

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

LUẬN VĂN TỐT NGHIỆP

**THIẾT KẾ VÀ THỰC HIỆN BỘ ĐIỀU KHIỂN
THÍCH NGHI TỰ CHỈNH CHO ĐỘNG CƠ DC**

Giảng Viên Hướng Dẫn: PGS TS Huỳnh Thái Hoàng

Sinh Viên Thực Hiện: Nguyễn Tấn Khoa

MSSV: 40901245

Thành Phố Hồ Chí Minh, Tháng 12 Năm 2013

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA
Thành phố Hