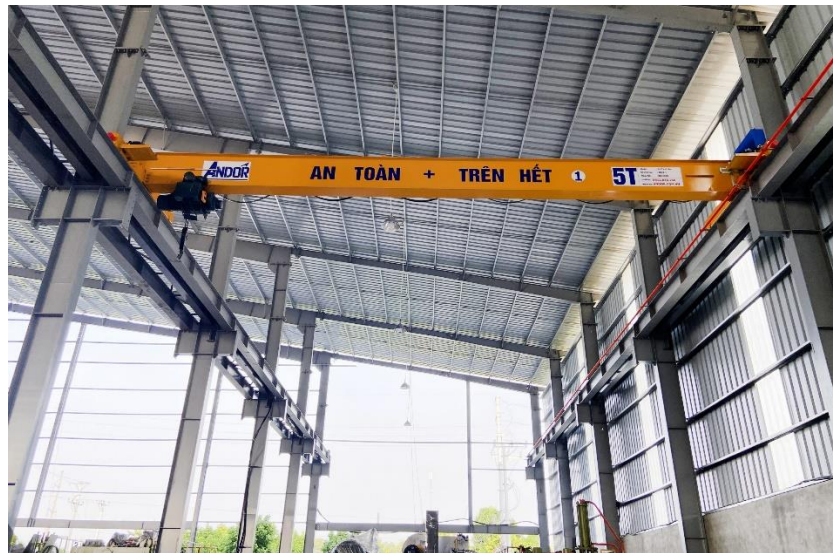
CHƯƠNG 1. GIỚI THIỆU CHUNG

Chương 1 giới thiệu chung về hệ thống cầu trục bao gồm nguyên lý hoạt động, phân loại, ứng dụng. Sau đó đặt ra yêu cầu điều khiển chống rung lắc cho cầu trục, đưa ra giải pháp tối ưu hơn so với các thuật toán điều khiển đã được nghiên cứu ở các đề án trước

1.1 Hệ thống cầu trục

1.1.1 Giới thiệu chung về hệ thống cầu trục

Khái niệm: Cầu trục là một thiết bị được sử dụng để nâng, hạ và di chuyển hàng hóa trong các nhà xưởng. Với sức nâng từ 1 đến 500 tấn và hoạt động chủ yếu bằng động cơ điện, cầu trục rất phổ biến trong các nhà máy sản xuất công nghiệp. Việc sử dụng cầu trục giúp tăng hiệu suất làm việc, nâng cao năng suất lao động.



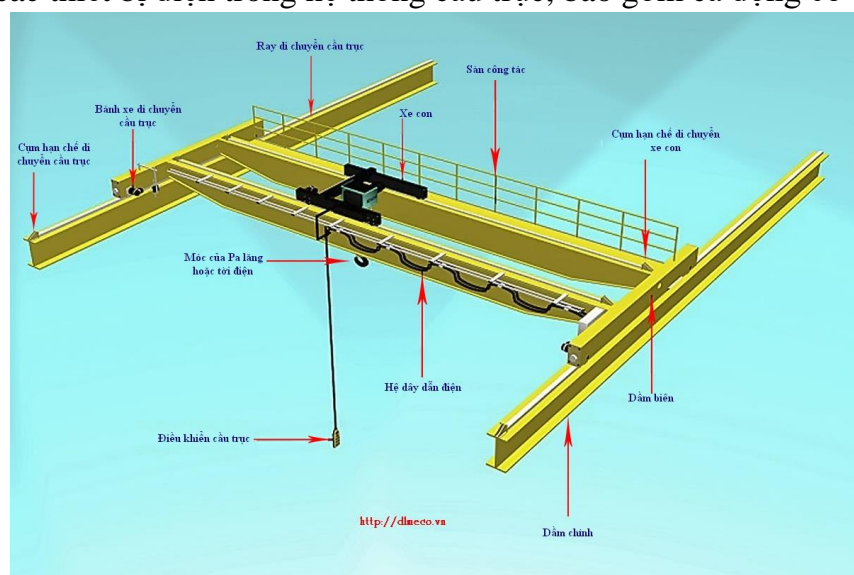
Hình 1.1 Hệ thống cầu trục

Cấu tạo: Cầu trục có cấu trúc dạng cầu thép, trên đó lắp bộ phận di chuyển sử dụng bánh xe lăn trên đường ray đặc biệt được lắp đặt trên tường hoặc dầm của nhà xưởng, do đó còn được gọi là cầu lăn. Dựa vào cấu trúc thép của cầu trục, ta có thể chia cầu trục thành các loại như: cầu trục một dầm và cầu trục dầm đôi, cầu trục treo, cầu trục monorail, cầu trục quay...

Cấu trúc của cầu trục bao gồm: Cầu trục là một hệ thống có cấu trúc dạng cầu thép, được di chuyển bằng bánh xe lăn trên đường ray. Hãy xem xét từng thành phần của hệ thống cầu trục:

- Dầm chính: Đây là thành phần chịu lực chính của cầu trục, có thiết kế dạng hộp hoặc thép chữ I. Dầm chính chịu trách nhiệm chịu tải trọng của cầu trục và đảm bảo độ cứng và độ đàn hồi cần thiết.
- Dầm đầu: Đây là cấu trúc thép hình hộp chữ nhật nằm ở hai đầu dầm chính. Dầm đầu được lắp đặt các bộ phận di chuyển và hệ thống giảm chấn bằng cao su, giúp giảm va chạm khi cầu trục di chuyển đến vị trí dừng cuối trên đường ray.

- Bánh xe cầu trục: Đây là bộ phận di chuyển của cầu trục. Bánh xe lắp trên trục truyền động và được di chuyển trên đường ray, đẩy dầm chính di chuyển trong không gian làm việc.
- Cột nhà xưởng, dầm chạy: Đây là các cấu trúc hỗ trợ và định vị cầu trục trong nhà xưởng. Cột nhà xưởng được sử dụng để chống đỡ và cung cấp vị trí cố định cho hệ thống cầu trục. Dầm chạy là đường ray đặc biệt được lắp đặt trên cột nhà xưởng, trên đó bánh xe cầu trục di chuyển.
- Đường ray đặc biệt: Đường ray là một thành phần quan trọng trong hệ thống cầu trục. Nó được lắp đặt trên cột nhà xưởng hoặc dầm chạy và đóng vai trò quyết định trong việc hướng dẫn và hỗ trợ bánh xe cầu trục di chuyển một cách chính xác và ổn định.
- Hệ thống giảm chấn: Hệ thống giảm chấn được sử dụng để giảm va chạm và dao động trong quá trình di chuyển của cầu trục. Thông qua sử dụng các bộ phận giảm chấn như cao su, hệ thống này giúp làm giảm tác động và đảm bảo hoạt động ổn định của cầu trục.
- Động cơ di chuyển: Động cơ điện được sử dụng để tạo chuyển động cho bánh xe cầu trục. Năng lượng từ động cơ được truyền qua hệ thống truyền động để di chuyển cầu trục theo yêu cầu.
- Động cơ di chuyển xe con: Động cơ di chuyển xe con là thành phần chịu trách nhiệm cho việc di chuyển xe con trên đường ray của cầu trục. Xe con là phần nâng hạ và di chuyển hàng hóa.
- Phần nâng hạ: Phần này bao gồm các thiết bị như palang cáp điện, palang xích điện hoặc xe con. Chức năng chính của phần nâng hạ là nâng và hạ hàng hóa theo yêu cầu.
- Hệ thống điều khiển cầu trục: Hệ thống này đảm bảo quá trình điều khiển và điều chỉnh hoạt động của cầu trục. Nó có thể được điều khiển từ cabin, tay điều khiển nối trực tiếp với cầu trục hoặc từ xa.
- Hệ thống dẫn điện: Hệ thống dẫn điện cung cấp nguồn điện và dẫn dòng điện đến các thiết bị điện trong hệ thống cầu trục, bao gồm cả động cơ



Hình 1.2 Cấu tạo hệ thống cầu trục

Nguyên lí hoạt động: Cầu trục hoạt động bằng cách sử dụng động cơ điện truyền chuyển động qua trục truyền động và khớp nối tới các hộp giảm tốc. Sau đó, chuyển động được truyền cho bánh xe cầu trục, làm di chuyển dầm chính được gắn trên các dầm đầu. Xe con, có cơ cấu nâng, di chuyển trên ray gắn trên dầm chính. Hệ thống phanh được sử dụng để hãm lại cầu trục khi cần thiết. Các động cơ điện được điều khiển bằng hệ thống điều khiển được đặt trong cabin.

1.1.2 Ứng dụng

Cầu trục được sử dụng rộng rãi trong nhiều ứng dụng công nghiệp khác nhau, bao gồm:

- Lắp ráp và ghép sản phẩm: Cầu trục được dùng để di chuyển các bộ phận và bán thành phẩm trong quá trình lắp ráp và ghép sản phẩm trên dây chuyền sản xuất.
- Vận chuyển: Cầu trục được sử dụng để nâng hạ các sản phẩm thành phẩm từ dây chuyền sản xuất lên các phương tiện vận chuyển như xe tải hoặc container.
- Nâng và giữ chi tiết: Cầu trục được dùng để nâng và giữ các chi tiết trong quá trình sản xuất, giúp tăng hiệu suất và tiện lợi trong công việc.
- Quản lý kho bãi: Cầu trục cũng được sử dụng để vận chuyển hàng hóa nặng trong quá trình nhập và xuất kho, giúp tăng năng suất và hiệu quả trong hoạt động quản lý kho.
- Những ứng dụng trên chỉ là một số ví dụ điển hình, cầu trục còn có thể được sử dụng trong nhiều ngành công nghiệp khác như xây dựng, khai thác mỏ, công trình cảng biển và nhiều lĩnh vực khác.

1.2 Vấn đề chống rung cho cầu trục

1.2.1 Yêu cầu điều khiển

Trong hệ thống cầu trục di chuyển từ vị trí A đến vị trí B theo trục X, khi cung cấp một lực F theo phương X để xe con di chuyển, hệ hàng hóa treo bên dưới có khối lượng lớn sẽ bị dao động và tạo ra những dao động rung lắc. Tình trạng dao động này gây nguy hiểm cho con người và hàng hóa. Vì vậy, để đảm bảo an toàn cho hệ thống, chúng ta cần hạn chế góc dao động xuống mức thấp nhất có thể. Yêu cầu điều khiển đặt ra:

- Điều khiển chính xác vị trí: Đảm bảo vật được di chuyển đến nơi yêu cầu.
- Điều khiển chống rung lắc : Đảm bảo trên quãng đường dịch chuyển, hạn chế giao động xuống thấp nhất có thể

Phương pháp thông thường và đơn giản nhất để hạn chế góc dao động là làm cho hệ thống di chuyển chậm. Tuy nhiên, phương pháp này lại không hiệu quả về mặt hiệu suất. Do đó, chúng ta cần tìm những phương án khác, mang tính hiệu quả hơn, để giải quyết vấn đề chống rung.

1.2.2 Đề xuất thuật toán điều khiển

Một trong những thuật toán mạnh mẽ được áp dụng rộng rãi cho điều khiển chống rung cầu trục đó là thuật toán Input Shaping. Input shaping là một thuật toán

được sử dụng để giảm thiểu dao động và rung lắc trong hệ thống cơ học hoặc các hệ thống điều khiển. Nó được áp dụng để điều chỉnh tín hiệu đầu vào (input signal) để tạo ra một tín hiệu đầu vào tối ưu, giúp giảm thiểu dao động và rung lắc trong hệ thống. Một số bộ Input Shaping đã được sử dụng trong các bài báo cáo trước: ZV, ZVD.

Phân tích nhược điểm các bộ Input Shaping thông thường:

- Dễ bị ảnh hưởng bởi các yếu tố : Chiều dài dây, khối lượng tải, ma sát ...
- Khi tính toán bộ Input Shaping cần biết rõ các thông số mô hình hệ thống

Đề xuất phương pháp điều khiển cải tiến: Ứng dụng của mạng Neural vào bộ định hình Input Shaping giảm thiểu giao động của hệ thống cầu trục.

Ưu điểm của phương pháp

- Đáp ứng tốt với sự thay đổi thông số hệ thống: Chiều dài dây treo, khối lượng tải trọng
- Khi phân tích đưa ra thông số bộ điều khiển không cần nắm quá rõ thông số mô hình hệ thống

1.2.3 Nội dung đồ án

Với đề tài tài trên, nội dung nghiên cứu chính của đồ án bao gồm:

- Tìm hiểu về cầu trục, mô hình hóa cho đối tượng cầu trục có tải trọng và móc tạo thành hệ dao động con lắc đơn, mô phỏng đối tượng cầu trục trên matlab/simulink.
- Cải tiến bộ lọc Input shaping nhờ ứng dụng mạng neural. Sử dụng thuật toán tối ưu hóa bầy hạt Particle Swarm Optimization (PSO) tiến hành tạo mẫu và huấn luyện cho bộ Input shaping
- Ứng dụng bộ điều khiển ADRC xây dựng bộ điều khiển vị trí cho cầu trục.
- Tiến hành chạy mô phỏng trên mô hình Matlab/simulink và chạy thực nghiệm đối với trường hợp chiều dài dây thay đổi.

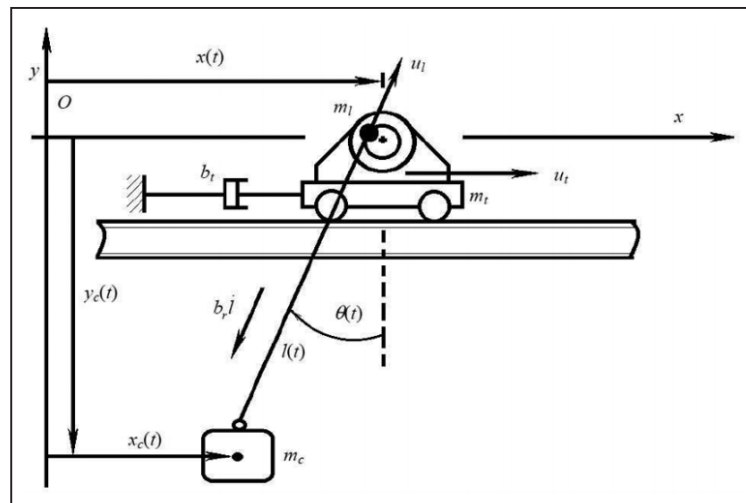
CHƯƠNG 2. XÂY DỰNG MÔ HÌNH MÔ PHỎNG

Chương 2 thực hiện việc xây dựng mô hình toán học cho hệ thống cầu trục con lắc đơn. Đồng thời tiến hành xây dựng mô hình cầu trục từ mô hình toán học trên Matlab simulink và tiến hành kiểm chứng kết quả mô hình khi tác động một lực đẩy F lên cầu trục. Tiếp đến sẽ thực hiện việc thiết kế bộ điều khiển ADRC điều khiển vị trí và chiều dài dây. Cuối cùng giới thiệu các bộ Input Shaping đã áp dụng cho các bài báo trước đây.

2.1 Mô hình hóa hệ thống cầu trục

2.1.1 Mô hình toán học hệ thống cầu trục

Xuất phát từ yêu cầu thực tế: Trên quãng đường dịch chuyển của hàng hóa, cần phải tránh vật cản, điều này yêu cầu hệ thống phải thay đổi chiều dài dây. Tức có ba biến phải theo dõi bao gồm: Vị trí chuyển động hệ thống cầu trục $x(t)$, chiều dài dây $l(t)$, góc giao động $\theta(t)$, khối lượng tải m_c , khối lượng dây dẫn m_l , khối lượng cơ cấu di chuyển m_t , hệ số cản cơ cấu di chuyển trục ngang b_t , hệ số cản của con lắc b_r .



Hình 2.1 Mô hình cầu trục 2D

Trong đó:

- $x(t)$: vị trí chuyển động của hệ thống cầu trục (Chiều dương là chiều từ trái sang phải theo phương x)
- $l(t)$: chiều dài dây (Chiều dương là chiều hướng từ dưới lên trên theo phương y)
- $\theta(t)$: góc dao động (Chiều dương là chiều cùng so với vị trí)
- m_l : khối lượng cơ cấu nâng hạ
- m_c : khối lượng tải
- m_t : khối lượng xe tải

- u_l : tín hiệu điều khiển dây
- u_t : tín hiệu điều khiển tải
- $x_c(t)$: khoảng cách từ con lắc đến trục Y

Để thuận tiện cho việc mô hình hóa, các giả thiết sau đây được đưa ra:

- Bỏ qua khối lượng và hệ số đàn hồi của sợi dây.
- Bỏ qua sự ảnh hưởng của gió tác động từ bên ngoài nhà xưởng vì cầu trục thường được vận hành trong nhà xưởng.
- Chuyển động của tất cả các thành phần hệ thống được xem xét trong một hệ.
- Góc xoay của hàng bị giới hạn $\theta \leq \frac{\pi}{2}$, và chiều dài dây cáp: $0.2 \leq l \leq 0.65$

Mô hình động cho cầu trục bao gồm các hoạt động đồng thời chuyển động của xe đẩy và cầu hàng. Vị trí tải trọng x_c, y_c được xác định bởi:

$$x_c = x + l \sin \theta \quad (2.1)$$

$$y_c = -l \cos \theta \quad (2.2)$$

Động năng và thế năng của cầu trục và tải trọng được biểu diễn như sau:

$$K = \frac{1}{2} (m_t \dot{x}^2 + m_l \dot{l}^2) + \frac{m}{2} v_c^2 \quad (2.3)$$

$$P = m_c g l (1 - \cos \theta) \quad (2.4)$$

Trong đó $g = 9.81 \text{ m/s}^2$, v_c vận tốc tải trọng. Tốc độ tải trọng được mô tả bằng phương trình dưới đây:

$$v_c^2 = \dot{x}_c^2 + \dot{y}_c^2 = \dot{x}^2 + \dot{l}^2 + l^2 \dot{\theta}^2 + 2(\sin \theta \dot{l} + l \cos \theta \dot{\theta}) \dot{x} \quad (2.5)$$

Hàm Lagrange L và hàm tiêu tán Rayleigh F xác định như sau:

$$L = K - P = \frac{1}{2} (m_t \dot{x}^2 + m_l \dot{l}^2) + \frac{m}{2} v_c^2 - m_c g l (1 - \cos \theta) \quad (2.6)$$

$$F = \frac{1}{2} (b_t \dot{x}^2 + b_r \dot{l}^2 + b_c \dot{\theta}^2) \quad (2.7)$$

Ở đây b_t, b_r, b_c , là các hệ số ma sát, lực cản liên quan đến các chuyển động của x, l , và góc dao động tương ứng. Sử dụng nguyên lý công ảo và phương trình Lagrange, các phương trình chuyển động mô tả động lực học của hệ thống như sau [1]:

$$(m_t + m_c) \ddot{x} + m_c \sin \theta \ddot{l} + m_c l \cos \theta \ddot{\theta} + b_t \dot{x} + 2m_c \cos \theta \dot{l} \dot{\theta} - m_c l \sin \theta \dot{x}^2 = u_t(t) \quad (2.8)$$

$$m_c \sin \theta \ddot{x} + (m_c + m_l) \ddot{l} + b_r \dot{l} - m_c l \dot{\theta}^2 - m_c g \cos \theta = u_l(t) \quad (2.9)$$

$$m_c l^2 \ddot{\theta} + m_c l \cos \theta \ddot{x} + 2m_c l \dot{l} \dot{\theta} + m_c g l \sin \theta + b_c \dot{\theta} = 0 \quad (2.10)$$

Bảng 2.1 Thông số mô hình toán học cầu trục

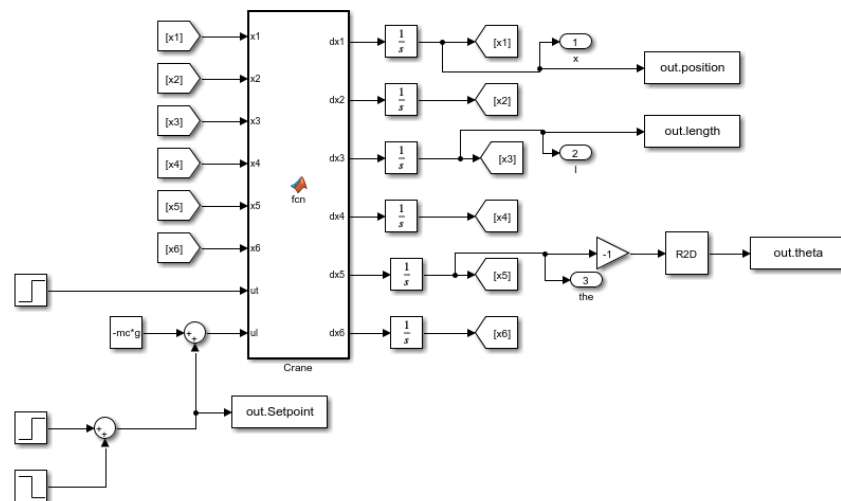
| Thông số | Giá trị |
|---|---------------|
| Khối lượng tải m_p | 0.8 – 1.5 kg |
| Khối lượng xe đẩy m_t | 4 kg |
| Khối lượng cơ cấu nâng hạ m_l | 3 kg |
| Chiều dài dây l | 0.2 – 0.65 m |
| Hệ số cản b_t giữa xe lăn và trục x | 20 Ns/m |
| Hệ số cản b_r giữa giữa dây và ngoại quan | 50 Ns/m |
| Gia tốc trọng trường | $9.81 s^{-2}$ |
| Hệ số giảm xóc (b_c) | 0.1 Ns/rad |

2.1.2 Kiểm chứng mô hình toán học

Để xác thực mô hình toán học thu được, các kịch bản thử nghiệm sẽ được thực hiện trong MATLAB/Simulink bao gồm:

- Chuyển động của cơ cấu theo phương ngang
- Chuyển động của cơ cấu theo phương dọc
- Chuyển động theo cả hai phương

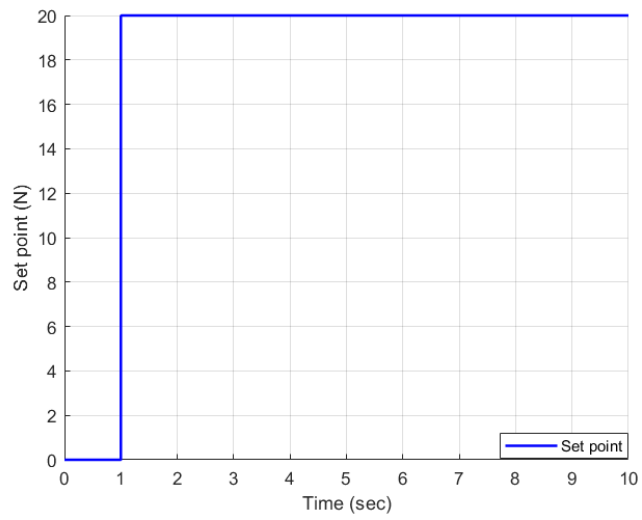
2.1.3 Triển khai và kiểm chứng mô hình trên Matlab Simulink



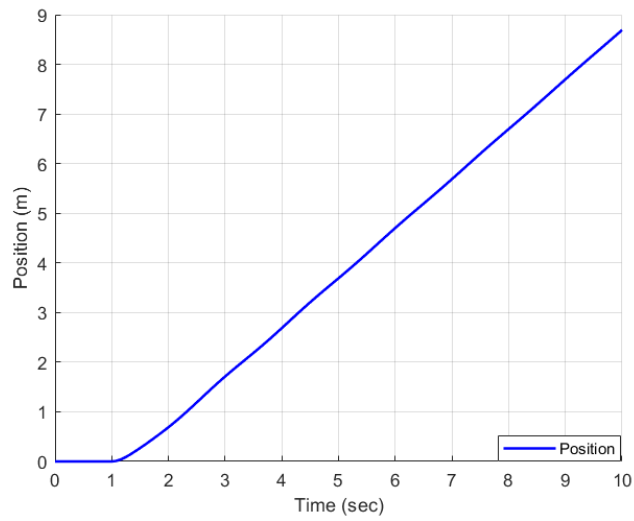
Hình 2.2 Mô hình cầu trục trên Matlab Simulink

2.1.3.1. TH1 chuyển động cơ cấu theo phương ngang $F_x = 20\text{ N}$

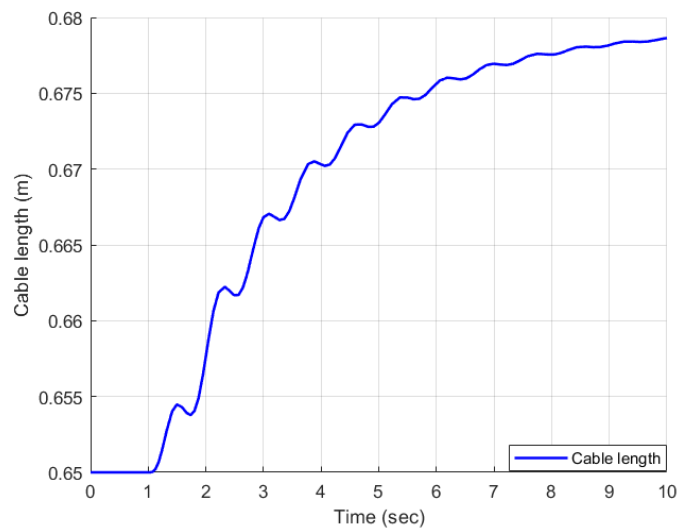
Tác dụng một lực không đổi $F_x = 20\text{ N}$ (Hình 2.2) để xe đẩy chuyển động dọc theo trục x. Ban đầu, xe đẩy tăng tốc khi tác dụng lực đẩy F_x (Hình 2.3) và sau đó chuyển sang chuyển động thẳng đều dọc theo trục x dương. Chuyển động ban đầu của con lăn này gây ra dao động tải trọng, ảnh hưởng đến chiều dài cáp như trong Hình 2.4. Mức độ giảm chiều dài cáp phụ thuộc vào độ lớn của dao động tải trọng Hình 2.5.



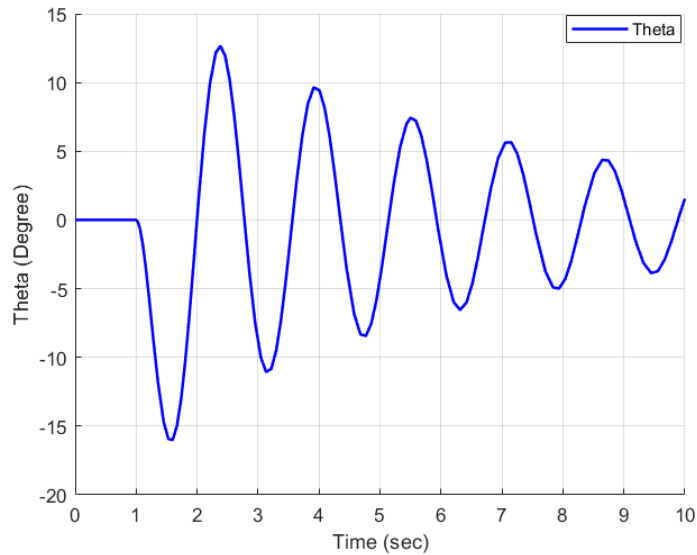
Hình 2.3 Giá trị đặt vào trục x theo TH1



Hình 2.4 Đáp ứng vị trí khi x theo TH1



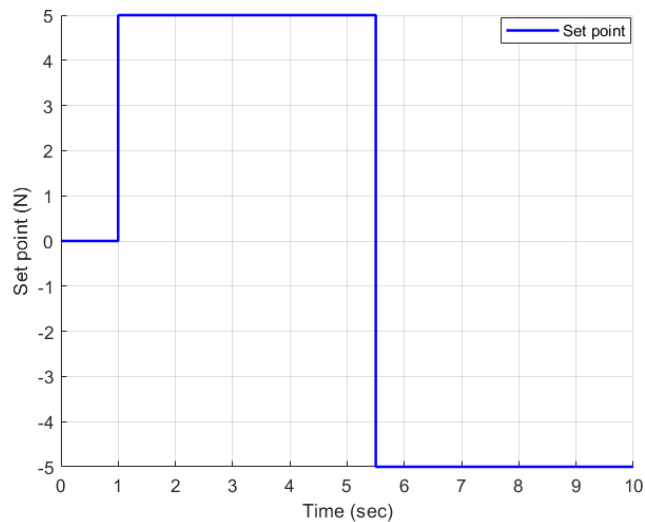
Hình 2.5 Đáp ứng chiều dài dây x theo TH1



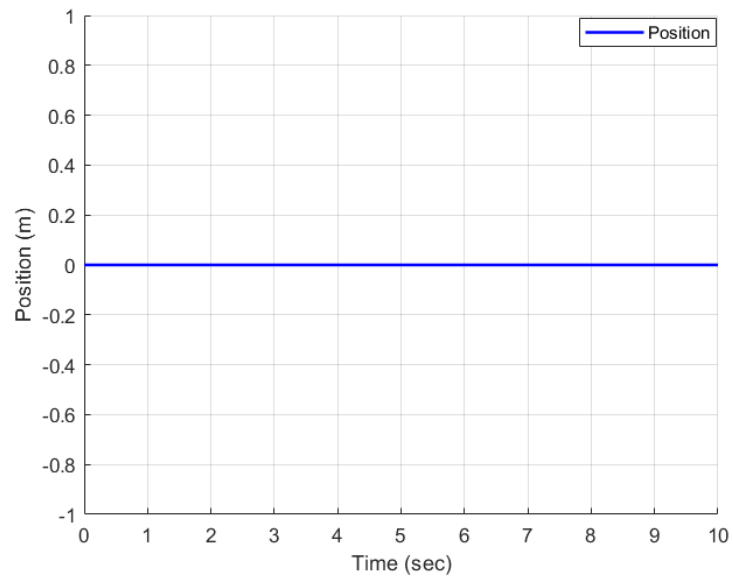
Hình 2.6 Đáp ứng góc giao động x theo TH1

2.1.3.2. Chuyển động của cơ cấu theo phương dọc $F_y=5N$

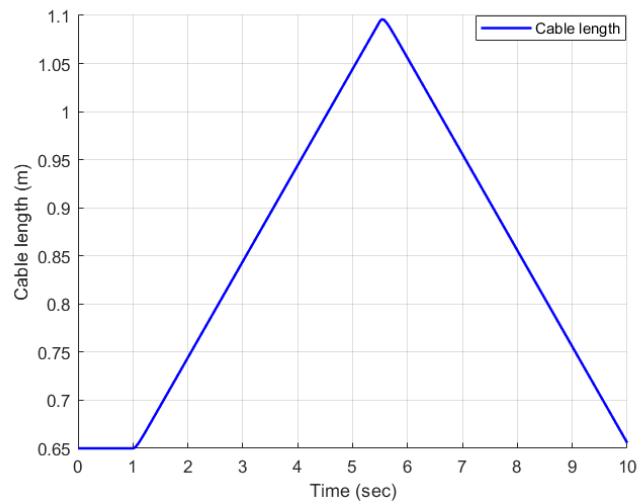
Trong trường hợp này, thực hiện việc hạ thấp có kiểm soát tải trọng từ vị trí ban đầu của nó ở mức 0,2 m, sau đó nâng tải trọng lên bằng cách sử dụng lực F_1 theo thời gian, như được mô tả trong Hình 2.7. Theo quan sát, ban đầu tải trọng sẽ chịu gia tốc đều và sau đó ổn định thành chuyển động ổn định dọc theo trục y , như thể hiện trong Hình 2.9. Điều đáng chú ý là trong quá trình nâng chuyển động của tải trọng, không có tác động nào đến sự dịch chuyển của xe con và không có dao động nào được tạo ra trong tải trọng, như thể hiện rõ trong Hình 2.8 và 2.10



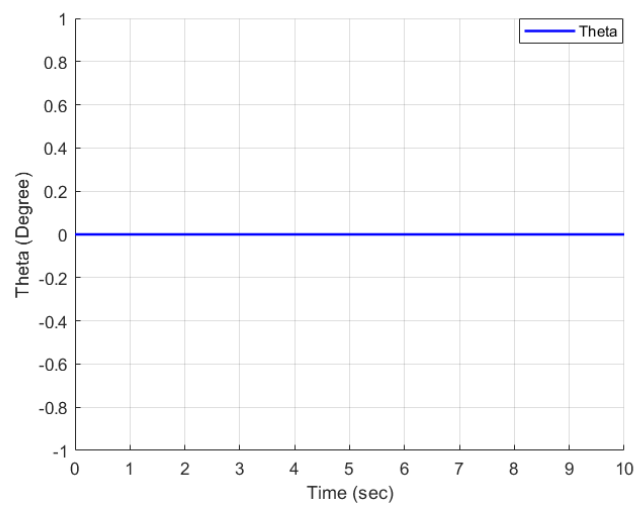
Hình 2.7 Giá trị đặt vào trục z theo TH2 (Tác động $F_z=5N$ theo phương z)



Hình 2.8 Đáp ứng vị trí theo TH2



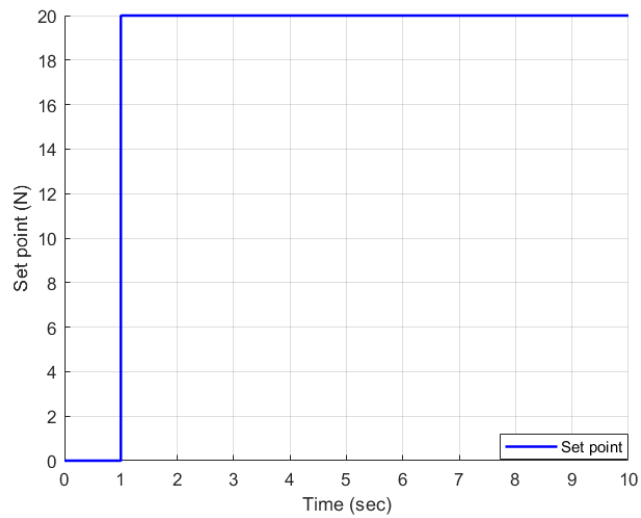
Hình 2.9 Đáp ứng chiều dài dây theo TH2



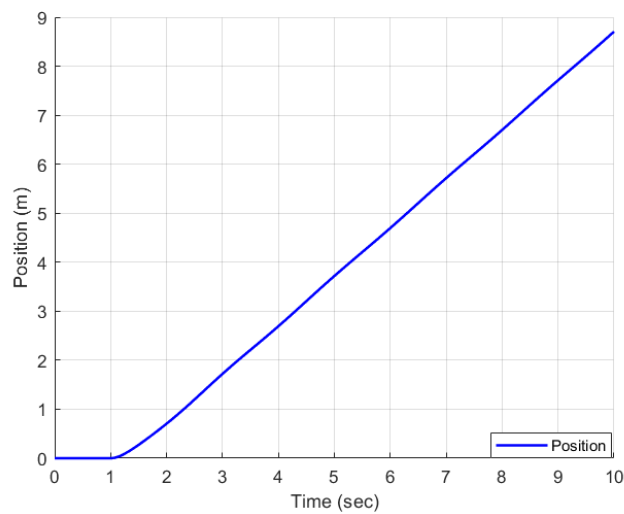
Hình 2.10 Đáp ứng góc lắc theo TH2

2.1.3.3. Chuyển động theo cả hai phương

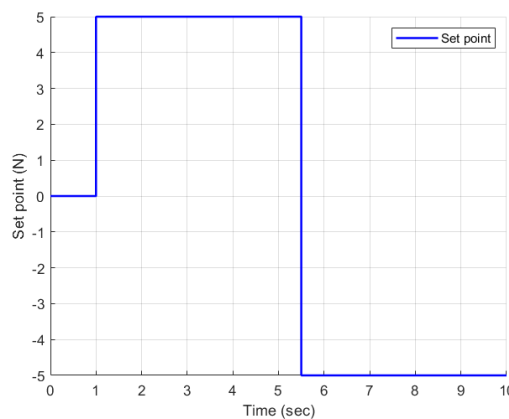
Lực đẩy $F_x = 20 \text{ N}$ và lực nâng F_l sẽ được tác dụng đồng thời vào mô hình hệ thống cầu trục. Từ Hình 2.12 và 2.14, 2.15, phản hồi của hệ thống gần giống với kết quả thu được từ các trường hợp thử nghiệm riêng lẻ đã nói ở trên.



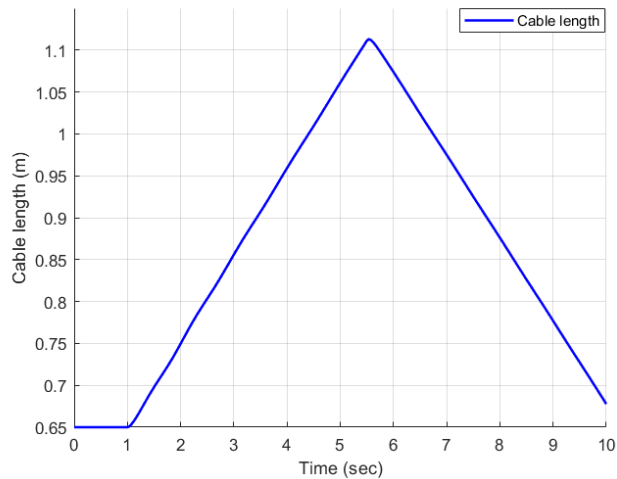
Hình 2.11 Giá trị đặt vào trục x theo TH3 (Tác động lực vào cả hai phương)



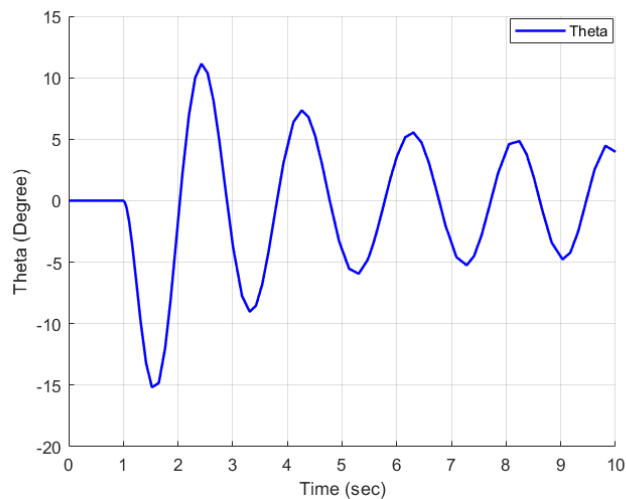
Hình 2.12 Đáp ứng vị trí theo TH3



Hình 2.13 Giá trị đặt vào trục z theo TH3



Hình 2.14 Đáp ứng chiều dài dây theo trường hợp 3



Hình 2.15 Đáp ứng góc lắc theo TH3

Kết luận : Mô hình của hệ thống cần trục, bao gồm chuyển động tịnh tiến của xe con và quá trình nâng tải trọng. Thông qua các kịch bản mô phỏng thực quan cho thấy: Mô hình toán học và mô hình mô phỏng biểu diễn tương đối chính xác về đặc tính đối tượng cần trục. Biểu thị được các biến theo dõi cần thiết như vị trí chuyển động hệ thống cần trục $x(t)$, chiều dài dây $l(t)$, góc giao động $\theta(t)$,

2.2 Thiết kế bộ điều khiển ADRC cho mô hình mô phỏng

2.2.1 Bộ điều khiển ADRC

Từ khi được giới thiệu, bộ điều khiển loại bỏ nhiễu chủ động ADRC đã được xem như một phương pháp thay thế cho bộ điều khiển PID truyền thống . ADRC được coi là một chiến lược điều khiển mạnh mẽ và triển khai nó dễ dàng hơn so với bộ điều khiển PID truyền thống, vì không yêu cầu một mô hình toán học chính xác để thiết kế các tham số điều khiển. ADRC tạo ra một biến trạng thái mới bao gồm cả động học và nhiễu loạn chưa biết. Nền tảng của ADRC là bộ quan sát trạng thái mở rộng (ESO), nơi bộ quan sát này theo dõi và ước lượng nhiễu loạn thực tế và sai số của mô hình toán học của đối tượng. Nhờ đó, bộ điều khiển

có thể được thiết kế với chất lượng tốt và mạnh mẽ để chống lại các nhiễu loạn thực tế, ngay cả khi chỉ có một mô hình toán học không chính xác .

Xét một hệ thống bậc 2 có dạng:

$$P(s) = \frac{y(s)}{u(s)} = \frac{K}{T^2 s^2 + 2DTs + 1} \quad PT\ 2.1$$

Trong đó:

K - hệ số khuếch đại

D - hệ số tắt dần

T - hằng số thời gian

Theo tài liệu [2], việc thiết kế bộ điều khiển ADRC được thực hiện như sau:

Biến đổi Laplace ngược phương trình 3.1, ta được:

$$T^2 \cdot \ddot{y}(t) + 2DT \cdot \dot{y}(t) + y(t) = K \cdot u(t) \quad PT\ 2.2$$

Thêm vào phương trình 2.2 một nhiễu đầu vào $d(t)$, đặt $b = \frac{K}{T^2}$, chuyển b thành hai phần, phần đã biết b_0 và phần chưa biết Δb , $b = b_0 + \Delta b$, ta được:

$$\ddot{y}(t) = -\frac{2D}{T} \cdot \dot{y}(t) - \frac{1}{T^2} \cdot y(t) + \frac{1}{T^2} \cdot d(t) + \Delta b \cdot u(t) + b_0 \cdot u(t) \quad PT\ 2.3$$

Đặt $f(t) = -\frac{2D}{T} \cdot \dot{y}(t) - \frac{1}{T^2} \cdot y(t) + \frac{1}{T^2} \cdot d(t) + \Delta b \cdot u(t)$ được gọi là nhiễu tổng quát, ta được:

$$\ddot{y}(t) = f(t) + b_0 \cdot u(t) \quad PT\ 2.4$$

Phương trình 3.4 có thể viết dưới dạng không gian trạng thái như sau:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}}_A \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix}}_B \cdot u(t) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{f}(t) \quad PT\ 2.5$$

$$y(t) = \underbrace{(1 \quad 0 \quad 0)}_C \begin{pmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{pmatrix} \quad PT\ 2.6$$

Ta có bộ quan sát trạng thái mở rộng có dạng như sau:

$$\begin{aligned}
\begin{pmatrix} \dot{\hat{x}}_1(t) \\ \dot{\hat{x}}_2(t) \\ \dot{\hat{x}}_3(t) \end{pmatrix} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{pmatrix} + \begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix} \cdot u(t) \\
&+ \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} (y(t) - \hat{x}_1(t)) \\
&= \underbrace{\begin{pmatrix} -l_1 & 1 & 0 \\ -l_2 & 0 & 1 \\ -l_3 & 0 & 0 \end{pmatrix}}_{A-LC} \begin{pmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{x}_3(t) \end{pmatrix} + \underbrace{\begin{pmatrix} 0 \\ b_0 \\ 0 \end{pmatrix}}_B \cdot u(t) \\
&+ \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \cdot y(t)
\end{aligned}
\tag{PT 2.7}$$

Trong đó:

$$\hat{x}_1(t) = \hat{y}(t) \approx y(t)$$

$$\hat{x}_2(t) = \dot{\hat{y}}(t) \approx \dot{y}(t)$$

$$\hat{x}_3(t) = \hat{f}(t) \approx f(t)$$

Từ đó, việc loại bỏ nhiễu dựa vào luật điều khiển sau:

$$u(t) = \frac{(u_0 - \hat{f}(t))}{b_0} \tag{PT 2.8}$$

Với:

$$u_0(t) = K_p(r(t) - \hat{y}(t)) - K_D \cdot \dot{\hat{y}}(t) \tag{PT 2.9}$$

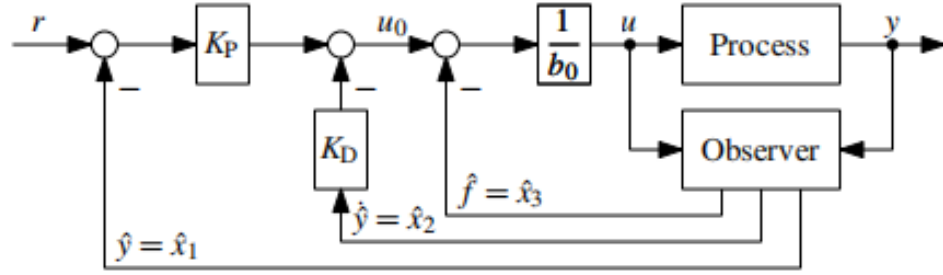
Thay phương trình 2.8, 2.9 vào phương trình 2.4, ta được:

$$\ddot{y}(t) \simeq u_0 = K_p(r(t) - y(t)) - K_D \cdot \dot{y}(t) \tag{PT 2.10}$$

Từ đó, ta thu được hàm truyền:

$$\frac{Y(s)}{R(s)} = \frac{K_p}{s^2 + K_D \cdot s + K_p} \tag{PT 2.11}$$

Sơ đồ khối của bộ điều khiển ADRC như hình 3.1:



Hình 2.16 Cấu trúc bộ điều khiển ADRC

2.2.2 Thiết kế bộ điều khiển ADRC cho mô hình mô phỏng

Từ phương trình (2.8) và (2.9) ta có phương trình vi phân của vị trí xe đẩy $x(t)$ và chiều dài cáp $l(t)$ là:

$$\ddot{x}(t) = f_x(t, \dot{x}, x, \dot{l}, l, \dot{\theta}, \theta, F_l) + \frac{m_c + m_l}{m_t m_c + m_t m_l + m_c m_l} \cdot F_x \quad (2.12)$$

$$= f_l(t, \dot{x}, x, \dot{l}, l, \dot{\theta}, \theta, F_x) + b_{0x} \cdot F_l$$

$$\ddot{l}(t) = f_l(t, \dot{x}, x, \dot{l}, l, \dot{\theta}, \theta, F_x) + \frac{m_c + m_t}{m_t m_c + m_t m_l + m_c m_l} \cdot F_l \quad (2.13)$$

$$= f_l(t, \dot{x}, x, \dot{l}, l, \dot{\theta}, \theta, F_x) + b_{0l} \cdot F_l$$

Để đạt được vị trí đồng thời của xe đẩy và chiều dài cáp, đặt thời gian ổn định là $T_{\text{settle}} = 3$ giây và độ lợi của người quan sát là $k_{\text{ESO}}=15$ cho cả vị trí xe đẩy và bộ điều khiển ADRC chiều dài cáp. Bằng cách tham khảo Mục 3.2 và xem xét các tham số hệ thống được cung cấp trong Bảng, chúng tôi đã tính toán các tham số cụ thể cho hai ADRC, như được trình bày trong Bảng 5.1.

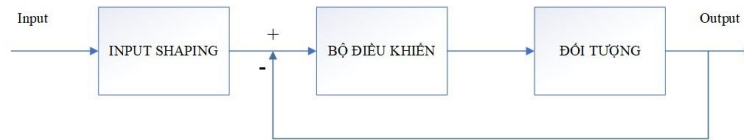
Trong Bảng đã chọn khối lượng tải trọng là $m_c=0.85$ kg, nằm trong phạm vi $[0.8, 1.5]$ kg, cho phép đánh giá độ bền của ADRC đối với các thay đổi về khối lượng tải trọng. Trường hợp thay đổi chiều dài dây khi cầu trục đang di chuyển

| Thông số bộ điều khiển | ADRC1 | ADRC2 |
|------------------------|----------|----------|
| b_0 | 0.1787 | 0.3668 |
| s_{CL} | -1.95 | -1.95 |
| K_P | 3.8025 | 3.8025 |
| K_D | 3.9 | 3.9 |
| s_{ESO} | -19.5 | -19.5 |
| l_1 | 58.5 | 58.5 |
| l_2 | 1140.75 | 1140.75 |
| l_3 | 7414.875 | 7414.875 |

2.3 Giới thiệu bộ lọc Input Shaping

2.3.1 Thuật toán định hình đầu vào (Input shaping)

Khi cầu trục hoạt động, thường xảy ra hiện tượng rung động, gây hạn chế về năng suất và chất lượng của hệ thống. Để giải quyết vấn đề này, chúng ta sử dụng một kỹ thuật gọi là định hình đầu vào (input shaping). Kỹ thuật này sử dụng một chuỗi các xung điều khiển để giảm hoặc loại bỏ rung động dư thừa. Để thiết kế chuỗi xung, chúng ta cần biết các thông số của mô hình rung động. Từ đó, chúng ta có thể xác định thời gian và biên độ của từng xung trong chuỗi. Điều này được thực hiện bằng cách giải các phương trình ràng buộc.



Hình 2.17 Cấu trúc điều khiển thuật toán Input Shaping

Một điểm mạnh của kỹ thuật định hình đầu vào là nó được đặt bên ngoài mạch điều khiển phản hồi. Điều này có nghĩa là nó không ảnh hưởng đến sự ổn định của hệ thống điều khiển. Điều này khác biệt so với các phương pháp điều khiển phản hồi thông thường. Tóm lại, kỹ thuật định hình đầu vào giúp chúng ta tạo ra tín hiệu điều khiển mong muốn, giúp cầu trục di chuyển chính xác và loại bỏ rung động dư thừa.

Khi có một xung đầu vào tác động lên hệ thống thì đầu ra của hệ thống sẽ là một tín hiệu dao động tắt dần. Ý tưởng của việc chống rung cho một hệ thống dựa trên phương pháp tạo dạng tín hiệu đầu vào input shaping là tạo ra một tín hiệu xung thứ 2 có độ lớn và thời điểm tác động phù hợp để triệt tiêu đi dao động tạo ra của tín hiệu xung thứ nhất làm cho tổng dao động của hệ bị triệt tiêu.

2.3.2 Thiết kế bộ lọc Input Shaping

Xét một hệ thống rung đơn giản có thể được biểu diễn dưới dạng hệ thống bậc hai như sau:

$$G_s = \frac{Y(s)}{V(s)} = \frac{\omega_0^2}{s^2 + 2\omega_n \zeta_o s + \omega_0^2}$$

Đáp ứng của xung $A_i \delta(t - t_i)$ với hệ thống bậc 2 có tần số dao động riêng ω_n và hệ số tắt dần ξ là một véc-tơ xung \mathbf{I}_i có biên độ I_i và góc θ_i được biểu diễn trong hệ tọa độ 2 chiều với:

$$I_i = A_i e^{\xi \omega_n t_i}, \quad \theta_i = \omega_d t_i$$

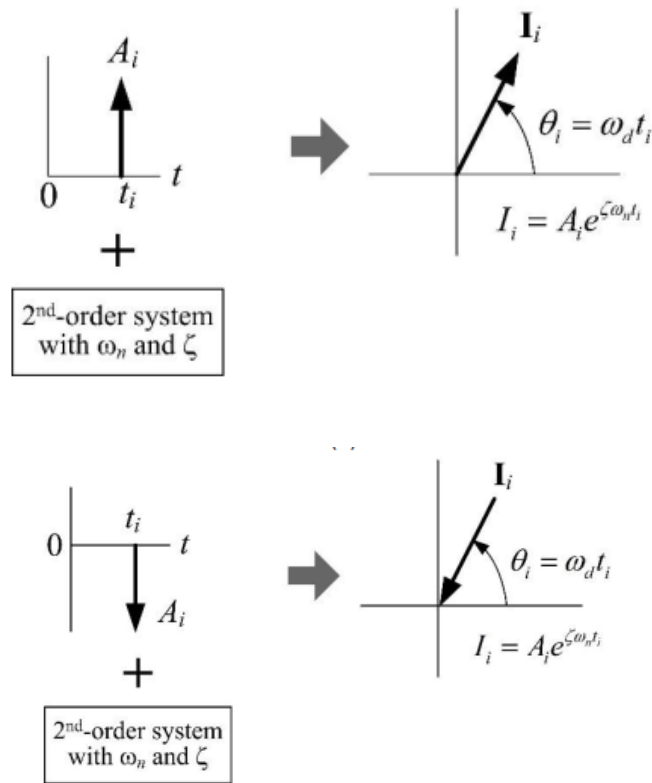
Trong đó:

A_i là biên độ của xung

t_i là thời điểm phát xung

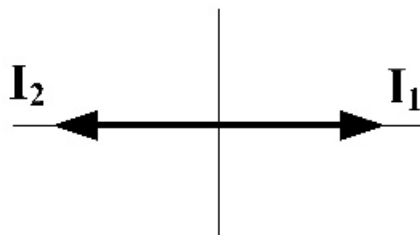
$\omega_d = \omega_n \sqrt{1 - \xi^2}$ là tần số dao động tắt dần

Đối với xung dương ($A_i > 0$), điểm ban đầu của véc-tơ xung nằm ở điểm gốc của hệ tọa độ cực, trong khi đối với xung âm ($A_i < 0$), điểm cuối của véc-tơ xung được đặt tại gốc tọa độ.



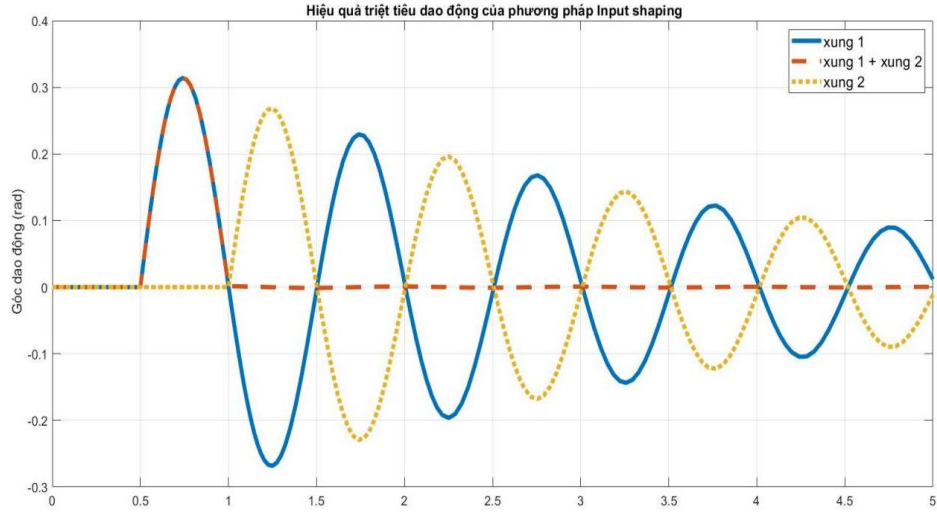
Hình 2.18 Đáp ứng của hệ bậc hai với vector xung

Áp dụng việc sử dụng vectơ xung thay cho phương pháp cổ điển đối với ZV shaper ta có:



Hình 2.19 Vị trí của Vector xung ZV Shaper

Khi có một xung đầu vào tác động lên hệ thống thì đầu ra của hệ thống sẽ là một tín hiệu dao động tắt dần. Ý tưởng của việc chống rung cho một hệ thống dựa trên phương pháp tạo dạng tín hiệu đầu vào input shaping là tạo ra một tín hiệu xung thứ 2 có độ lớn và thời điểm tác động phù hợp để triệt tiêu đi dao động tạo ra của tín hiệu xung thứ nhất làm cho tổng dao động của hệ bị triệt tiêu.



Hình 2.20 Đáp ứng của hệ với các xung riêng lẻ và đáp ứng với chuỗi xung

Ta đặt vectơ xung thứ nhất tại góc 0 (rad). Để triệt tiêu dao động, vectơ xung đặt tại góc π (rad) với độ lớn bằng độ lớn của xung thứ nhất. Ta thu được các phương trình sau:

$$I_1 = A_1 e^{\zeta \omega_n t_1}, \theta_1 = \omega_d t_1 = 0 \quad (3.12)$$

$$I_2 = A_2 e^{\zeta \omega_n t_2}, \theta_2 = \omega_d t_2 = \pi \quad (3.13)$$

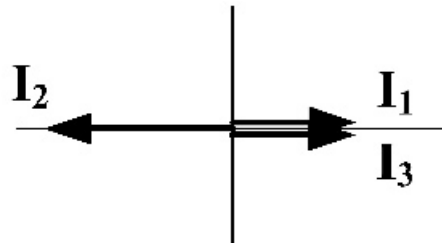
$$I_1 = I_2 \quad (3.14)$$

$$A_1 + A_2 = 1 \quad (3.15)$$

Giải các phương trình từ 3.12 – 3.15, ta thu được kết quả :

$$\begin{bmatrix} t_i \\ A_i \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{\omega_d} \\ \frac{K}{1+K} & \frac{1}{1+K} \end{bmatrix}, K = e^{\frac{\xi \pi}{\sqrt{1-\xi^2}}}$$

Đối với trường hợp input shaper từ 3 xung trở lên, ta có nhiều cách để thiết kế với phương pháp vectơ xung, để thuận tiện trong việc tính toán ta có thể đặt 3 vectơ xung lần lượt tại thời điểm 0, π và 2π rad với độ lớn tỉ lệ độ lớn là 1:2:1 như hình



Vị trí của vectơ xung ZVD shaper

Ta có các phương trình sau:

$$I_1 = A_1 e^{\zeta \omega_n t_1}, \theta_1 = \omega_d t_1 = 0 \quad (3.16)$$

$$I_2 = A_2 e^{\zeta \omega_n t_2}, \theta_2 = \omega_d t_2 = \pi \quad (3.17)$$

$$I_3 = A_3 e^{j\omega_n t_3}, \theta_3 = \omega_d t_3 = 2\pi \quad (3.18)$$

$$I_1 = I_3 = \frac{I_2}{2} \quad (3.19)$$

$$A_1 + A_2 + A_3 = 1 \quad (3.20)$$

Giải các phương trình 3.16 – 3.20, ta thu được ZVD shaper như sau:

$$\begin{bmatrix} t_i \\ A_i \end{bmatrix} = \begin{bmatrix} 0 & \frac{\pi}{w_d} & \frac{2\pi}{w_d} \\ K^2 & 2K & 1 \\ \frac{1}{(1+K)^2} & \frac{2K}{(1+K)^2} & \frac{1}{(1+K)^2} \end{bmatrix}, K = e^{\frac{\xi\pi}{\sqrt{1-\xi^2}}} \quad (3.21)$$

Với phương pháp sử dụng vecto xung, việc thiết kế input shaper trở nên dễ dàng thực hiện hơn dù số lượng vecto xung có thể trở nên nhiều hơn với n xung. Tuy nhiên với việc tăng số lượng xung nên thì thời gian thực hiện trở nên lâu hơn do các vecto xung được bố trí nhiều hơn một vòng tròn lượng giác. Việc tăng thời gian sẽ khiến cho thời gian đáp ứng của đối tượng trở nên chậm hơn, không đảm bảo được thời gian đáp ứng cho đối tượng. Tuy nhiên, có một phương pháp khác cho phép thiết kế input shaper với n xung nhưng vẫn đảm bảo việc thực hiện trong một vòng tròn lượng giác tức là có thời gian thực hiện tương đương với trường hợp 3 xung ZVD.

CHƯƠNG 3. TRIỂN KHAI BỘ LỌC INPUT SHAPING CẢI TIẾN TRÊN MÔ HÌNH MÔ PHỎNG

Ở chương 3, trình bày các bước xây dựng mạng Neural dựa trên cơ sở thuật toán Input Shaping. Sau đó triển khai trên mô hình Matlab sử dụng kết hợp với mô hình toán học và bộ điều khiển ADRC đã được xây dựng ở chương 2 và 3. Cuối cùng so sánh các bộ Input Shaping truyền thống đã giới thiệu ở chương 3

3.1 Mạng neural

3.1.1 Tổng quan về mạng Neural

Mạng neural, hay còn được gọi là mạng nơ-ron nhân tạo, là một khái niệm quan trọng trong lĩnh vực trí tuệ nhân tạo và học máy. Được lấy cảm hứng từ cấu trúc và hoạt động của hệ thống nơ-ron trong não người, mạng neural là một mô hình tính toán có khả năng học và nhận biết mẫu. Mạng neural bao gồm một tập hợp các đơn vị tính toán gọi là nơ-ron nhân tạo, được kết nối với nhau thông qua các liên kết trọng số. Mỗi nơ-ron nhận đầu vào từ các nơ-ron khác, thực hiện một phép tính trên đầu vào đó, và sau đó truyền kết quả đến nơ-ron tiếp theo trong mạng. Các kết nối trọng số cho phép mạng neural "học" thông qua việc điều chỉnh giá trị của chúng dựa trên dữ liệu đầu vào và kết quả mong muốn. Mạng neural đã chứng tỏ khả năng xuất sắc trong nhiều nhiệm vụ, bao gồm nhận dạng mẫu, phân loại, dự đoán và điều khiển. Với sự phát triển của công nghệ và sức mạnh tính toán, mạng neural đã trở thành công cụ quan trọng trong nhiều lĩnh vực, bao gồm thị giác máy tính, xử lý ngôn ngữ tự nhiên, ô tô tự lái, y học và nhiều lĩnh vực khác.

Tuy mạng neural mang lại nhiều ưu điểm, nhưng việc xây dựng và huấn luyện mạng có thể đòi hỏi nhiều thời gian và tài nguyên tính toán. Ngoài ra, việc hiểu và giải thích quyết định của mạng neural cũng là một thách thức. Tuy nhiên, nhờ sự phát triển liên tục và tiềm năng lớn, mạng neural đang trở thành một công cụ quan trọng trong việc giải quyết các vấn đề phức tạp và tạo ra những tiến bộ đáng kể trong lĩnh vực trí tuệ nhân tạo.

3.1.2 Lựa chọn cấu trúc mạng neural

Mạng neural nhân tạo tiêu chuẩn (Feedforward Neural Network):

- Đặc điểm: Cấu trúc đơn giản, gồm các lớp nút và truyền tuyến tính từ lớp đầu vào đến lớp đầu ra.
- Ưu điểm: Dễ triển khai, phù hợp cho các bài toán phân loại đơn giản.
- Nhược điểm: Không thể xử lý các tác vụ phức tạp, không giữ được thông tin về mối quan hệ không gian hay thời gian.

Mạng neural tái cấu trúc (Recurrent Neural Network):

- Đặc điểm: Cấu trúc có các chuỗi liên kết và kết nối phản hồi, giúp duy trì thông tin qua thời gian.
- Ưu điểm: Xử lý dữ liệu dạng chuỗi, đưa ra dự đoán dựa trên ngữ cảnh trước đó.

- Nhược điểm: Dễ bị ảnh hưởng bởi hiện tượng mất thông tin lâu dài (vanishing/exploding gradient), không thích hợp cho dữ liệu dài.

Mạng neural hồi quy (Convolutional Neural Network):

- Đặc điểm: Sử dụng các lớp tích chập để trích xuất đặc trưng từ dữ liệu không gian như ảnh.
- Ưu điểm: Hiệu quả trong việc xử lý dữ liệu không gian, phân loại hình ảnh, nhận dạng đối tượng.
- Nhược điểm: Không thể xử lý các dữ liệu có cấu trúc phức tạp hơn như dữ liệu dạng chuỗi hoặc biểu đồ.

Mạng neural hợp nhất (Generative Adversarial Network):

- Đặc điểm: Gồm một mạng sinh tạo dữ liệu giả và một mạng phân biệt dữ liệu thật/giả, huấn luyện qua việc cạnh tranh.
- Ưu điểm: Tạo ra dữ liệu mới theo phân phối huấn luyện, ứng dụng trong tạo ảnh, âm thanh, văn bản, ...
- Nhược điểm: Khó khăn trong việc ổn định huấn luyện, có thể dẫn đến việc tạo ra kết quả không chính xác hoặc nhạy cảm.

Mạng neural gắn kết (Graph Neural Network):

- Đặc điểm: Xử lý dữ liệu biểu đồ hoặc dữ liệu có cấu trúc mạng lưới, như mạng xã hội, phân tử hóa học.
- Ưu điểm: Tận dụng thông tin cấu trúc của mạng để thực hiện các tác vụ như phân loại, dự đoán trạng thái.
- Nhược điểm: Đòi hỏi nhiều tài nguyên tính toán, khó khăn trong việc khám phá các cấu trúc biểu đồ phức tạp.

Các mạng neural này đại diện cho một phần trong hệ sinh thái mạng neural phong phú và mỗi mạng đều có ưu nhược điểm riêng, phù hợp với các bài toán và dữ liệu cụ thể. Trong đề án này, sử dụng mạng neural nhân tạo tiêu chuẩn (Feedforward Neural Network) vì tính đơn giản và hiệu quả của mạng neural này cho hệ thống. Trong mạng này, dữ liệu chuyển qua mạng từ lớp đầu vào, thông qua các lớp ẩn (nếu có), và cuối cùng đến lớp đầu ra. Mỗi nơ-ron trong mạng truyền thẳng tính toán kết quả dựa trên các trọng số và hàm kích hoạt.

3.1.3 Các bước xây dựng mạng Feedforward Neural Network

Quá trình xây dựng một mạng neural Feedforward Neural Network bao gồm các bước sau:

- Xác định kiến trúc mạng: Xác định số lượng lớp (input layer, hidden layers, output layer) và số lượng nút trong mỗi lớp. Quyết định các hàm kích hoạt (activation functions) cho các nút trong mạng.
- Chuẩn bị dữ liệu: Chuẩn bị dữ liệu huấn luyện và dữ liệu kiểm tra và tiền xử lý dữ liệu, bao gồm chia tỷ lệ, chuẩn hóa, mã hóa các biến đầu vào.
- Khởi tạo trọng số: Khởi tạo giá trị ban đầu cho các trọng số của mạng, có thể là ngẫu nhiên hoặc theo một phân phối xác định.

- Tiến hành huấn luyện: Sử dụng thuật toán để cập nhật trọng số theo chiều từ đầu ra đến đầu vào. Chia dữ liệu huấn luyện thành các batch và thực hiện các vòng lặp để cập nhật trọng số.
- Đánh giá và điều chỉnh mô hình: Đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra và kiểm tra. Nếu kết quả không đạt yêu cầu, điều chỉnh các tham số của mô hình như hàm kích hoạt, số lượng lớp, số lượng nút, tốc độ học (learning rate), v.v.
- Đánh giá và sử dụng mô hình: Đánh giá hiệu suất của mô hình trên dữ liệu kiểm tra độc lập. Sử dụng mô hình để thực hiện dự đoán trên dữ liệu mới.
- Tinh chỉnh và cải tiến: Tinh chỉnh mạng neural bằng cách thay đổi kiến trúc, thử các hàm kích hoạt khác nhau, tăng số lượng lớp hoặc nút, thay đổi thuật toán huấn luyện, v.v.

3.2 Thuật toán tối ưu hóa bầy đàn (PSO)

3.2.1 Tổng quan thuật toán tối ưu hóa bầy đàn

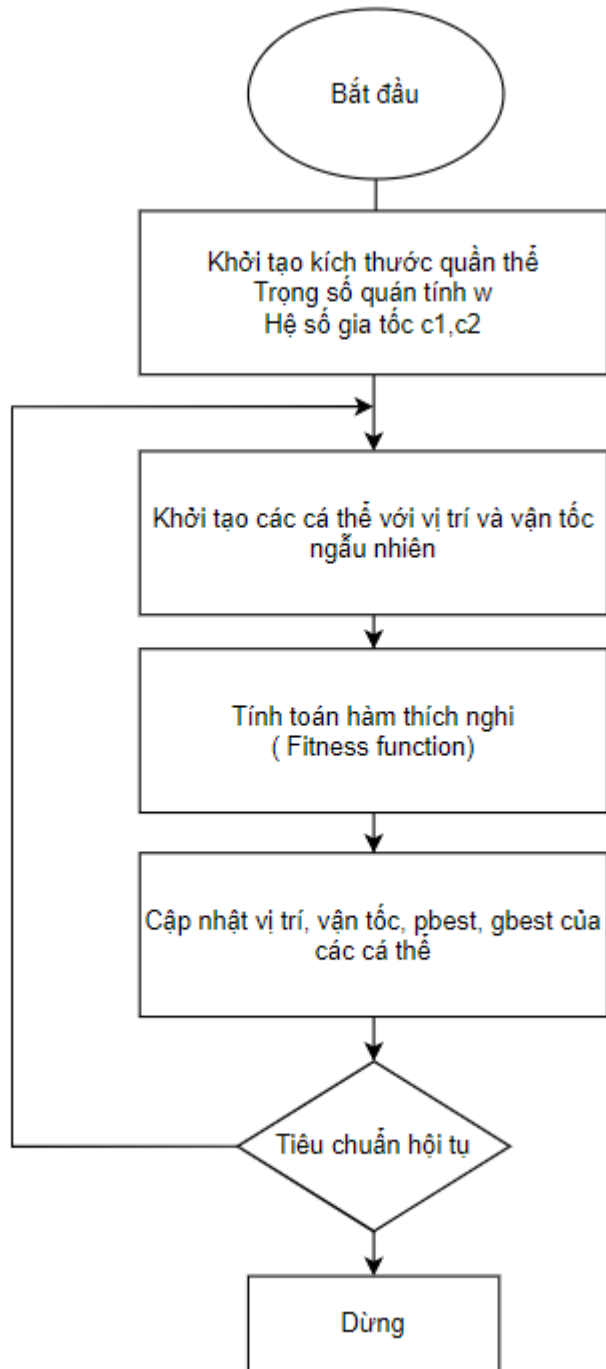
Thuật toán tối ưu hóa bầy đàn đi (Particle Swarm Optimization - PSO) là một thuật toán tìm kiếm được phát triển bởi James Kennedy và Russell Eberhart vào năm 1995, lấy cảm hứng từ hành vi tụ tập của đàn chim. PSO được thiết kế để tìm kiếm và tối ưu hóa các giải pháp trong không gian tìm kiếm lớn và không đồng nhất.

Ứng dụng của thuật toán PSO:

- Tối ưu hóa hàm số: PSO có thể được sử dụng để tìm giá trị tối ưu của một hàm số đơn giản hoặc phức tạp trong không gian đa chiều. Nó đã được áp dụng rộng rãi trong các lĩnh vực như tối ưu hóa vô điều kiện, tối ưu hóa có ràng buộc và tối ưu hóa đa mục tiêu.
- Tối ưu hóa mạng nơ-ron nhân tạo: PSO có thể được sử dụng để tối ưu hóa các trọng số và cấu trúc của mạng nơ-ron nhân tạo. Nó giúp tìm ra các bộ trọng số tối ưu để mạng nơ-ron có thể học và dự đoán tốt hơn trên tập dữ liệu huấn luyện.
- Tối ưu hóa trong các bài toán kỹ thuật: PSO có thể được áp dụng để tối ưu hóa các tham số và thiết kế trong các bài toán kỹ thuật như tối ưu hóa mô hình, lập lịch sản xuất, tối ưu hóa hệ thống vận tải, tối ưu hóa mạng viễn thông, và nhiều ứng dụng khác.
- Tìm kiếm trong không gian tìm kiếm lớn: PSO có thể được sử dụng để tìm kiếm giải pháp tối ưu trong không gian tìm kiếm lớn và không đồng nhất. Với khả năng tìm kiếm dựa trên sự tương tác và hợp tác giữa particles, PSO có thể giúp khám phá các vùng không gian tìm kiếm mà các thuật toán tối ưu hóa khác khó có thể khám phá được.

Tóm lại, thuật toán PSO là một phương pháp tìm kiếm và tối ưu hóa sử dụng bầy đàn particles. Nó đã được áp dụng thành công trong nhiều lĩnh vực và bài toán khác nhau, từ tối ưu hóa hàm số đơn giản đến tối ưu hóa trong các bài toán kỹ thuật phức tạp.

3.2.2 Giải thuật thuật toán PSO



Hình 3.1 Lưu đồ thuật toán PSO

Thuật toán PSO (Particle Swarm Optimization) là một thuật toán tối ưu hóa được lấy cảm hứng từ hành vi di cư của một đàn chim đang tìm kiếm thức ăn trong không gian. Thuật toán PSO bao gồm các bước sau đây:

- Bước 1: Khởi tạo quần thể (Population): Đầu tiên, ta cần khởi tạo một quần thể gồm N cá thể (Particle) trong không gian tìm kiếm. Mỗi cá thể sẽ đại diện cho một giải pháp trong không gian tìm kiếm.
- Bước 2: Khởi tạo vị trí (Position) vận tốc (Velocity) cho mỗi cá thể: Vị trí, vận tốc của mỗi cá thể được khởi tạo ngẫu nhiên trong không gian tìm

kiếm. Vận tốc này sẽ ảnh hưởng đến việc cá thể di chuyển trong không gian tìm kiếm.

- Bước 3: Đánh giá giá trị tương ứng với mỗi cá thể : Sử dụng hàm mục tiêu (objective function), ta tính giá trị tương ứng với mỗi cá thể trong quần thể. Giá trị này thường được sử dụng để đánh giá chất lượng của giải pháp tương ứng.
- Bước 4: Xác định vị trí tốt nhất của mỗi cá thể (local best): Với mỗi cá thể, ta lưu trữ vị trí tốt nhất mà nó đã đạt được trong quá khứ. Điều này giúp mỗi cá thể có khả năng tự khai thác các giải pháp tốt mà nó đã tìm thấy.
- Bước 5: Cập nhật vận tốc và vị trí của mỗi cá thể: Xác định vị trí tốt nhất của cả quần thể (global best): Ta cần xác định cá thể nào trong quần thể có giá trị tốt nhất (cao nhất hoặc thấp nhất, tùy thuộc vào bài toán). Vị trí tốt nhất này sẽ được lưu trữ và cập nhật theo thời gian. Dựa trên vị trí tốt nhất của cá thể hiện tại và vị trí tốt nhất của cả quần thể, ta cập nhật vận tốc và vị trí của mỗi cá thể theo các công thức tính toán tương ứng. Việc cập nhật này giúp các cá thể di chuyển trong không gian tìm kiếm.

Công thức cập nhật vận tốc mới của cá thể i tại thời điểm $t+1$ được tính bằng công thức sau:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (pbest_i(t) - x_i(t)) + c_2 \cdot r_2 \cdot (gbest(t) - x_i(t)) \quad (4.1)$$

Trong đó:

- $v_i(t)$ là vận tốc hiện tại của cá thể i .
- w là hệ số trọng số (weight) để điều chỉnh độ quan trọng của vận tốc hiện tại trong việc di chuyển.
- c_1 và c_2 là các hệ số tỷ lệ (acceleration coefficients) để điều chỉnh độ quan trọng của vận tốc dựa trên vị trí tốt nhất của cá thể và vị trí tốt nhất của cả quần thể.
- r_1 và r_2 là các số ngẫu nhiên trong khoảng từ 0 đến 1.
- $pbest_i(t)$ là vị trí tốt nhất mà cá thể i đã đạt được tại thời điểm t .
- $gbest(t)$ là vị trí tốt nhất mà cả quần thể đã đạt được tại thời điểm t .
- $x_i(t)$ là vị trí hiện tại của cá thể i .

Công thức cập nhật vị trí

Vị trí mới của cá thể i tại thời điểm $t+1$ được tính bằng công thức sau:

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4.2)$$

Trong đó:

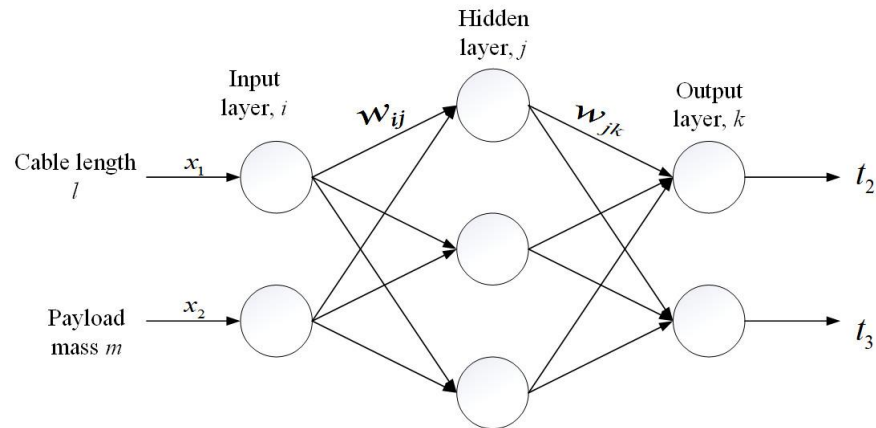
- $x_i(t)$ là vị trí hiện tại của cá thể i .
 - $v_i(t+1)$ là vận tốc mới của cá thể i đã được cập nhật.
- Bước 6: Kiểm tra tiêu chí dừng: Ta kiểm tra xem điều kiện dừng của thuật toán đã được đáp ứng chưa. Các tiêu chí dừng có thể là số lượng lặp lại đã

đạt, giá trị hàm mục tiêu đạt được đủ nhỏ, hoặc một tiêu chí khác tương ứng với bài toán cụ thể.

- Lặp lại các bước 3 đến 6 cho đến khi tiêu chí dừng được đáp ứng: Các bước từ 3 đến 6 được lặp lại cho đến khi tiêu chí dừng của thuật toán được đáp ứng. Sau đó trả về giải pháp tốt nhất: Sau khi thuật toán dừng, ta trả về giải pháp tốt nhất tìm được trong quần thể hoặc vị trí tốt nhất của cả quần thể tương ứng.

3.3 Thiết kế mạng neural

3.3.1 Xác định kiến trúc mạng



Hình 3.2 Kiến trúc mạng neural lựa chọn

Hình 3.6 mô tả cấu trúc của mạng nơ-ron nhân tạo (ANN) dựa trên mạng nơ-ron truyền thẳng được áp dụng trong công việc này. Cấu trúc này có dạng 2-3-2, gồm hai nơ-ron trong lớp đầu vào, ba nơ-ron trong lớp ẩn và hai nơ-ron trong lớp đầu ra. Mỗi nơ-ron trong các lớp được kết nối bởi một tham số được gọi là trọng số. Các chỉ số i , j và k đề cập đến các nơ-ron trong các lớp đầu vào, ẩn và đầu ra, tương ứng. Lớp đầu vào bao gồm các nơ-ron đầu vào ($x_i = 1, \dots, u$) chấp nhận tín hiệu đầu vào (độ dài cáp và khối lượng hàng hóa) trong thời gian thực và phân phối lại các tín hiệu này cho tất cả các nơ-ron trong lớp ẩn.

3.3.2 Lựa chọn hàm kích hoạt

Mục đích của việc thêm hàm kích hoạt là tăng tính phi tuyến cho hệ Neural. Hàm kích hoạt đóng vai trò quan trọng trong mạng nơ-ron bằng cách giới thiệu một bước tính toán bổ sung tại mỗi lớp trong quá trình truyền thuận.

Giả sử chúng ta có một mạng nơ-ron hoạt động mà không sử dụng các hàm kích hoạt. Trong trường hợp đó, mỗi nơ-ron sẽ chỉ thực hiện một phép biến đổi tuyến tính trên các đầu vào bằng cách sử dụng trọng số và độ lệch. Điều này xảy ra vì không quan trọng chúng ta gắn kết bao nhiêu lớp ẩn vào mạng nơ-ron, tất cả các lớp sẽ hoạt động theo cách tương tự vì tổ hợp của hai hàm tuyến tính là một hàm tuyến tính.

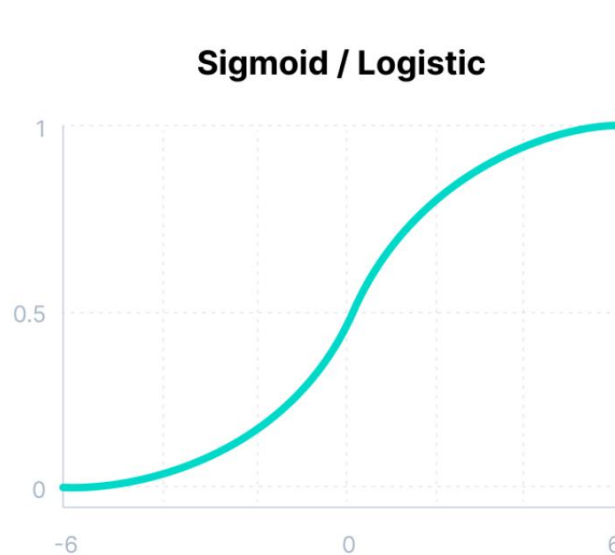
Mặc dù mạng nơ-ron trở nên đơn giản hơn, việc học các nhiệm vụ phức tạp là không thể, và mô hình của chúng ta chỉ là một mô hình hồi quy tuyến tính.

Bằng cách sử dụng các hàm kích hoạt, mạng nơ-ron có khả năng học và biểu diễn các mối quan hệ phi tuyến phức tạp hơn. Các hàm kích hoạt giúp giới hạn đầu ra của mỗi nơ-ron trong một khoảng giá trị cụ thể, tạo ra độ lớn và phạm vi đầu ra khác nhau. Điều này cho phép mạng nơ-ron tìm hiểu mối tương quan không tuyến tính giữa các đặc trưng và kết quả, và tạo ra các quyết định phức tạp dựa trên các dữ liệu đầu vào. Vì vậy, sử dụng các hàm kích hoạt là cần thiết để mạng nơ-ron có thể học và biểu diễn các mối quan hệ phi tuyến phức tạp, và trở nên mạnh mẽ hơn so với mô hình tuyến tính đơn giản.

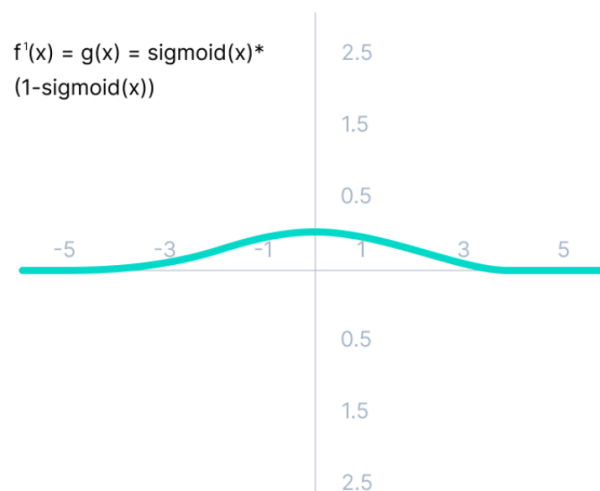
Các hàm kích hoạt phi tuyến thường gặp:

1. Hàm Sigmoid / Logistic: Hàm này chuyển đổi mọi giá trị thực thành giá trị nằm trong khoảng từ 0 đến 1. Hàm sigmoid được sử dụng phổ biến trong các mô hình dự đoán xác suất.

$$f(x) = \frac{1}{1 + e^{-x}}$$



Hình 3.3 Hàm sigmoid



Hình 3.4 Đạo hàm của hàm sigmoid

Ưu điểm của hàm sigmoid:

- Khoảng giá trị hữu ích: Hàm sigmoid giới hạn đầu ra trong khoảng từ 0 đến 1, làm cho nó phù hợp cho các mô hình mà chúng ta muốn dự đoán xác suất. Vì xác suất chỉ tồn tại trong khoảng này, hàm sigmoid là lựa chọn phù hợp để chuyển đổi đầu ra thành xác suất.
- Đạo hàm dễ tính: Đạo hàm của hàm sigmoid có thể tính toán dễ dàng và có công thức đơn giản là $f'(x) = \text{sigmoid}(x) * (1 - \text{sigmoid}(x))$. Điều này làm cho việc tối ưu hóa và huấn luyện mạng nơ-ron trở nên thuận tiện.
- Độ mịn và tính liên tục: Hàm sigmoid có đường cong mượt, không có nhảy nhót đột ngột trong giá trị đầu ra khi đầu vào thay đổi nhỏ. Điều này đảm bảo tính liên tục và mịn trong quá trình tính toán và lan truyền ngược trong mạng nơ-ron.

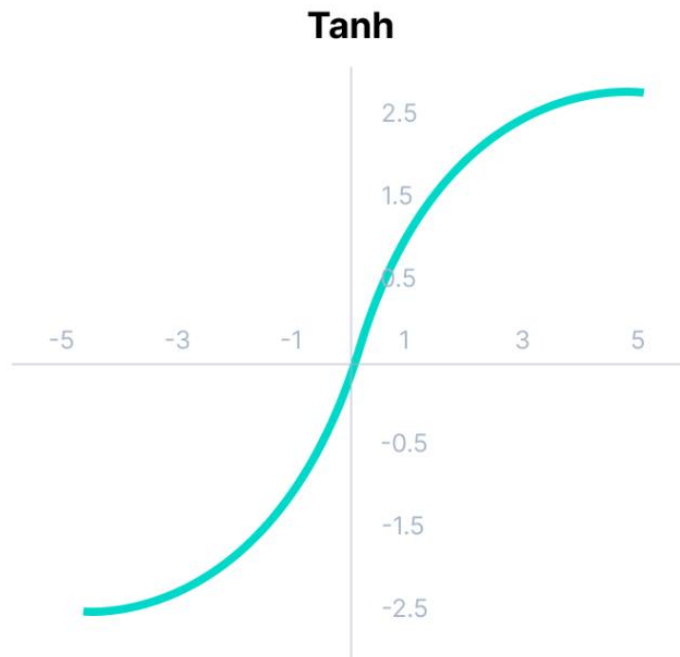
Nhược điểm của hàm sigmoid:

- Vanishing gradient: Một vấn đề phổ biến khi sử dụng hàm sigmoid là vấn đề vanishing gradient. Khi giá trị đầu vào vượt quá khoảng -3 đến 3, đạo hàm của hàm sigmoid rất nhỏ, gần tiến đến 0. Điều này làm cho quá trình lan truyền ngược trong mạng nơ-ron trở nên khó khăn và dẫn đến hiện tượng gradient biến mất, ảnh hưởng đến khả năng học của mạng.
- Không đối xứng xung quanh số không: Hàm sigmoid không đối xứng xung quanh số không, tức là đầu ra của tất cả các neuron sẽ cùng dấu. Điều này có thể làm cho quá trình huấn luyện mạng nơ-ron trở nên không ổn định và khó khăn hơn.
- Độ dốc không đồng đều: Hàm sigmoid có độ dốc lớn chỉ trong khoảng giá trị từ -3 đến 3. Khi đầu vào vượt qua khoảng này, độ dốc giảm đáng kể. Điều này có thể làm cho việc tối ưu hóa và huấn luyện mạng trở nên chậm hơn và không hiệu quả trong các vùng nằm ngoài khoảng giá trị này.

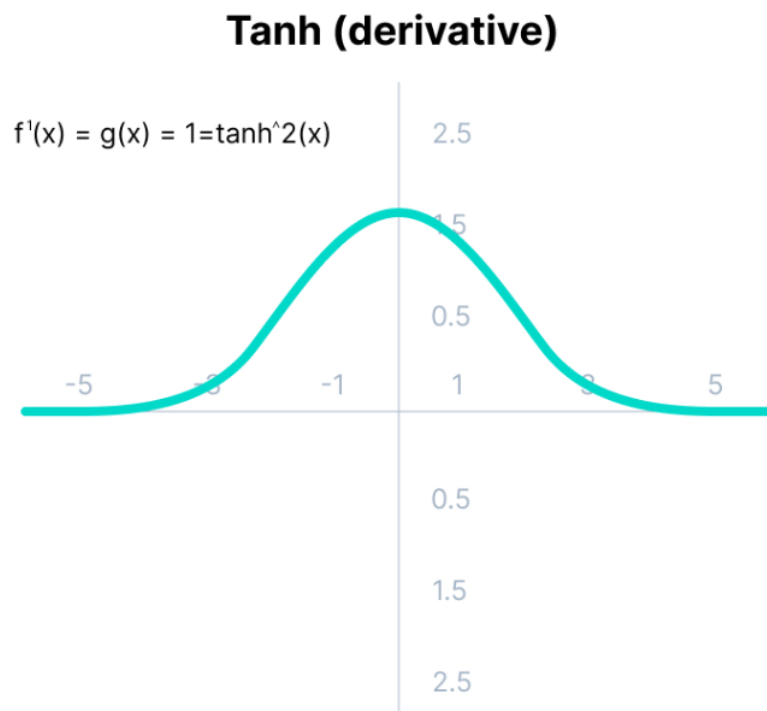
Tóm lại, mặc dù hàm sigmoid có những ưu điểm như giới hạn giá trị và tính liên tục, nhưng nó cũng gặp nhược điểm như vanishing gradient và không đối xứng xung quanh số không, làm cho nó không phù hợp cho một số ứng dụng trong mạng nơ-ron hiện đại.

2. Hàm Tanh: Hàm tanh tương tự như sigmoid, nhưng đầu ra nằm trong khoảng từ -1 đến 1. Nó được sử dụng trong các lớp ẩn của mạng nơ-ron. Hàm "tanh" (còn được gọi là hàm sinh sinh) là một hàm số được sử dụng trong mạng nơ-ron và có hình dạng tương tự như hàm sigmoid. Hàm tanh được định nghĩa như sau:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Hình 3.5 Hàm tank



Hình 3.6 Đạo hàm của hàm tanh

Ưu điểm của hàm tank:

- Khoảng giá trị rộng: Hàm tank giới hạn đầu ra trong khoảng từ -1 đến 1, giống như hàm sigmoid. Điều này làm cho nó hữu ích trong các mô hình cần đầu ra có giới hạn trong khoảng này.

- Đạo hàm dễ tính: Đạo hàm của hàm tanh cũng có công thức đơn giản, cho phép tính toán dễ dàng và thuận tiện cho việc tối ưu hóa và huấn luyện mạng nơ-ron.
- Tính đồng đều của độ dốc: Trong khoảng giá trị từ -3 đến 3, hàm tanh có độ dốc khá lớn và đồng đều. Điều này giúp cho việc tối ưu hóa mạng nơ-ron diễn ra hiệu quả và nhanh chóng trong khoảng giá trị này.
- Đối xứng xung quanh số 0: Khác với hàm sigmoid, hàm tanh là đối xứng xung quanh số 0. Điều này có nghĩa là đầu ra của các neuron có thể có cả giá trị dương và âm, làm cho quá trình huấn luyện mạng nơ-ron ổn định hơn và dễ dàng hơn.

Nhược điểm của hàm tanh:

- Không đảm bảo giới hạn giá trị trong khoảng 0 đến 1: Trong một số mô hình yêu cầu đầu ra trong khoảng từ 0 đến 1, hàm tanh không phù hợp và hàm sigmoid thường được ưu tiên sử dụng.
- Không đảm bảo tính liên tục và mịn: Mặc dù hàm tanh khá mượt và không có nhảy nhót đột ngột trong giá trị đầu ra, nhưng nó có thể không đảm bảo tính liên tục và mịn như hàm sigmoid.

Tóm lại, hàm tanh có nhiều ưu điểm như khoảng giá trị rộng, đạo hàm dễ tính và tính đồng đều của độ dốc. Tuy nhiên, nó cũng có nhược điểm như không đảm bảo giới hạn giá trị trong khoảng 0 đến 1 và không đảm bảo tính liên tục và mịn. Sự lựa chọn giữa hàm tanh và hàm sigmoid phụ thuộc vào yêu cầu cụ thể của mô hình và bài toán.

3. Hàm Parametric ELU được định nghĩa như sau

$$\begin{cases} x & \text{Với } x \geq 0 \\ \alpha (e^x - 1) & \text{Với } x < 0 \end{cases}$$

Ưu điểm của hàm ELU:

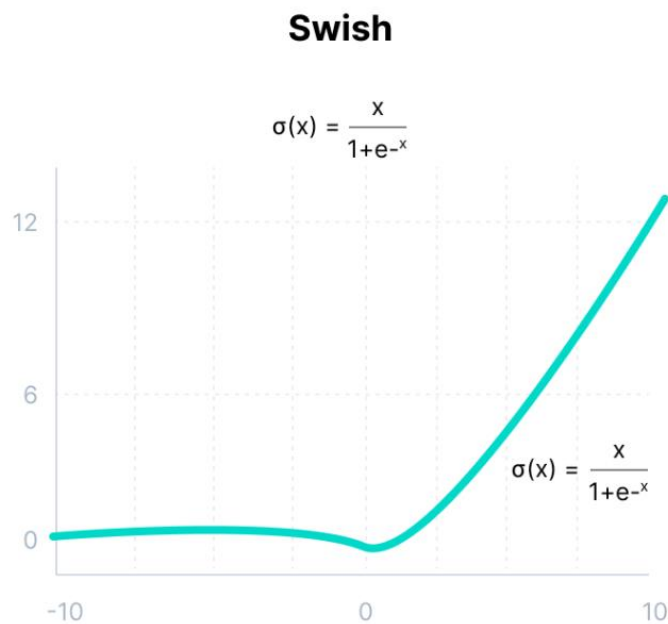
- Khoảng giá trị không bị giới hạn: Hàm ELU không có giới hạn trên khoảng giá trị đầu ra. Điều này cho phép nó mô hình hóa các mẫu có giá trị lớn hơn so với các hàm có giới hạn như hàm sigmoid hoặc hàm tanh.
- Giảm hiện tượng biến mất đạo hàm (Vanishing gradient): Hàm ELU giúp giảm tình trạng biến mất đạo hàm bằng cách sử dụng phần đường cong dương khi đầu vào là âm ($x < 0$). Điều này giúp tăng độ lớn của đạo hàm và khắc phục vấn đề của vanishing gradient, làm cho quá trình huấn luyện mạng nơ-ron hiệu quả hơn.
- Đạo hàm dễ tính: Đạo hàm của hàm ELU có công thức đơn giản và có thể tính toán dễ dàng. Điều này làm cho việc tối ưu hóa và huấn luyện mạng nơ-ron trở nên thuận tiện.
- Đối xứng xung quanh số 0: Tương tự như hàm tanh, hàm ELU là đối xứng xung quanh số 0. Điều này có nghĩa là đầu ra của các neuron

có thể có cả giá trị dương và âm, làm cho quá trình huấn luyện mạng nơ-ron ổn định hơn và dễ dàng hơn.

Nhược điểm của hàm ELU:

- Tính toán phức tạp: So với các hàm kích hoạt khác như hàm sigmoid hoặc hàm ReLU, tính toán hàm ELU có phần phức tạp hơn và đòi hỏi thêm thời gian tính toán.
- Không đảm bảo tính liên tục và mịn: Mặc dù hàm ELU giảm hiện tượng biến mất đạo hàm, nhưng nó không đảm bảo tính liên tục và mịn như hàm sigmoid.
- Tóm lại, hàm ELU có nhiều ưu điểm như khoảng giá trị không bị giới hạn, giảm hiện tượng biến mất đạo hàm và tính đối xứng xung quanh số 0. Tuy nhiên, nó cũng có nhược điểm như tính toán phức tạp hơn và không đảm bảo tính liên tục và mịn. Sự lựa chọn giữa hàm ELU và các hàm kích hoạt khác phụ thuộc vào yêu cầu cụ thể của mô hình và bài toán.

4. Hàm Swish: Đây là hàm kích hoạt tự điều chỉnh được phát triển bởi các nhà nghiên cứu tại Google. Nó tương thích tốt với hàm ReLU và thường cho hiệu suất tốt hơn trên các mạng nơ-ron sâu.



Hình 3.7 Hàm Swish

Ưu điểm của hàm Swish:

- Độ mượt và tính liên tục: Hàm Swish có đồ thị mượt và liên tục, tương tự như hàm sigmoid. Điều này giúp trong việc tối ưu hóa mạng nơ-ron và thuận tiện trong quá trình lan truyền ngược.

- Tính không bị giới hạn: Hàm Swish không có giới hạn về khoảng giá trị đầu ra, cho phép nó mô hình hóa các mẫu có giá trị lớn hơn so với các hàm có giới hạn như hàm sigmoid hoặc hàm tanh.
- Đạo hàm không bị chặn: Hàm Swish có đạo hàm không bị chặn và không bị biến mất như hàm sigmoid trong các vùng giá trị lớn hoặc nhỏ. Điều này giúp tránh hiện tượng vanishing gradient và cải thiện khả năng học của mạng nơ-ron.
- Tính phi tuyến: Hàm Swish là một hàm phi tuyến, cho phép mạng nơ-ron mô hình hóa các mối quan hệ phức tạp và khả năng học các đặc trưng phi tuyến của dữ liệu.

Nhược điểm của hàm Swish:

- Tính toán phức tạp: Tính toán hàm Swish phức tạp hơn so với các hàm kích hoạt khác như hàm sigmoid hoặc hàm ReLU. Điều này có thể yêu cầu thời gian tính toán và tài nguyên tính toán cao hơn.
- Thiếu sự tối ưu đối với một số tác vụ: Mặc dù hàm Swish đã cho thấy hiệu suất tốt trong nhiều tác vụ, nhưng nó không phải lúc nào cũng là hàm kích hoạt tốt nhất cho mọi tác vụ. Có thể tồn tại các tác vụ nơi một hàm kích hoạt khác như hàm ReLU hoặc hàm tanh hoạt động tốt hơn.

Tóm lại, hàm Swish có nhiều ưu điểm như tính mượt và liên tục, tính không bị giới hạn, đạo hàm không bị chặn và tính phi tuyến. Tuy nhiên, nó cũng có nhược điểm như tính toán phức tạp và không phù hợp với mọi tác vụ. Sự lựa chọn giữa hàm Swish và các hàm kích hoạt khác phụ thuộc vào yêu cầu cụ thể của mô hình và bài toán.

Mạng neural có thể khó huấn luyện vì hai vấn đề chính: độ mất mát gradient và độ mất mát gradient vô hạn. Vấn đề độ mất mát gradient xảy ra khi đạo hàm của hàm kích hoạt rất nhỏ, khiến việc cập nhật trọng số trở nên không hiệu quả. Vấn đề độ mất mát gradient vô hạn xảy ra khi đạo hàm trở nên quá lớn, dẫn đến việc tràn số và các giá trị không xác định. Vì vậy việc lựa chọn hàm kích hoạt (Activate function) là điều rất quan trọng.

Trong các hàm kích hoạt kể trên, mỗi loại đều có ưu nhược điểm và ứng dụng khác nhau. Việc lựa chọn hàm kích hoạt phụ thuộc vào dự đoán dạng output đầu ra của mạng neural. Trong đồ án này, $0 < t_2 < 1$, $0 < t_3 < 2$, dải dữ liệu đầu ra là giống nhau và bị chặn, cần sự phi tuyến giữa input và output. Chọn hàm Sigmoid

3.3.3 Chuẩn bị dữ liệu

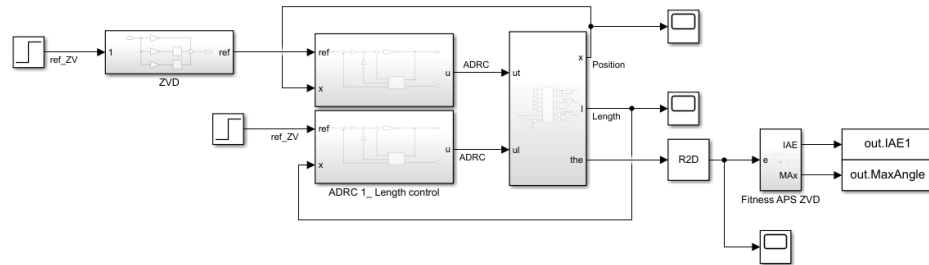
Thuật toán PSO được sử dụng thông qua mô phỏng để tìm ra các tham số tối ưu của bộ Input Shaping nhằm giảm góc dao động nhỏ nhất cho hệ thống. Chỉ số hiệu suất dựa trên sai số tuyệt đối tích phân (IAE) [2], $I(e)$, được xem như một hàm fitness, được xác định như sau:

$$J = I(e) = \int_0^{\infty} |\phi_d - \phi| dt$$

Trong đó:

ϕ_d : Góc giao động thực tế mà hệ nhận được

ϕ : Góc giao động mong muốn



Hình 3.8 Mô hình huấn luyện dữ liệu

Ứng với mỗi đầu vào l , m_c , mạng neural sẽ đưa ra kết quả hệ số t_2 và t_3 tương ứng. Trong đồ án, chuẩn bị tập mẫu huấn luyện cho mạng neural với l từ 0.2 m đến 0.65 m và m_c từ 0.8 kg đến 1.5 kg. Quá trình tối ưu hóa được thực hiện để tìm ra tất cả các bộ số tối ưu tham số bằng cách sử dụng mô phỏng trong MATLAB và kết quả được trình bày trong Bảng 3.1

Bảng 3.1 Dữ liệu huấn luyện được

| Sample No | Input 1 (l) | Input 2 (mc) | Output 1 (t1) | Output 2 (t2) | Sample No | Input 1 (l) | Input 2 (mc) | Output 1 (t1) | Output 2 (t2) |
|-----------|-------------|--------------|---------------|---------------|-----------|-------------|--------------|---------------|---------------|
| 1 | 0.2 | 0.8 | 0.443 | 0.897 | 41 | 0.2 | 1.2 | 0.453 | 0.890 |
| 2 | 0.25 | 0.8 | 0.495 | 1.009 | 42 | 0.25 | 1.2 | 0.499 | 0.995 |
| 3 | 0.3 | 0.8 | 0.541 | 1.090 | 43 | 0.3 | 1.2 | 0.539 | 1.100 |
| 4 | 0.35 | 0.8 | 0.598 | 1.189 | 44 | 0.35 | 1.2 | 0.589 | 1.182 |
| 5 | 0.4 | 0.8 | 0.631 | 1.268 | 45 | 0.4 | 1.2 | 0.624 | 1.268 |
| 6 | 0.45 | 0.8 | 0.677 | 1.343 | 46 | 0.45 | 1.2 | 0.680 | 1.341 |
| 7 | 0.5 | 0.8 | 0.705 | 1.428 | 47 | 0.5 | 1.2 | 0.708 | 1.410 |
| 8 | 0.55 | 0.8 | 0.747 | 1.482 | 48 | 0.55 | 1.2 | 0.733 | 1.493 |
| 9 | 0.6 | 0.8 | 0.772 | 1.561 | 49 | 0.6 | 1.2 | 0.785 | 1.562 |
| 10 | 0.65 | 0.8 | 0.812 | 1.614 | 50 | 0.65 | 1.2 | 0.803 | 1.626 |
| 11 | 0.2 | 0.9 | 0.453 | 0.903 | 51 | 0.2 | 1.3 | 0.457 | 0.907 |
| 12 | 0.25 | 0.9 | 0.500 | 1.009 | 52 | 0.25 | 1.3 | 0.511 | 1.008 |
| 13 | 0.3 | 0.9 | 0.542 | 1.102 | 53 | 0.3 | 1.3 | 0.548 | 1.089 |
| 14 | 0.35 | 0.9 | 0.599 | 1.182 | 54 | 0.35 | 1.3 | 0.603 | 1.196 |
| 15 | 0.4 | 0.9 | 0.625 | 1.270 | 55 | 0.4 | 1.3 | 0.642 | 1.263 |
| 16 | 0.45 | 0.9 | 0.676 | 1.337 | 56 | 0.45 | 1.3 | 0.674 | 1.339 |
| 17 | 0.5 | 0.9 | 0.714 | 1.412 | 57 | 0.5 | 1.3 | 0.704 | 1.426 |
| 18 | 0.55 | 0.9 | 0.751 | 1.488 | 58 | 0.55 | 1.3 | 0.738 | 1.477 |
| 19 | 0.6 | 0.9 | 0.779 | 1.551 | 59 | 0.6 | 1.3 | 0.780 | 1.562 |
| 20 | 0.65 | 0.9 | 0.812 | 1.609 | 60 | 0.65 | 1.3 | 0.818 | 1.609 |
| 21 | 0.2 | 1 | 0.445 | 0.896 | 61 | 0.2 | 1.4 | 0.454 | 0.897 |
| 22 | 0.25 | 1 | 0.495 | 1.002 | 62 | 0.25 | 1.4 | 0.498 | 1.005 |
| 23 | 0.3 | 1 | 0.545 | 1.105 | 63 | 0.3 | 1.4 | 0.542 | 1.093 |
| 24 | 0.35 | 1 | 0.585 | 1.196 | 64 | 0.35 | 1.4 | 0.595 | 1.194 |
| 25 | 0.4 | 1 | 0.644 | 1.264 | 65 | 0.4 | 1.4 | 0.642 | 1.259 |
| 26 | 0.45 | 1 | 0.682 | 1.343 | 66 | 0.45 | 1.4 | 0.675 | 1.340 |
| 27 | 0.5 | 1 | 0.712 | 1.420 | 67 | 0.5 | 1.4 | 0.711 | 1.422 |
| 28 | 0.55 | 1 | 0.743 | 1.477 | 68 | 0.55 | 1.4 | 0.735 | 1.485 |
| 29 | 0.6 | 1 | 0.771 | 1.557 | 69 | 0.6 | 1.4 | 0.785 | 1.560 |
| 30 | 0.65 | 1 | 0.811 | 1.620 | 70 | 0.65 | 1.4 | 0.799 | 1.622 |
| 31 | 0.2 | 1.1 | 0.454 | 0.905 | 71 | 0.2 | 1.5 | 0.442 | 0.899 |
| 32 | 0.25 | 1.1 | 0.511 | 0.998 | 72 | 0.25 | 1.5 | 0.499 | 1.001 |
| 33 | 0.3 | 1.1 | 0.541 | 1.094 | 73 | 0.3 | 1.5 | 0.542 | 1.108 |
| 34 | 0.35 | 1.1 | 0.603 | 1.194 | 74 | 0.35 | 1.5 | 0.583 | 1.185 |
| 35 | 0.4 | 1.1 | 0.644 | 1.273 | 75 | 0.4 | 1.5 | 0.624 | 1.262 |
| 36 | 0.45 | 1.1 | 0.675 | 1.350 | 76 | 0.45 | 1.5 | 0.675 | 1.345 |
| 37 | 0.5 | 1.1 | 0.710 | 1.418 | 77 | 0.5 | 1.5 | 0.699 | 1.410 |
| 38 | 0.55 | 1.1 | 0.750 | 1.490 | 78 | 0.55 | 1.5 | 0.745 | 1.485 |
| 39 | 0.6 | 1.1 | 0.770 | 1.554 | 79 | 0.6 | 1.5 | 0.772 | 1.558 |
| 40 | 0.65 | 1.1 | 0.811 | 1.625 | 80 | 0.65 | 1.5 | 0.814 | 1.616 |

Trong số các mẫu thu thập được, mẫu số 1,12,,23,34,45,56 chọn để kiểm chứng mạng neural. Các mẫu còn lại được sử dụng để huấn luyện cho mạng Neural

3.3.4 Tiến hành huấn luyện

Từ việc xác định kiến trúc mạng và lựa chọn hàm kích hoạt , theo hình 3.2 đầu ra thực tế của các lớp ẩn:

$$y_j = \text{Sigmoid} \left[\sum_{i=1}^u x_i w_{ij} - \theta_j \right]$$

Đầu ra thực tế lớp đầu ra:

$$y_k = \text{Sigmoid} \left[\sum_{j=1}^q x_{jk} w_{jk} - \theta_k \right]$$

Trong đó

- Sigmoid là hàm kích hoạt
- u là số lượng neural lớp đầu vào
- w_{ij}, θ_j là trọng số (weight) và độ lệch (bias) giữa lớp đầu vào và lớp ẩn
- w_{jk}, θ_k là trọng số (weight) và độ lệch (bias) giữa lớp ẩn và lớp đầu ra

Hàm mục tiêu là tổng bình phương sai lệch giữa đầu ra thực tế (Được lấy từ đầu ra mong muốn) và đầu ra mong muốn (Được lấy từ việc chuẩn bị dữ liệu) [2]

$$J = \frac{1}{2b} \sum_{s=1}^b \sum_{k=1}^p (y_{d_k}(s) - y_k(s))^2$$

Trong đó, b và p lần lượt là số lượng mẫu huấn luyện và số lượng neuron đầu ra. $y_{d_k}(s)$ và $y_k(s)$ là đầu ra mong muốn và thực tế cho mẫu thứ s tại neuron thứ k trong tầng đầu ra, tương ứng. Giá trị hàm mục tiêu (fitness) của mỗi hạt sẽ được tính toán trong mỗi lần lặp để cập nhật giá trị pb và gb. Giá trị thích nghi của mỗi hạt được so sánh với giá trị tốt nhất cá nhân đạt được cho đến thời điểm đó, trong đó giá trị thích nghi tốt nhất sẽ được gán làm giá trị tốt nhất cá nhân mới. Hơn nữa, giá trị thích nghi của mỗi hạt được so sánh với giá trị tốt nhất toàn cầu và giá trị tốt hơn sẽ trở thành giá trị tốt nhất toàn cầu mới. Vị trí của mỗi hạt được cập nhật bằng vận tốc hiện tại cho đến khi tất cả các hạt hội tụ và đạt được giải pháp tối ưu. Trong bài toán này, mục tiêu của PSO là huấn luyện mạng Neural sao cho các hạt (trọng số và độ lệch) được điều chỉnh để giảm thiểu giá trị thích nghi và phát triển bản đồ mạng cần thiết.

Các thông số ban đầu:

- $c1=c2=2$ trọng số trong công thức cập nhật vận tốc
- Số lượng cá thể trong quần thể: 50
- Số lượng mẫu: 74
- Số vòng lặp tối đa: 100
- $W_{max}=0.9$
- $W_{min}=0.4$

Dựa vào dữ liệu đã chuẩn bị ở mục 4.3.3, kiến trúc mạng, hàm kích hoạt (sigmoid), hàm fitness đã được trình bày ở trên đã tìm được các trọng số và độ lệch như sau:

Bảng 3.2 Trọng số và độ lệch huấn luyện được

| | | | | | | | | |
|--------|--------|---------|---------|---------|--------|--------|--------|--------|
| W11 | W12 | W13 | W21 | W22 | W23 | B11 | B12 | B13 |
| 3.5483 | 3.9885 | 3.2894 | -1.4864 | -0.6866 | 0.0537 | 1.8811 | 2.8082 | 2.0380 |
| W11 | W12 | W21 | W22 | W31 | W32 | B11 | B12 | |
| 3.1842 | 1.3176 | -5.6165 | -2.1100 | 3.1849 | 3.0650 | 2.2711 | 1.3325 | |

3.3.5 Đánh giá sử dụng mô hình

Bảng 3.2 biểu diễn t_2, t_3 là mẫu huấn luyện được và t'_2 và t'_3 là dữ liệu thông qua mạng neural. Để đánh giá độ hiệu quả khi sử dụng mô hình, sử dụng sai số phần trăm để đánh giá (giữa mẫu và dữ liệu qua bộ neural). Mục tiêu sai số $\delta \leq 5\%$. Kết quả được thể hiện ở bảng phía dưới:

Bảng 3.3 Đánh giá sử dụng mô hình

| No. | 1 | 12 | 23 | 34 | 45 | 56 |
|----------|--------|--------|--------|--------|--------|--------|
| t_2 | 0.4433 | 0.5003 | 0.5451 | 0.6031 | 0.6241 | 0.6735 |
| t'_2 | 0.4618 | 0.4976 | 0.5374 | 0.5809 | 0.6270 | 0.6743 |
| δ | 4.00% | -0.54% | -1.43% | -3.82% | 0.47% | 0.11% |
| t_3 | 0.8967 | 1.0085 | 1.1052 | 1.1942 | 1.2681 | 1.3390 |
| t'_3 | 0.9335 | 1.0070 | 1.0866 | 1.1705 | 1.2565 | 1.3417 |
| δ | 3.95% | -0.15% | -1.71% | -2.02% | -0.92% | 0.20% |

Kết luận: Dữ liệu được huấn luyện có độ chính xác cao, thỏa mãn yêu cầu đặt ra sai số $\delta \leq 5\%$.

3.4 Triển khai mô hình mạng Neural trên Matlab Simulink

Để đánh giá hiệu quả của chiến lược đề xuất, chúng ta sẽ so sánh nó với bộ điều khiển ADRC kết hợp với hai phương pháp Hình dạng đầu vào (Input Shaping - IS) dựa trên phương pháp trung bình của độ dài và tần số. Do không có giá trị chính xác, chúng ta sẽ đặt hệ số tắt dần cho các tham số tính toán của hai phương pháp này thành 0.

Tần số tự nhiên của con lắc đơn sẽ được tính toán bằng công thức $\omega = \sqrt{g/l}$. Đối với phương pháp đầu tiên, gọi là Hình dạng đầu vào Tần số Hoạt động Trung bình (Average Operating Frequency-Input Shaping - AOF-IS), chúng ta sẽ thiết kế định hình đầu vào dựa trên tần số trung bình, tức là giá trị trung bình của các tần số tương ứng với độ dài dây cáp lớn nhất và nhỏ nhất. Tần số tương ứng với độ dài dây cáp 0.2m là 7.0035 rad/s, trong khi đối với độ dài dây cáp 0.65m, tần số là 3.8848 rad/s. Lấy trung bình của hai tần số này, chúng ta có tần số trung bình là 5.4441 rad/s.

Phương pháp thứ hai, gọi là Hình dạng đầu vào Độ dài Hoạt động Trung bình (Average Operating Length-Input Shaping - AOL-IS), sẽ tính toán tần số tự nhiên dựa trên độ dài dây cáp 0.425m, là trung bình của độ dài ban đầu 0.65m và độ dài sau quá trình nâng hạ dây cáp 0.2m.

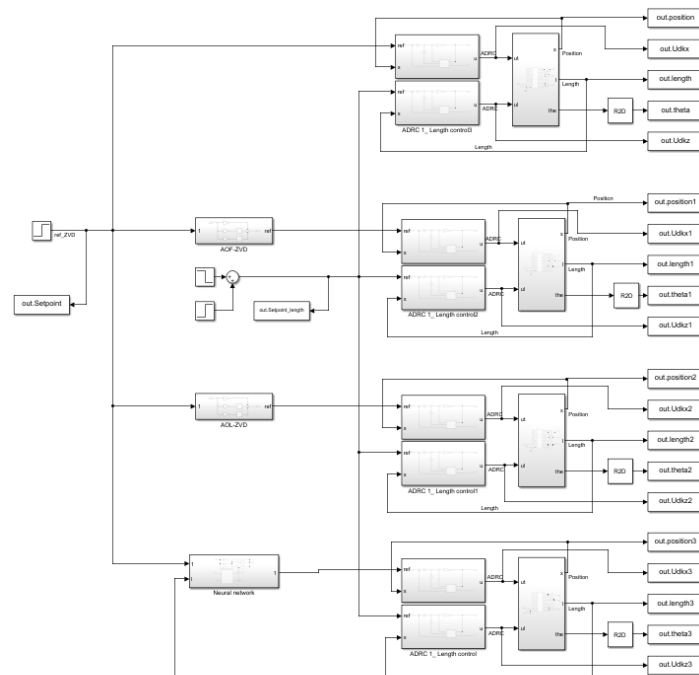
Các giá trị tham số của phương pháp AOF-IS và AOL-IS như sau:

Bảng 3.4 Thông số bộ lọc Input Shaping

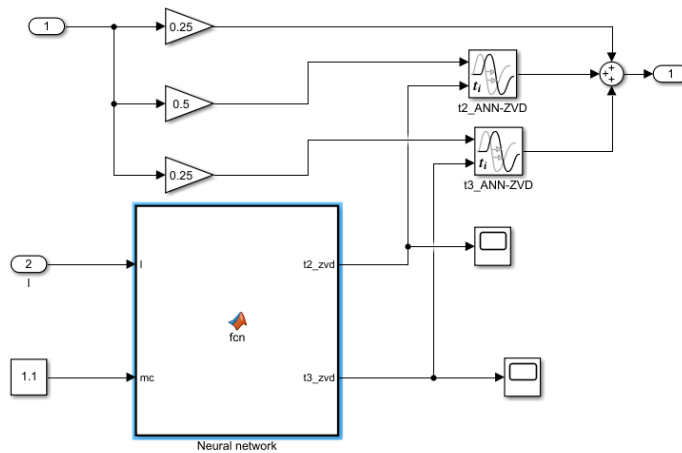
| Thông số | AOF-IS | AOL-IS |
|----------|--------|--------|
| A_1 | 0.25 | 0.25 |
| A_2 | 0.5 | 0.5 |
| A_3 | 0.25 | 0.25 |
| t_1 | 0 | 0 |
| t_2 | 0.5770 | 0.6538 |
| t_3 | 1.1541 | 1.3077 |

Trong Bảng đã chọn khối lượng tải trọng là $m_c=0.85$ kg, nằm trong phạm vi $[0.8, 1.5]$ kg, cho phép đánh giá độ bền của ADRC đối với các thay đổi về khối lượng tải trọng. Trường hợp thay đổi chiều dài dây khi cầu trục đang di chuyển

| Thông số bộ điều khiển | ADRC1 | ADRC2 |
|------------------------|----------|----------|
| b_0 | 0.21052 | 0.26315 |
| s_{CL} | -1.95 | -1.95 |
| K_P | 3.8025 | 3.8025 |
| K_D | 3.9 | 3.9 |
| s_{ESO} | -19.5 | -19.5 |
| l_1 | 58.5 | 58.5 |
| l_2 | 1140.75 | 1140.75 |
| l_3 | 7414.875 | 7414.875 |



Hình 3.9 Triển khai mô hình trên Simulink



Hình 3.10 Mô hình mạng Neural trên Matlab Simulink

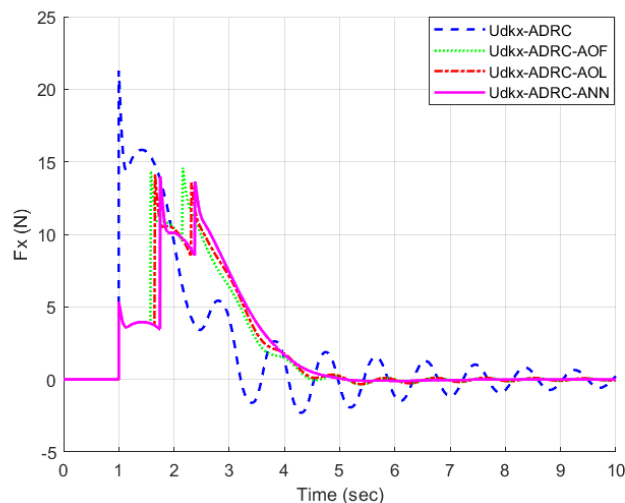
Có ba trường hợp sẽ được xem xét dựa trên chuyển động của tải trọng:

- Trường hợp 1: Nâng tải trọng lên.
- Trường hợp 2: Hạ tải trọng xuống.
- Trường hợp 3: Kiểm soát tải trọng để tránh vật cản.

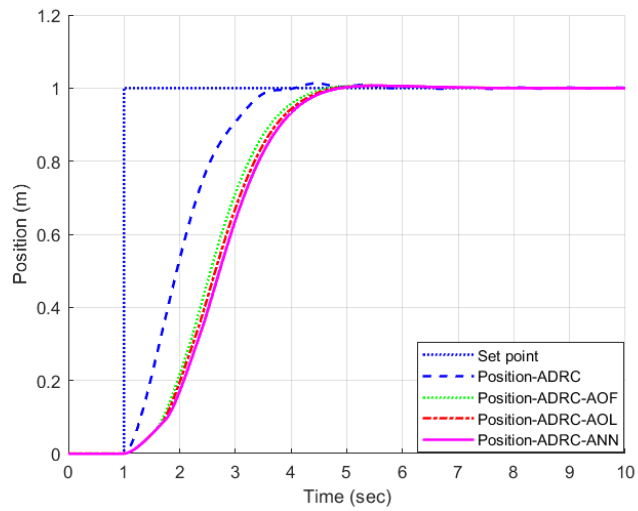
Trong trường hợp 1, mục tiêu chính là giảm độ dài dây ban đầu từ 0.65m xuống 0.2m. Trong trường hợp 2, việc điều khiển tải trọng hạ cánh sẽ được thực hiện từ vị trí khởi đầu, trong đó độ dài dây là 0.2m, đến một vị trí mới với độ dài dây là 0.65m. Trong trường hợp 3, có một chướng ngại vật xuất hiện trên đường đi của tải trọng, yêu cầu tải trọng được nâng lên để tránh nó. Ban đầu, tải trọng được đặt ở độ dài dây 0.65m và sau đó được nâng lên đến 0.2m trong một khoảng thời gian cụ thể. Sau đó, nó dần được hạ xuống lại vị trí ban đầu ở độ dài dây 0.65m.

Đối với việc kiểm soát chính xác, hai bộ điều khiển Active Disturbance Rejection Control (ADRC) được sử dụng đồng thời - một cho xe đẩy và một cho tải trọng. Cả hai cần đạt đến vị trí mục tiêu trong vòng 4 giây.

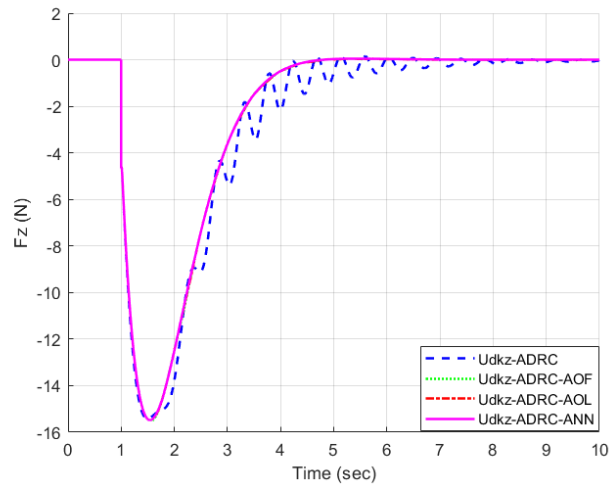
3.4.1 TH1 nâng tải



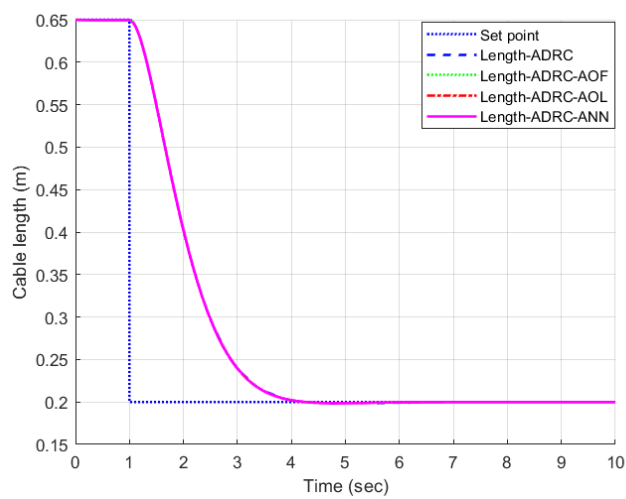
Hình 3.11 Đồ thị tín hiệu điều khiển trục x TH1



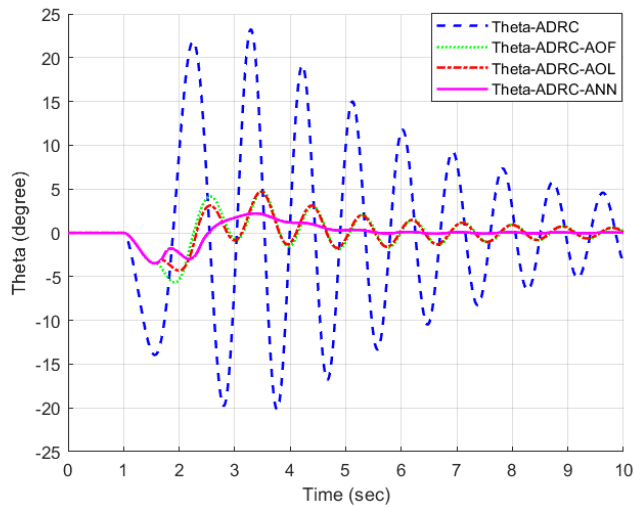
Hình 3.12 Đồ thị đáp ứng vị trí TH1



Hình 3.13 Đồ thị tín hiệu điều khiển trực z TH1



Hình 3.14 Đồ thị đáp ứng chiều dài dây TH1



Hình 3.15 Đồ thị đáp ứng góc lệch TH1

Trong đánh giá của Hình 3.12 và Hình 3.14, rõ ràng rằng cả bốn thuật toán điều khiển bao gồm bộ điều khiển ADRC, ADRC kết hợp với AOF-ZVD, ADRC kết hợp với AOL-ZVD và ADRC kết hợp với ANN-IS đều hiệu quả trong việc quản lý vị trí xe đẩy, chặt chẽ theo dõi các giá trị mục tiêu đã đặt. Thời gian ổn định cho mỗi phương pháp đều gần với giá trị mong muốn $T_{settle} = 3$ giây, trong khi độ quá điều chỉnh hạn duy trì ở mức khá nhỏ. Tuy nhiên do có bộ Input Shaping vào nên thời gian đáp ứng vị trí của các bộ Input Shaping (4s) lớn hơn so với bộ không có Input Shaping (3s)

Đánh giá hình 3.15, ta có bảng sau:

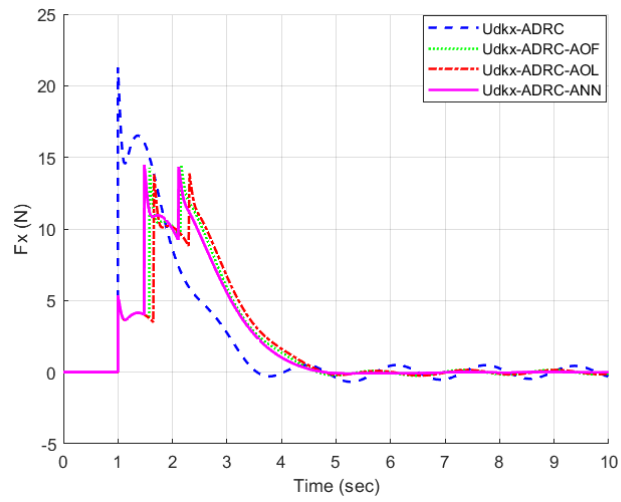
Bảng 3.5 Đáp ứng góc lắc ứng với các thuật toán điều khiển trong TH1

| Các trường hợp điều khiển | Góc lệch(Degree) |
|---------------------------|-------------------|
| ADRC | 23 |
| ADRC-AOF-IS | 1.50 |
| ADRC-AOL-IS | 1.45 |
| ADRC-ANN-IS | 0.11 |

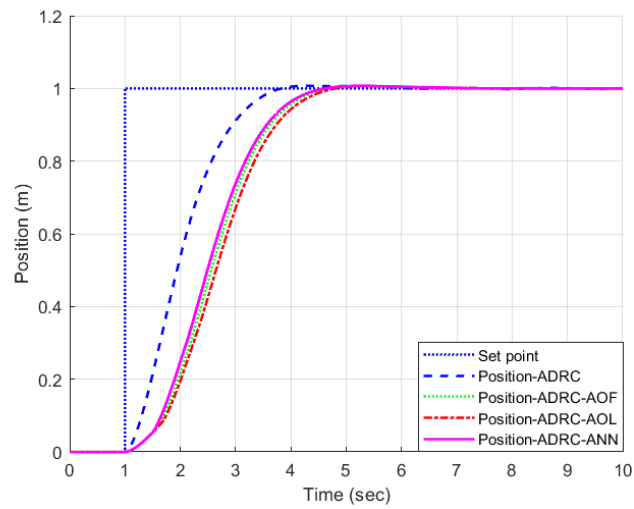
Góc lệch trong trường hợp thuật toán điều khiển ANN-IS giảm đáng kể so với các thuật toán khác trong trường hợp nâng tải trọng lên. Trong quá trình nâng tải, bộ ANN-IS thể hiện góc lắc cũng được thu hẹp một cách đáng kể so với các bộ còn lại

Đánh giá hình 3.11 và 3.13, tín hiệu điều khiển phù hợp với logic tín hiệu đặt đề ra. Với bộ điều khiển ADRC không có bộ lọc Input Shaping, do con lắc giao động rất nhiều gây ra lực lớn ảnh hưởng đến điều khiển, buộc bộ điều khiển phải đưa ra tín hiệu điều khiển đáp ứng. Vì vậy, phần tín hiệu điều khiển có tính thay đổi theo chu kỳ giao động con lắc.

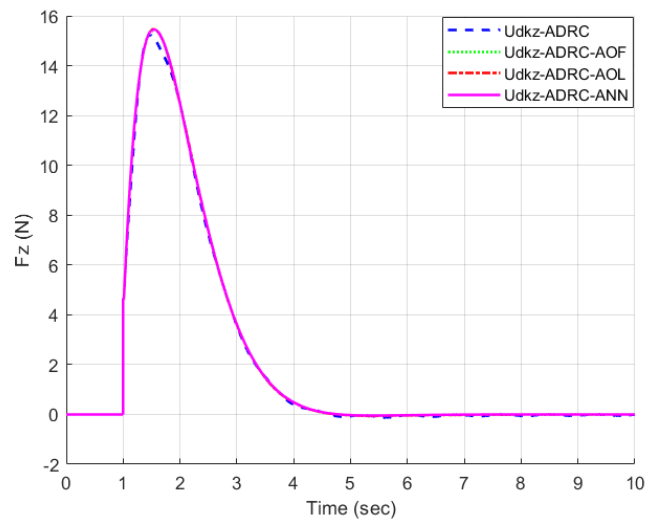
3.4.2 TH2 hạ tải



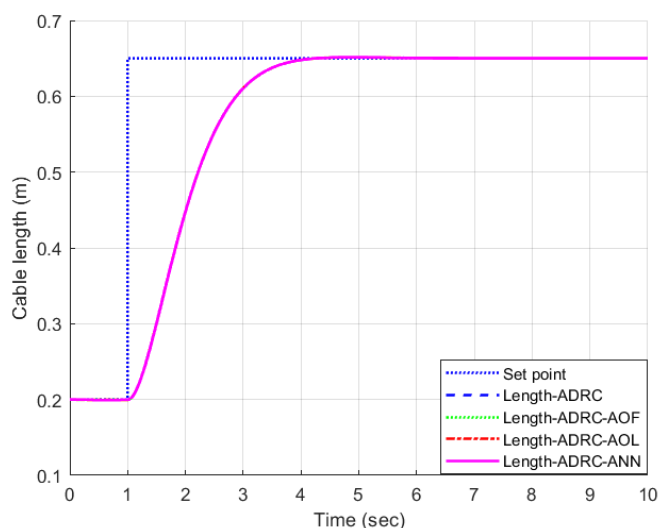
Hình 3.16 Đồ thị tín hiệu điều khiển vị trí TH2



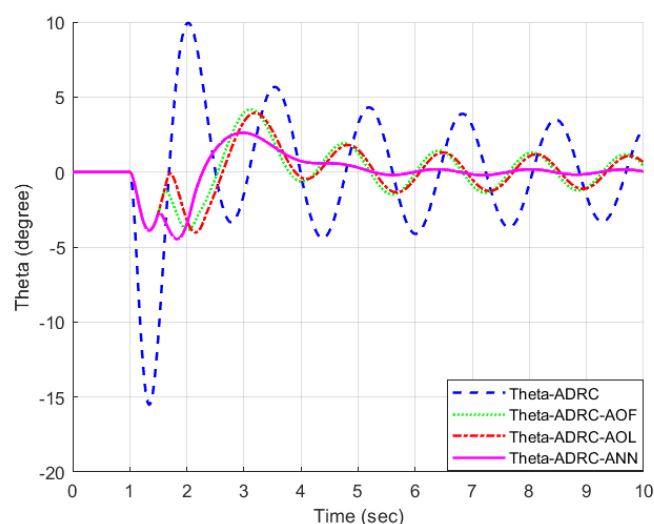
Hình 3.17 Đồ thị đáp ứng vị trí TH2



Hình 3.18 Đồ thị tín hiệu điều khiển chiều dài dây TH2



Hình 3.19 Đồ thị đáp ứng chiều dài dây TH2



Hình 3.20 Đồ thị đáp ứng góc lắc TH2

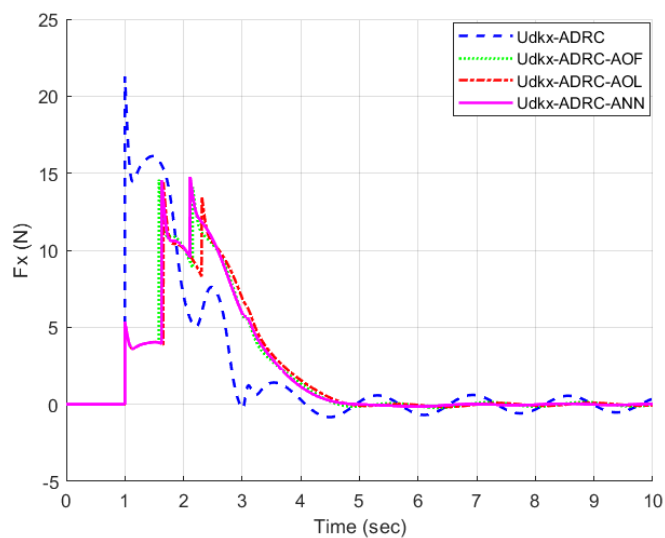
Trường hợp 2 hạ tải, đáp ứng góc được thể hiện ở bảng sau

Bảng 3.6 Đáp ứng góc lệch trong TH2

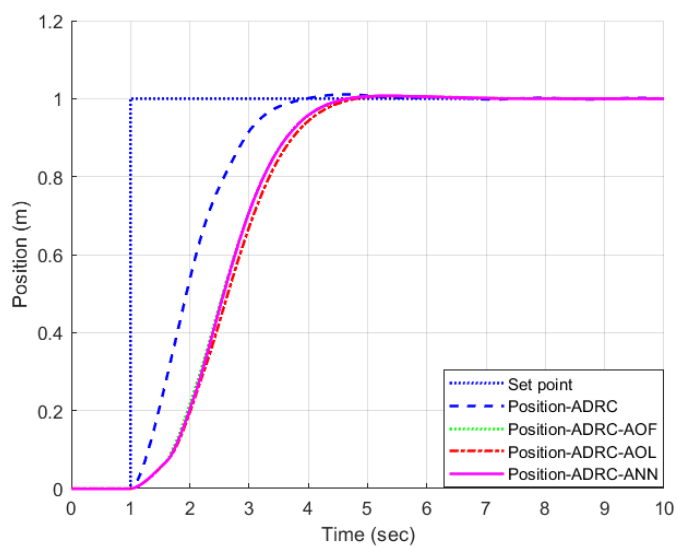
| Các trường hợp điều khiển | Góc lệch(Degree) |
|---------------------------|-------------------|
| ADRC | 6.69 |
| ADRC-AOF-IS | 1.41 |
| ADRC-AOL-IS | 1.30 |
| ADRC-ANN-IS | 0.18 |

Nhận xét: Đáp ứng vị trí và chiều dài dây thỏa mãn yêu cầu điều khiển. Độ quá điều khiển thấp, thời gian xác lập như mong muốn. Góc lệch trong trường hợp thuật toán điều khiển ANN-IS giảm đáng kể so với các thuật toán khác trong trường hợp hạ tải trọng.

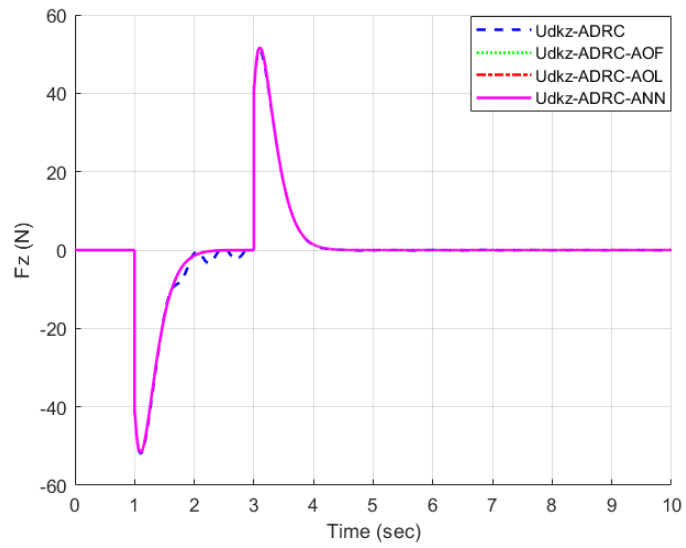
3.4.3 TH3 nâng hạ tải tránh vật cản



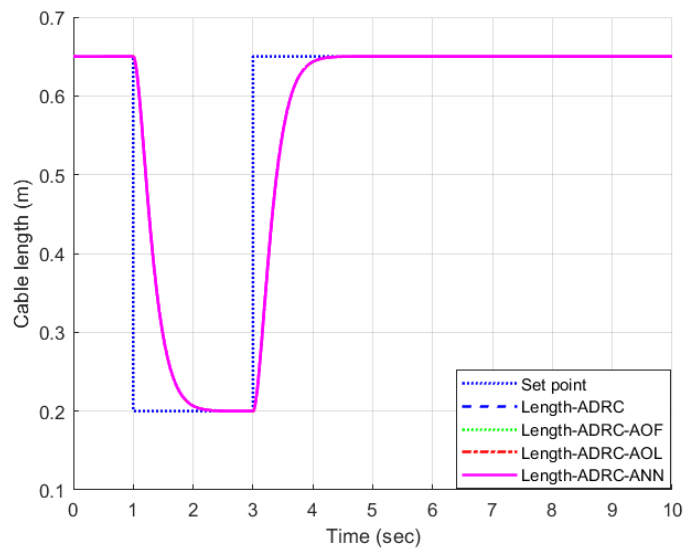
Hình 3.21 Đồ thị tín hiệu điều khiển trực x



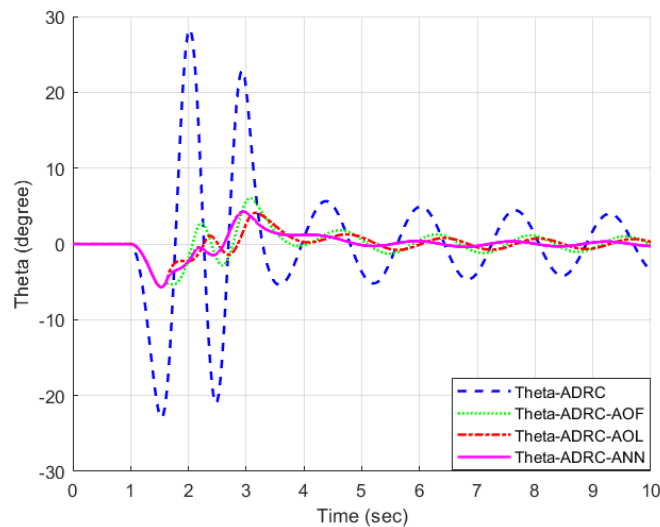
Hình 3.22 Đồ thị đáp ứng vị trí TH3



Hình 3.23 Đồ thị tín hiệu điều khiển trực z



Hình 3.24 Đồ thị đáp ứng chiều dài dây TH3



Hình 3.25 Đồ thị đáp ứng góc lắc TH3

Trường hợp 3, đáp ứng góc được thể hiện ở bảng sau

| Các trường hợp điều khiển | Góc lệch |
|---------------------------|----------|
| ADRC | 4.92 |
| ADRC-AOF-IS | 1.20 |
| ADRC-AOL-IS | 0.73 |
| ADRC-ANN-IS | 0.36 |

Nhận xét: Đáp ứng vị trí và chiều dài dây thỏa mãn yêu cầu điều khiển. Độ vọt lố thấp, thời gian xác lập như mong muốn. Góc lệch trong trường hợp thuật toán điều khiển ANN-IS giảm đáng kể so với các thuật toán khác trong trường hợp hạ tải trọng.

Kết luận: Trong cả 3 hợp, thuật toán điều khiển ứng dụng mạng neural mang lại hiệu quả chống rung lắc tốt hơn rất nhiều so với các bộ Input Shaping truyền thống

Kết luận chương 3: Chương 3 đã tiến hành xây dựng và triển khai hệ thống điều khiển chống rung cầu trục trên mô hình mô phỏng Matlab Simulink. Kết quả cho thấy chiến lược điều khiển sử dụng mạng Neural mang lại hiệu quả chống rung lắc đáng kể so với các bộ Input Shaping sử dụng nguyên lý trung bình

CHƯƠNG 4. TRIỂN KHAI TRÊN MÔ HÌNH THỰC NGHIỆM

Chương 4 giới thiệu tổng quan về hệ thống cầu trục thực nghiệm được dùng trong đồ án này. Nhận dạng hàm truyền động cơ bằng công cụ systems identification của matlab. Thiết kế bộ điều khiển và bộ Input Shaping cải tiến và tiến hành kiểm chứng kết quả thực nghiệm.

4.1 Giới thiệu mô hình cầu trục



Hình 4.1 Mô hình cầu trục thực nghiệm

Mô hình cầu trục gồm 2 phần chính:

- Khối điều khiển: gồm có giao diện, phần mềm lập trình, điều khiển trên laptop và bộ điều khiển NI-myRIO 1900
- Hệ thống truyền động của cầu trục: gồm cầu trục và tủ điện.

4.1.1 Mô hình cầu trục

Từ hình 4.1, mô hình cầu trục thực nghiệm được thiết kế bao gồm:

- Khung và dầm.
- Xe cầu.
- Động cơ điều khiển xe cầu.
- Động cơ nâng hạ tải.
- Cảm biến đo góc.
- Công tắc hành trình.
- Quả nặng thay tải.

Hệ thống cầu trục được thiết kế để thực hiện di chuyển theo hai trục XZ, trong đó chiều chuyển động của xe cầu theo trục X, và việc nâng hạ tải theo trục

Z. Xe cầu được thiết kế theo kiểu xe cầu chạy dưới. Tại hai đầu của hành trình trục X được bố trí công tắc hành trình, khi xe cầu đi hết hành trình và chạm vào công tắc hành trình thì xe cầu dừng lại đảm bảo an toàn cho cơ cấu truyền động cầu trục. Tải của cầu trục được thiết kế là dạng con lắc đơn. Chiều dài dây của móc đến được thay đổi khi di chuyển thông qua động cơ trục z

Khi di chuyển, móc dao động dẫn đến cơ cấu cảm biến thay đổi, làm điện áp phản hồi từ cảm biến thay đổi. Dựa trên điện áp phản hồi về mã góc dao động được xác định. Do hệ thống chưa có giải pháp thiết kế lắp đặt cảm biến để xác định góc dao động của tải nên trong quá trình thực hiện đề tài, ta chỉ khảo sát góc dao động của móc.

Hai trục X và Z của cầu trục được sử dụng động cơ servo của Mitsubishi HG-KN23J-S100. Các thông số cơ bản của động cơ như sau:

- Công suất động cơ nhỏ, công suất đầu ra 200W.
- Tốc độ quay: 3000 vòng/phút.
- Không có phanh điện từ.
- Encoder có độ phân giải 131072 xung/vòng

4.1.2 Tủ điện



Hình 4.2 Mặt ngoài của tủ điện

Cánh tủ điện được trang bị một loạt các nút nhấn và công tắc nhằm hỗ trợ người vận hành trong việc điều khiển cấp nguồn và thực hiện các hoạt động truyền động cho các trục của cầu trục. Hệ thống cũng bao gồm đèn báo pha và nút nhấn dừng khẩn cấp, đảm bảo an toàn trong quá trình vận hành



Hình 4.3 Mặt trong của tủ điện

Bên trong tủ điện, có các thiết bị đóng cắt và cấp nguồn phục vụ hệ thống cầu trục. Đáng chú ý trong số đó là hai bộ điều khiển servo Mitsubishi có mã MR-JE-20A, được sử dụng để điều khiển hai động cơ trục của cầu trục. Nhiệm vụ của hai bộ điều khiển này là nhận tín hiệu điều khiển từ bộ điều khiển trung tâm và xử lý để cấp đủ điện áp cho các động cơ tương ứng. Bên cạnh đó, chúng còn có khả năng đọc tín hiệu từ bộ mã hóa của động cơ.

Dưới đây là một số thông số cơ bản của driver servo Mitsubishi MR-JE-20A:

- Nguồn cấp: 200-240VAC
- Dòng điện đầu vào: 1.5A
- Dòng điện đầu ra: 1.5A
- Tần số hoạt động: 50/60Hz
- Công suất: 200W

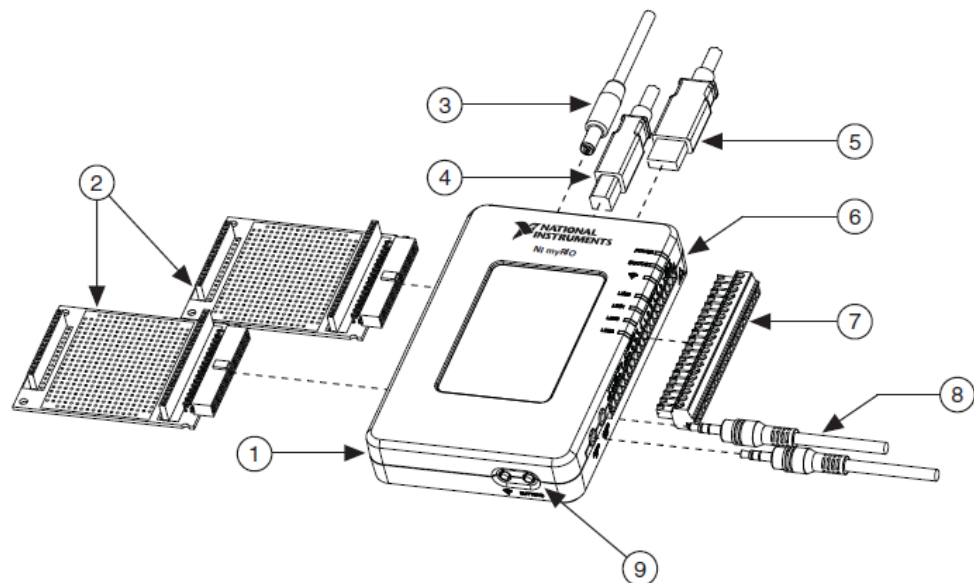
4.1.3 Bộ điều khiển NI-myRIO 1900



Hình 4.4 Bộ điều khiển NI-myRIO 1900

NI-myRIO là một công cụ vi xử lý do công ty National Instruments (NI) sản xuất, kèm với phần mềm lập trình LabVIEW. Được giới thiệu từ năm 1976, sau hơn 40 năm phát triển, NI-myRIO và LabVIEW đã thu hút sự quan tâm của nhiều nhà nghiên cứu và sinh viên trong lĩnh vực tự động hóa, robot và thiết bị thông minh trên toàn thế giới.

Thiết bị NI-myRIO là một nền tảng nhúng chạy độc lập với tích hợp phần cứng và phần mềm cho phép người dùng thiết kế và chế tạo hệ thống nhanh chóng. Nó tích hợp bộ vi xử lý ARM lõi kép và chip nhúng FPGA trên một hệ thống (System on a Chip). Được thiết kế đặc biệt cho việc giảng dạy và nghiên cứu kỹ thuật, NI-myRIO cũng có nhiều đầu vào ra (I/O), tích hợp WiFi và thiết kế vỏ rắn chắc.

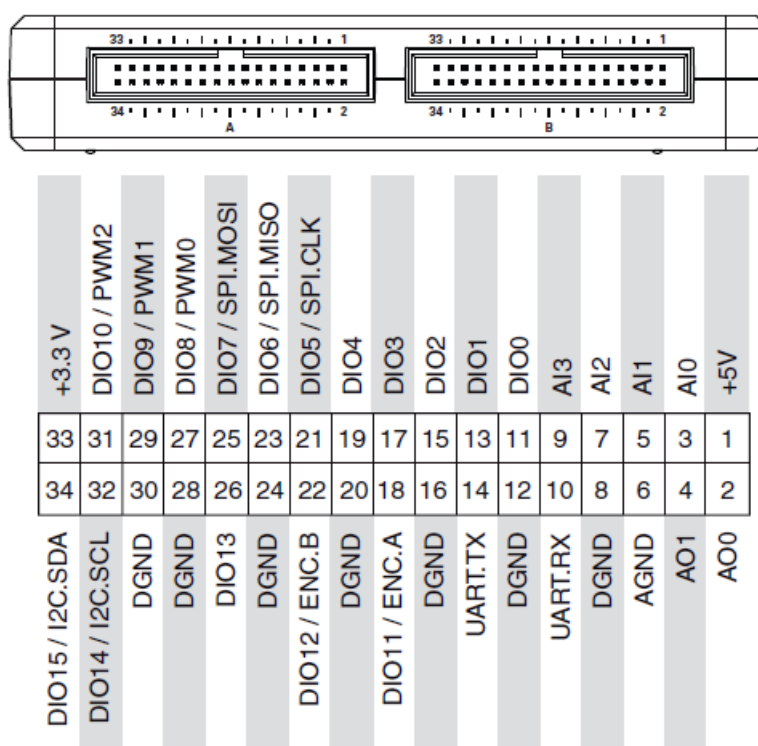


Hình 4.5 Cấu hình phần cứng của NI-myRIO 1900

Phần cứng của một bộ điều khiển NI-myRIO 1900 bao gồm 9 bộ phận chính như hình 4.3:

1. NI-myRIO 1900
2. Cổng mở rộng myRIO Expansion Port (MXP)
3. Cấp nguồn
4. Cấp kết nối USB
5. Cấp USB Host
6. Các đèn LED
7. Cổng Mini System Port (MSP)
8. Cấp vào/ra cổng âm thanh
9. Nút nhấn

NI-myRIO được trang bị 2 cổng mở rộng là MXP và MSP. Cổng mở rộng MXP có hỗ trợ 2 đầu nối là connector A và B, và chức năng của các chân I/O trên cả hai đầu nối này hoàn toàn giống nhau. Việc chọn I/O trên connector A hoặc B được quy định bởi phần mềm sử dụng.



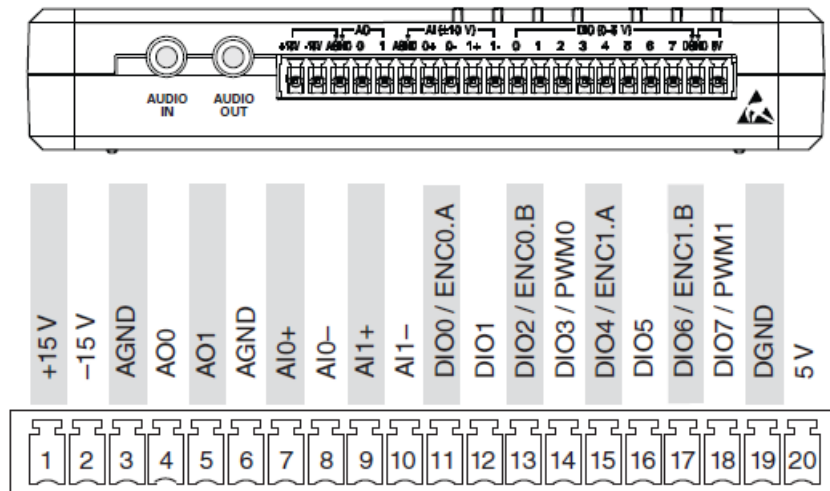
Hình 4.6 Sơ đồ I/O trên cổng mở rộng MXP

Chức năng của các chân được mô tả như sau:

Bảng 4.1 Bảng mô tả chức năng chân công MXP

| Chân | Loại tín hiệu | Mô tả |
|------------|---------------|--|
| +5V | Output | Nguồn ngõ ra +5 V. |
| AI <0...3> | Input | Ngõ vào tương tự (tham chiếu đất – single-ended). Điện áp từ 0-5V. |

| | | |
|-------------|-----------------|---|
| AO <0...1> | Output | Ngõ ra tương tự (tham chiếu đất – single-ended). Điện áp từ 0-5V. |
| AGND | N/A | Tham chiếu cho điện áp tương tự ngõ vào ngõ ra. |
| +3.3V | Output | Nguồn ngõ ra +3.3 V. |
| DIO<0...15> | Input or Output | Ngõ vào/ Ngõ ra số. Ngõ ra 3.3 V, ngõ vào 3.3 V/5 V. |
| UART.RX | Input | Nhận dữ liệu UART. |
| UART.TX | Output | Truyền dữ liệu UART. |
| DGND | N/A | Tham chiếu cho điện áp ngõ vào, ngõ ra số +5 V và +3.3 V. |



Hình 4.7 Sơ đồ I/O trên cổng mở rộng MSP

Bảng 4.2 Bảng mô tả chức năng chân cổng MSP

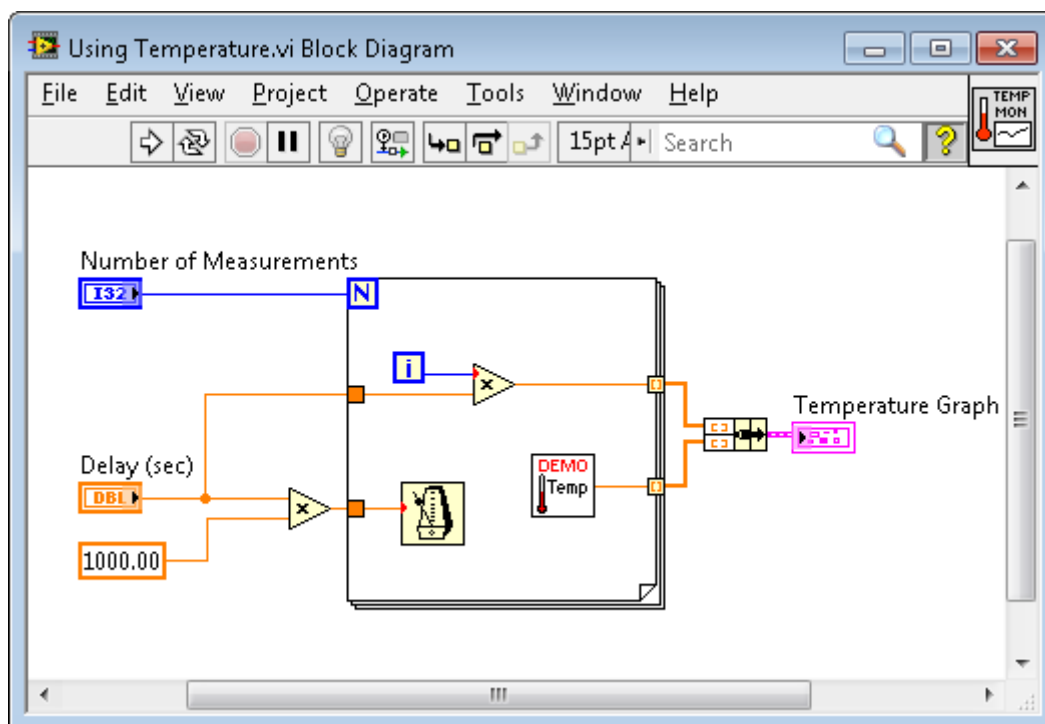
| Chân | Loại tín hiệu | Mô tả |
|-------------------------|-----------------|---|
| +15V/-15V | Output | Nguồn ngõ ra +15 V/-15 V. |
| AI0+/AI0-; AI1+/AI1- | Input | Ngõ vào tương tự vi sai ± 10 V. |
| AO <0...1> | Output | Ngõ ra tương tự vi sai ± 10 V. |
| AGND | N/A | Chân tham chiếu cho nguồn ngõ ra +15 V/-15 V và tương tự vào/tương tự ra. |
| +5V | Output | Nguồn ngõ ra +5 V. |
| DIO <0...7> | Input or Output | Ngõ vào/ Ngõ ra số. Ngõ ra 3.3 V, ngõ vào 3.3 V/5 V. |
| DGND | N/A | Tham chiếu cho điện áp ngõ vào, ngõ ra số +5 V và +3.3 V. |

4.1.4 Phần mềm LabVIEW

LabVIEW, viết tắt của Laboratory Virtual Instrument Engineering Workbench, là một phần mềm máy tính được phát triển bởi công ty National Instruments, Hoa Kỳ. Nó là một phần mềm chuyên dụng và thường được kèm theo

với thiết bị NI-myRIO. LabVIEW giúp người dùng triển khai phần mềm cho hệ thống một cách dễ dàng và có chức năng làm giao diện giám sát và vận hành HMI.

Ngôn ngữ lập trình sử dụng trong LabVIEW hoàn toàn khác so với các ngôn ngữ lập trình truyền thống như C, Pascal và các ngôn ngữ khác. Thay vì sử dụng mã lệnh, LabVIEW sử dụng hình ảnh trực quan để diễn đạt cú pháp trong môi trường soạn thảo, vì vậy nó thường được gọi là lập trình G (Graphical).



Hình 4.8 Giao diện lập trình trên LabVIEW

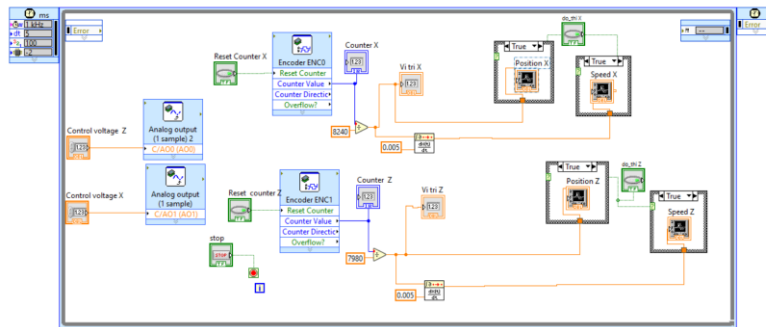
Một số chức năng chính của LabVIEW như sau:

- Thu thập tín hiệu từ các thiết bị ngoại vi.
- Truyền thông với các ngoại vi thông qua nhiều chuẩn giao tiếp thông qua các cổng truyền thông: RS232, RS485, USB, PCI, Ethernet.
- Mô phỏng và xử lý dữ liệu.
- Xây dựng giao diện người dùng nhanh chóng và thẩm mỹ hơn nhiều so với các ngôn ngữ khác.
- Cho phép thực hiện các thuật toán điều khiển như PID, Logic mờ một cách nhanh chóng thông qua các chức năng sẵn có trong LabVIEW.
- Dễ dàng tích hợp nhiều ngôn ngữ lập trình khác nhau trong chương trình như C, C++...

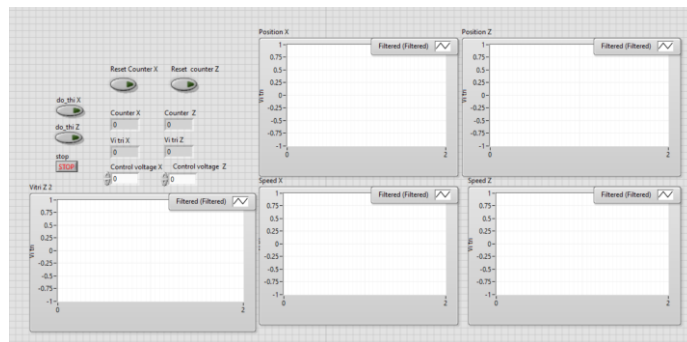
4.2 Nhận dạng hàm truyền mô hình

Trong đồ án này, mục tiêu là nhận dạng hàm truyền giữa điện áp điều khiển và vị trí của xe đẩy. Để làm được điều này, ta dựa vào hàm truyền giữa điện áp và tốc độ của xe đẩy. Phương pháp sử dụng là dựa trên đồ thị đáp ứng của đối tượng đối với một giá trị điện áp đầu vào cố định. Từ đó, ta xấp xỉ hàm truyền giữa điện áp và vị trí thành mô hình quán tính bậc nhất có dạng nhất định.

Khi có tín hiệu điều khiển, động cơ sẽ được driver cấp nguồn và di chuyển xe cầu. Tín hiệu phản hồi từ encoder sẽ được sử dụng để tính toán ra vận tốc của xe cầu. Từ công cụ nhận dạng đối tượng của matlab ta thu được hàm truyền liên hệ giữa vận tốc và U_{dk}.



Hình 4.9 Chương trình nhận dạng hàm truyền động cơ



Hình 4.10 Giao diện phần mềm nhận dạng hàm truyền động cơ

Ta có hàm truyền giữa vận tốc và điện áp điều khiển là khâu quán tính bậc nhất có dạng

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{Ts + 1} \quad (2.11)$$

Trong đó:

- K: hệ số khuếch đại tĩnh của đối tượng
- T: hằng số thời gian
- Y(s): tốc độ của động cơ, bằng đạo hàm giá trị của vị trí tại thời điểm đó
- U(s): điện áp cấp vào driver điều khiển động cơ.

Từ hàm truyền 4.1, ta có hàm truyền giữa vị trí và điện áp điều khiển có dạng:

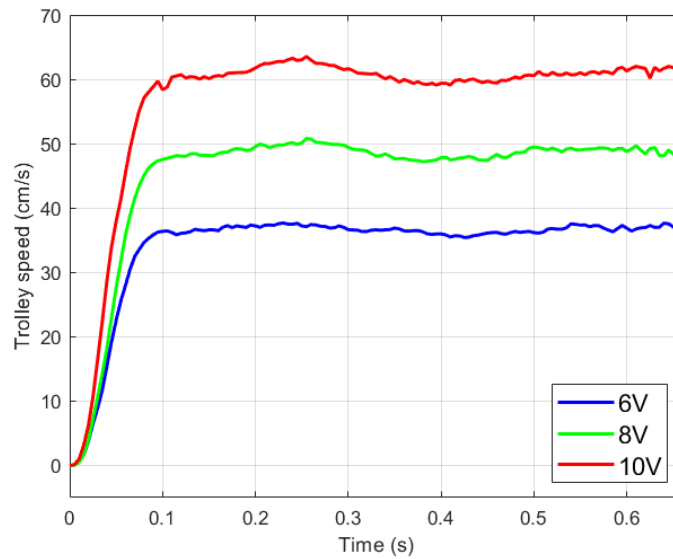
$$G(s) = \frac{Y(s)}{U(s)} = \frac{K}{s(Ts + 1)} \quad (2.12)$$

Trong đó:

- Y(s): vị trí xe cầu
- U(s): điện áp cấp vào driver điều khiển động cơ.

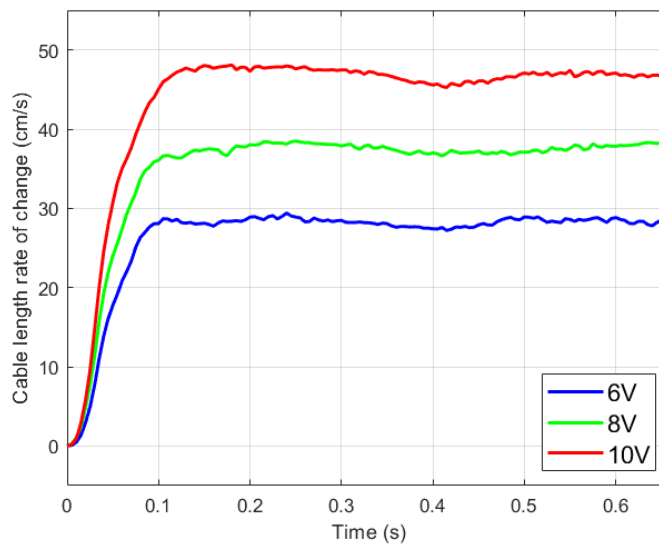
Sử dụng chức năng System identification trong Matlab sau khi chạy và thu thập dữ liệu dạng Excel, được hàm truyền giữa tín hiệu điều khiển và vị trí đối với 2 trục như sau

Trục x:



Hình 4.11 Đồ thị tốc độ động cơ trục x theo điện áp

Trục z:



Hình 4.12 Đồ thị tốc độ động cơ trục Z theo điện áp

Sử dụng chức năng System Identification trong Matlab sau khi chạy và thu thập dữ liệu dạng Excel, được hàm truyền giữa tín hiệu điều khiển và vị trí đối với 2 trục như sau

Trục x:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{6.139}{s(0.086s + 1)}$$

Trục z:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{5.897}{s(0.055s + 1)}$$

4.3 Thiết kế bộ điều khiển và bộ lọc cho mô hình thực nghiệm

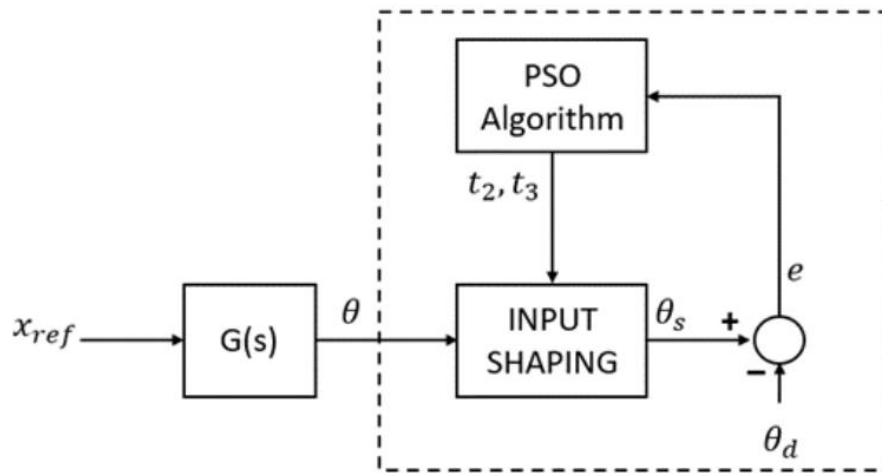
4.3.1 Thiết kế bộ điều khiển

Từ đó ta rút ra được hệ số $b_0 \approx \frac{K}{T}$ với $K = 6.139$, $T = 0.086$ đối với trục x và $K = 5.897$ và $T = 0.056$ là thông số từ hàm truyền nhận dạng. Áp dụng kết quả trên vào chạy mô hình thực nghiệm. Thông số bộ điều khiển ADRC

| Thông số | Giá trị BDK trục x | Giá trị BDK trục z |
|--------------|------------------------------------|------------------------------------|
| Setpoint | 40 (cm) | 20-70 (cm) |
| b_0 | $\frac{K}{T} \approx 80$ | $\frac{K}{T} \approx 105$ |
| T_{settle} | 3 (s) | 3 (s) |
| s_{cl} | $\frac{-5.85}{T_{settle}} = -1.95$ | $\frac{-5.85}{T_{settle}} = -1.95$ |
| K_{eso} | 100 | 100 |
| l | 20-70 (cm) | 20-70 (cm) |
| g | 9.81 (m/s ²) | 9.81 (m/s ²) |
| s^{ESO} | $100s_{cl} = -195$ | $100s_{cl} = -195$ |
| KD | $-2s_{cl} = 3.9$ | $-2s_{cl} = 3.9$ |
| KP | $s_{cl}^2 = 3.8025$ | $s_{cl}^2 = 3.8025$ |
| $l1$ | $-3s^{ESO} = 585$ | $-3s^{ESO} = 585$ |
| $l2$ | $3. (s^{ESO})^2 = 114075$ | $3. (s^{ESO})^2 = 114075$ |
| $l3$ | $-(s^{ESO})^3 = 7414875$ | $-(s^{ESO})^3 = 7414875$ |

4.3.2 Thiết kế bộ lọc

Đầu tiên, chúng ta sẽ sử dụng thuật toán PSO để tìm kiếm tập dữ liệu được gán nhãn. Theo [3], để áp dụng PSO trong trường hợp này, chúng ta sẽ đo đặc góc dao động của tải trọng với mỗi độ dài cáp cố định. Sau đó, góc dao động này sẽ được sử dụng để tìm các thông số tối ưu tương ứng t2 và t3 cho độ dài cáp đó bằng cách sử dụng PSO. Phương pháp tìm kiếm tập dữ liệu được thể hiện phía dưới



Hình 4.13: Phương pháp tìm kiếm dữ liệu trong thực nghiệm [3]

Các bước của phương pháp này như sau:

- Bước 1: $G(s)$ đại diện cho hệ thống vòng đóng bao gồm hai bộ điều khiển ADRC và mô hình cần cầu, với tín hiệu đầu vào là giá trị đặt điểm vị trí x_{ref} , và các thông số cố định là độ dài cáp 1 và khối lượng tải mp. Tín hiệu đầu ra là góc lắc của tải, được ghi lại.
- Bước 2: Góc θ sẽ được đưa vào hàm chuyển đổi của Input Shaping, và chúng ta thu được góc lắc được khống chế θ_s . Vì Input Shaping là một kỹ thuật cấp trước, việc thay đổi vị trí của nó sau hệ thống vòng đóng cho phép chúng ta kiểm tra khả năng khống chế rung của các thông số của nó.
- Bước 3: θ_s sau đó được trừ đi góc lắc mong muốn θ_d để thu được sai số góc lắc e .
- Bước 4: Tín hiệu e từ Bước 3 sẽ được sử dụng để tính toán hàm fitness $J(t)$, như được biểu diễn trong mục 3.3.3, từ đó cập nhật vị trí và vận tốc của các hạt theo thuật toán PSO.
- Bước 5: Lặp lại các bước 2-4 để tìm các thông số tối ưu t_2 và t_3 của Input Shaping cho khả năng khống chế rung tốt nhất cho độ dài cáp và khối lượng tải cụ thể.

Kết quả thu được như sau:

Bảng 4.3 Dữ liệu huấn luyện thực nghiệm

| No. | Input 1 (l) | Input 2 (mc) | Output 1 (t1)TN | Output 2 (t2)TN | No. | Input 1 (l) | Input 2 (mc) | Output 1 (t1)TN | Output 2 (t2)TN |
|-----|----------------|-----------------|--------------------|--------------------|-----|----------------|-----------------|--------------------|--------------------|
| 1 | 0.2 | 0.8 | 0.4402 | 0.8950 | 41 | 0.2 | 1.2 | 0.4490 | 0.8961 |
| 2 | 0.25 | 0.8 | 0.4916 | 1.0130 | 42 | 0.25 | 1.2 | 0.5068 | 0.9908 |
| 3 | 0.3 | 0.8 | 0.5436 | 1.0935 | 43 | 0.3 | 1.2 | 0.5376 | 1.1035 |
| 4 | 0.35 | 0.8 | 0.5931 | 1.1967 | 44 | 0.35 | 1.2 | 0.5876 | 1.1820 |
| 5 | 0.4 | 0.8 | 0.6375 | 1.2683 | 45 | 0.4 | 1.2 | 0.6181 | 1.2667 |
| 6 | 0.45 | 0.8 | 0.6729 | 1.3496 | 46 | 0.45 | 1.2 | 0.6863 | 1.3481 |
| 7 | 0.5 | 0.8 | 0.6984 | 1.4334 | 47 | 0.5 | 1.2 | 0.7101 | 1.4131 |
| 8 | 0.55 | 0.8 | 0.7426 | 1.4825 | 48 | 0.55 | 1.2 | 0.7329 | 1.4943 |
| 9 | 0.6 | 0.8 | 0.7669 | 1.5580 | 49 | 0.6 | 1.2 | 0.7814 | 1.5676 |
| 10 | 0.65 | 0.8 | 0.8152 | 1.6131 | 50 | 0.65 | 1.2 | 0.8033 | 1.6198 |
| 11 | 0.2 | 0.9 | 0.4565 | 0.8970 | 51 | 0.2 | 1.3 | 0.4608 | 0.9084 |
| 12 | 0.25 | 0.9 | 0.4947 | 1.0085 | 52 | 0.25 | 1.3 | 0.5064 | 1.0090 |
| 13 | 0.3 | 0.9 | 0.5407 | 1.1084 | 53 | 0.3 | 1.3 | 0.5483 | 1.0859 |
| 14 | 0.35 | 0.9 | 0.5912 | 1.1803 | 54 | 0.35 | 1.3 | 0.6057 | 1.1958 |
| 15 | 0.4 | 0.9 | 0.6237 | 1.2690 | 55 | 0.4 | 1.3 | 0.6357 | 1.2577 |
| 16 | 0.45 | 0.9 | 0.6777 | 1.3394 | 56 | 0.45 | 1.3 | 0.6810 | 1.3439 |
| 17 | 0.5 | 0.9 | 0.7181 | 1.4156 | 57 | 0.5 | 1.3 | 0.7074 | 1.4314 |
| 18 | 0.55 | 0.9 | 0.7557 | 1.4853 | 58 | 0.55 | 1.3 | 0.7461 | 1.4744 |
| 19 | 0.6 | 0.9 | 0.7806 | 1.5488 | 59 | 0.6 | 1.3 | 0.7772 | 1.5559 |
| 20 | 0.65 | 0.9 | 0.8186 | 1.6034 | 60 | 0.65 | 1.3 | 0.8139 | 1.6122 |
| 21 | 0.2 | 1 | 0.4529 | 0.8970 | 61 | 0.2 | 1.4 | 0.4561 | 0.9012 |
| 22 | 0.25 | 1 | 0.4986 | 1.0030 | 62 | 0.25 | 1.4 | 0.5017 | 0.9978 |
| 23 | 0.3 | 1 | 0.5494 | 1.1019 | 63 | 0.3 | 1.4 | 0.5356 | 1.0863 |
| 24 | 0.35 | 1 | 0.5862 | 1.1966 | 64 | 0.35 | 1.4 | 0.6031 | 1.1918 |
| 25 | 0.4 | 1 | 0.6431 | 1.2586 | 65 | 0.4 | 1.4 | 0.6401 | 1.2638 |
| 26 | 0.45 | 1 | 0.6894 | 1.3356 | 66 | 0.45 | 1.4 | 0.6732 | 1.3460 |
| 27 | 0.5 | 1 | 0.7175 | 1.4168 | 67 | 0.5 | 1.4 | 0.7095 | 1.4265 |
| 28 | 0.55 | 1 | 0.7389 | 1.4767 | 68 | 0.55 | 1.4 | 0.7327 | 1.4814 |
| 29 | 0.6 | 1 | 0.7718 | 1.5597 | 69 | 0.6 | 1.4 | 0.7848 | 1.5616 |
| 30 | 0.65 | 1 | 0.8146 | 1.6244 | 70 | 0.65 | 1.4 | 0.7928 | 1.6211 |
| 31 | 0.2 | 1.1 | 0.4602 | 0.9040 | 71 | 0.2 | 1.5 | 0.4499 | 0.8983 |
| 32 | 0.25 | 1.1 | 0.5045 | 0.9996 | 72 | 0.25 | 1.5 | 0.4987 | 0.9968 |
| 33 | 0.3 | 1.1 | 0.5463 | 1.0892 | 73 | 0.3 | 1.5 | 0.5430 | 1.1093 |
| 34 | 0.35 | 1.1 | 0.6083 | 1.1908 | 74 | 0.35 | 1.5 | 0.5897 | 1.1879 |
| 35 | 0.4 | 1.1 | 0.6429 | 1.2781 | 75 | 0.4 | 1.5 | 0.6315 | 1.2679 |
| 36 | 0.45 | 1.1 | 0.6804 | 1.3540 | 76 | 0.45 | 1.5 | 0.6781 | 1.3441 |
| 37 | 0.5 | 1.1 | 0.7098 | 1.4176 | 77 | 0.5 | 1.5 | 0.6971 | 1.4136 |
| 38 | 0.55 | 1.1 | 0.7546 | 1.4922 | 78 | 0.55 | 1.5 | 0.7423 | 1.4829 |
| 39 | 0.6 | 1.1 | 0.7616 | 1.5601 | 79 | 0.6 | 1.5 | 0.7772 | 1.5560 |
| 40 | 0.65 | 1.1 | 0.8137 | 1.6204 | 80 | 0.65 | 1.5 | 0.8127 | 1.6194 |

Tiến hành thiết kế mạng theo các bước ở mục 3.3, ta được trọng số cho bộ Neural

Bảng 4.4 Trọng số cho bộ Input Shaping cải tiến thực nghiệm

| W11 | W12 | W13 | W21 | W22 | W23 | B11 | B12 | B13 |
|---------|---------|---------|--------|--------|---------|--------|---------|--------|
| 3.9685 | 6.0278 | 11.4826 | 3.6256 | 7.1472 | -0.4378 | 0.5808 | -0.3182 | 4.0430 |
| W11 | W12 | W21 | W22 | W31 | W32 | B11 | B12 | |
| -0.8783 | -1.4610 | 0.3044 | 3.4662 | 0.4191 | 0.8289 | 0.0860 | 4.3253 | |

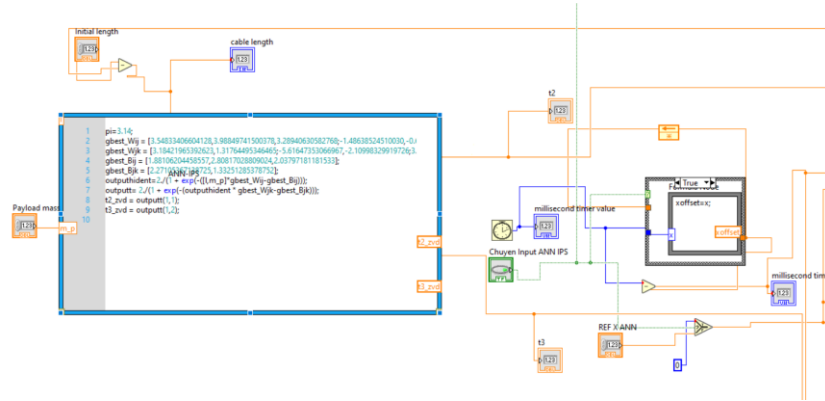
4.3.3 Kết quả chạy trên mô hình thực nghiệm

Có ba trường hợp sẽ được xem xét dựa trên chuyển động của tải trọng:

- Trường hợp 1: Nâng tải trọng lên.
- Trường hợp 2: Hạ tải trọng xuống.

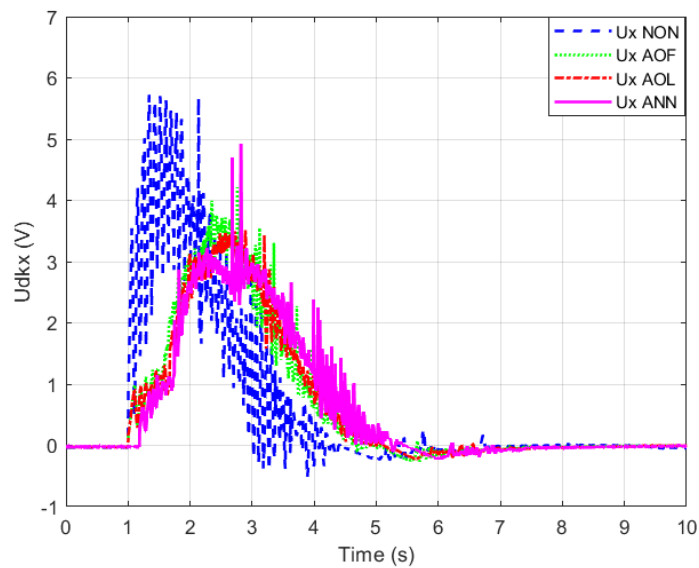
Trong trường hợp 1, mục tiêu chính là giảm độ dài dây ban đầu từ 0.7m xuống 0.2m. Trong trường hợp 2, việc điều khiển tải trọng hạ cánh sẽ được thực hiện từ vị trí khởi đầu, trong đó độ dài dây là 0.2m, đến một vị trí mới với độ dài dây là 0.7m.

Đối với việc kiểm soát chính xác, hai bộ điều khiển Active Disturbance Rejection Control (ADRC) được sử dụng đồng thời - một cho xe đẩy và một cho tải trọng. Cả hai cần đạt đến vị trí mục tiêu trong vòng 4 giây

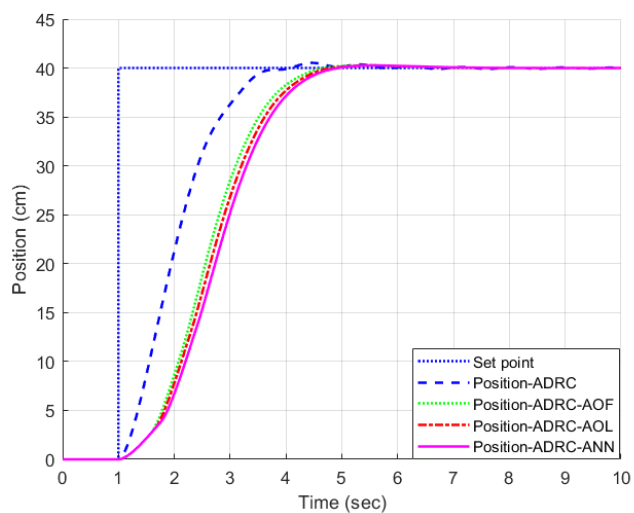


Hình 4.14 Bộ lọc Input Shaping cải tiến trên phần mềm Labview

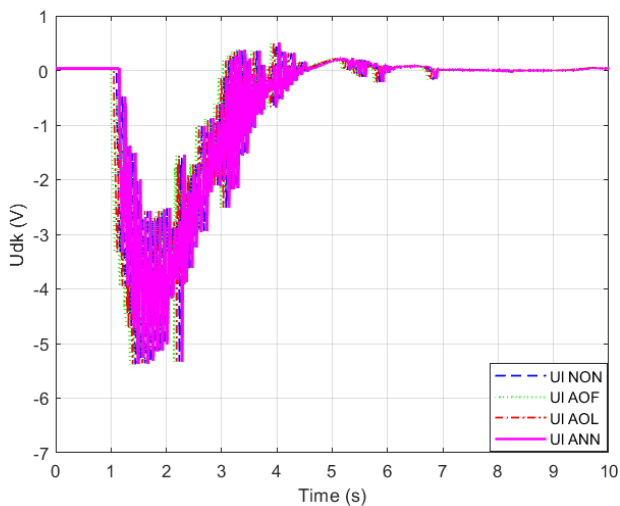
5.3.3.1 Trường hợp nâng tải



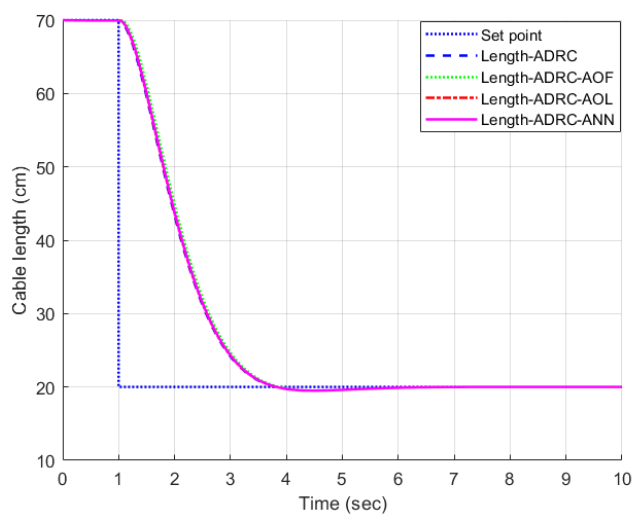
Hình 4.15 Điện áp điều khiển trực x (TH1 nâng tải)



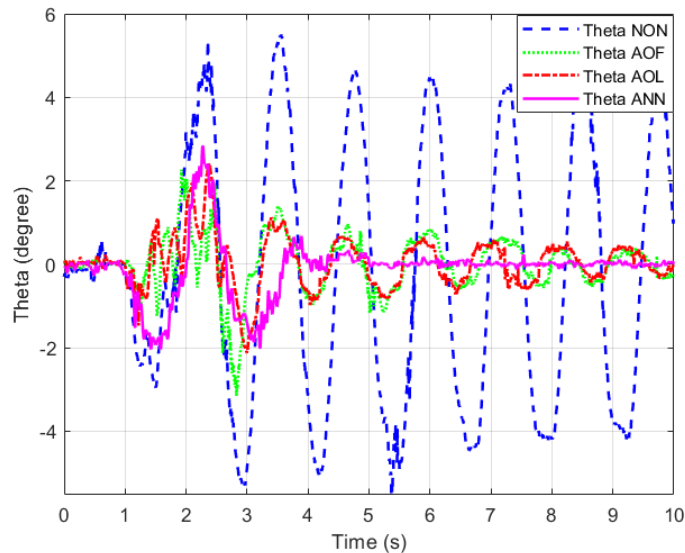
Hình 4.16 Đáp ứng vị trí trực x TH1



Hình 4.17 Điện áp điều khiển trực z TH1



Hình 4.18 Đáp ứng chiều dài dây trực z TH1



Hình 4.19 Đáp ứng góc lắc trường hợp nâng tải TH1

Trong đánh giá của Hình 4.16 và Hình 4.18, rõ ràng rằng cả bốn thuật toán điều khiển bao gồm bộ điều khiển ADRC, ADRC kết hợp với AOF-ZVD, ADRC kết hợp với AOL-ZVD và ADRC kết hợp với ANN-IS đều hiệu quả trong việc quản lý vị trí xe đẩy, chặt chẽ theo dõi các giá trị mục tiêu đã đặt. Thời gian ổn định cho mỗi phương pháp đều gần với giá trị mong muốn $T_{settle} = 3$ giây, trong khi độ quá điều chỉnh hạn duy trì ở mức khá nhỏ. Tuy nhiên do có bộ Input Shaping vào nên thời gian đáp ứng vị trí của các bộ Input Shaping (4s) lớn hơn so với bộ không có Input Shaping (3s)

Đánh giá hình 4.19, ta có bảng sau:

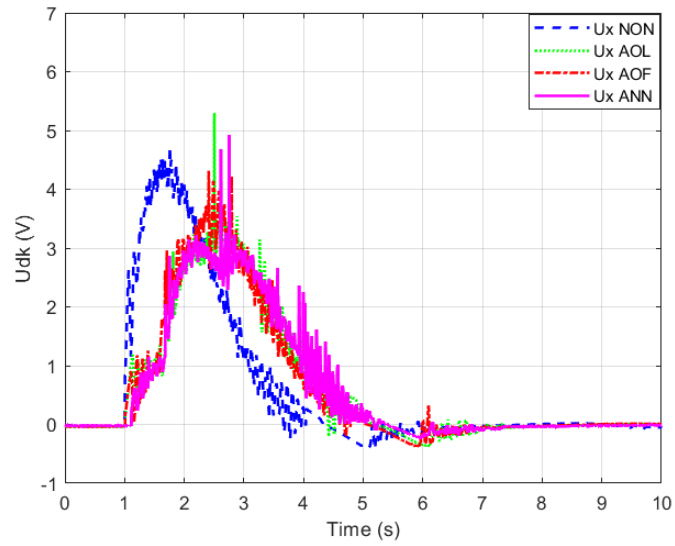
Bảng 4.5 Đáp ứng góc lắc TH1

| Các trường hợp điều khiển | Góc lệch(Degree) |
|---------------------------|-------------------|
| ADRC | 4.46 |
| ADRC-AOF-IS | 0.85 |
| ADRC-AOL-IS | 0.49 |
| ADRC-ANN-IS | 0.14 |

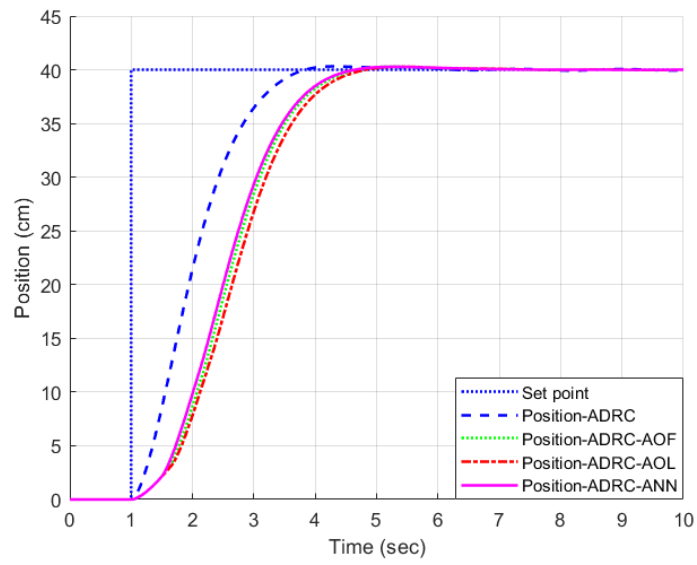
Góc lệch trong trường hợp thuật toán điều khiển ANN-IS giảm đáng kể so với các thuật toán khác trong trường hợp nâng tải trọng lên. Trong quá trình nâng tải, bộ ANN-IS thể hiện góc lắc cũng được thu hẹp một cách đáng kể so với các bộ còn lại.

Đánh giá hình 4.15 và 4.17 tín hiệu điều khiển phù hợp với logic tín hiệu vị trí đề ra. Bộ ANN-IS trên đồ thị vị trí và điện áp điều khiển đều xuất ra sau cùng so với các bộ còn lại..

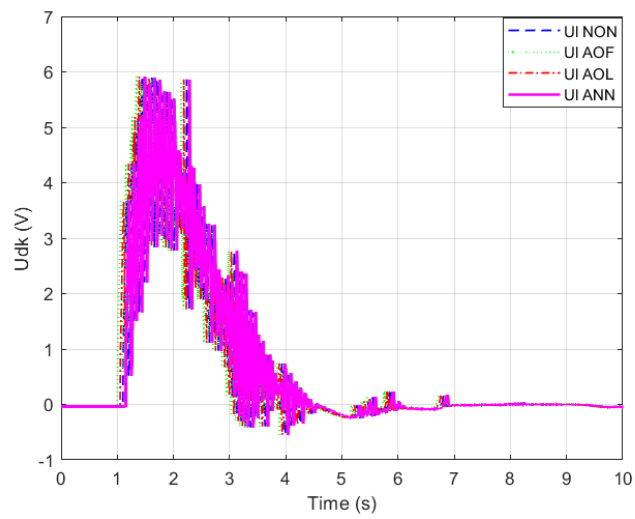
5.3.3.2 Trường hợp hạ tải



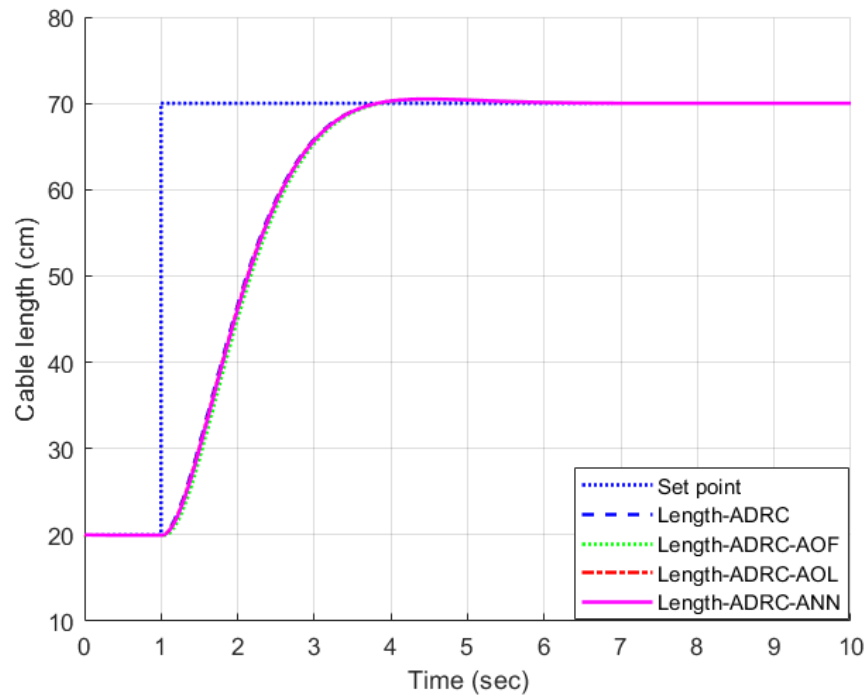
Hình 4.20 Điện áp điều khiển trục x TH2 (Hạ tải)



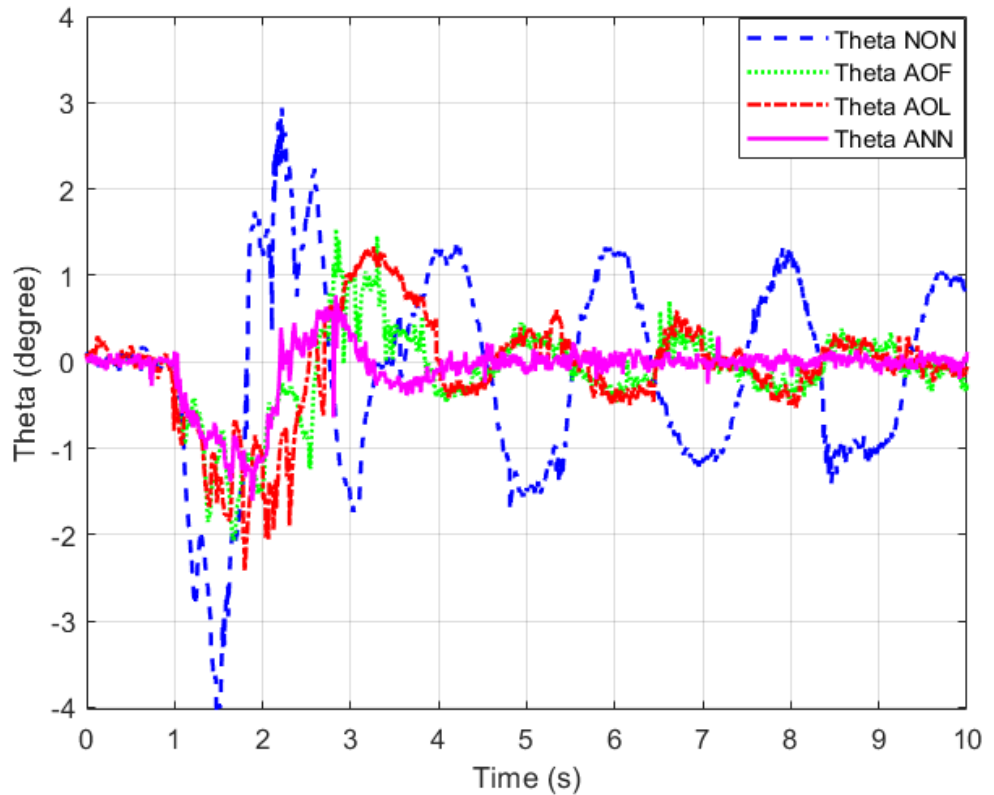
Hình 4.21 Đáp ứng vị trí trục x TH2



Hình 4.22 Điện áp điều khiển trục z TH2



Hình 4.23 Đáp ứng chiều dài dây trục z TH2



Hình 4.24 Đáp ứng góc lắc khi hạ tải TH2

Trong đánh giá của Hình 4.21 và Hình 4.23, rõ ràng rằng cả bốn thuật toán điều khiển bao gồm bộ điều khiển ADRC, ADRC kết hợp với AOF-ZVD, ADRC kết hợp với AOL-ZVD và ADRC kết hợp với ANN-IS đều hiệu quả trong việc quản

lý vị trí xe đẩy, chặt chẽ theo dõi các giá trị mục tiêu đã đặt. Độ quá điều chỉnh nhỏ, thời gian đáp ứng giống như ở trường hợp 1.

Đánh giá hình 4.24, ta có bảng sau:

Bảng 4.6 Đáp ứng góc lắc TH2

| Các trường hợp điều khiển | Góc lệch(Degree) |
|---------------------------|-------------------|
| ADRC | 4.2 |
| ADRC-AOF-IS | 0.448 |
| ADRC-AOL-IS | 0.36 |
| ADRC-ANN-IS | 0.085 |

Góc lệch trong trường hợp thuật toán điều khiển ANN-IS giảm đáng kể so với các thuật toán khác trong trường hợp hạ. Trong quá trình nâng tải, bộ ANN-IS thể hiện góc lắc cũng được thu hẹp một cách đáng kể so với các bộ còn lại.

Đánh giá hình 4.20 và 4.22 tín hiệu điều khiển phù hợp với logic tín hiệu đặt đề ra.

Kết luận: Trong cả hợp, thuật toán điều khiển ứng dụng mạng neural mang lại hiệu quả chống rung lắc tốt hơn rất nhiều so với các bộ Input Shaping truyền thống.

Kết luận chương 4: Chương 4 đã tiến hành xây dựng và triển khai hệ thống điều khiển chống rung cầu trục trên mô hình thực nghiệm. Phương pháp đề xuất ANN_ZVD đã được xác nhận vượt trội với mức giảm ít nhất 71.42% so với các bộ định hình ZVD được thiết kế bằng cách sử dụng tần số hoạt động trung bình và chiều dài hoạt động trung bình và 96.86% so với trường hợp không sử dụng bộ lọc đầu vào.

KẾT LUẬN

Sau một học kỳ nghiên cứu đề tài, em đã đạt được một phần của nội dung ban đầu đã đề ra. Kết quả của đồ án đã đáp ứng được yêu cầu trong phạm vi đề tài, mặc dù có một số sai lệch giữa lý thuyết và thực nghiệm. Em đã áp dụng khá nhiều kiến thức từ các môn học khác nhau, như đại số, giải tích, lý thuyết điều khiển, ... để xây dựng bài toán và giải quyết vấn đề trong đồ án.

Em xin gửi lời cảm ơn đến thầy giáo TS. Đỗ Trọng Hiếu và các thầy cô khoa Tự động hóa đã hướng dẫn và hỗ trợ tôi trong suốt thời gian thực hiện đề tài. Nhờ sự hướng dẫn, em đã đạt được những kết quả sau:

- Xây dựng mô hình toán học cho hệ thống cầu trục điều khiển hai trục.
- Tính toán và thiết kế bộ điều khiển ADRC kết hợp với bộ Input Shaping cải tiến điều khiển chống rung cầu trục khi chiều dài dây thay đổi.
- Thực hiện thử nghiệm với mô hình thực tế.

Em hy vọng đồ án này đóng góp một phần nhỏ vào việc nghiên cứu và ứng dụng trong lĩnh vực cầu trục và tự động hóa.

TÀI LIỆU THAM KHẢO

- [1] “Almutairi NB, Zribi M. Sliding Mode Control of a Three-dimensional Overhead Crane. *Journal of Vibration and Control*. 2009;15(11):1679-1730.”.
- [2] "Herbst, Gernot. "A simulative study on active disturbance rejection control (ADRC) as a control tool for practitioners." *Electronics* 2.3 (2013): 246-279."
- [3] "Ramli, Liyana, Z. Mohamed, and H. I. Jaafar. "A neural network-based input shaping for swing suppression of an overhead crane under payload hoisting and mass variations." *Mechanical Systems and Signal Processing* 107 (2018): 484-501."
- [4] "Kang, Chul-Goo, Rubayet Hassan, and Kyung-Youn Kim. "Analysis of a generalized ZVD shaper using impulse vectors." *International Journal of Control, Automation and Systems* 18 (2020): 2088-2094."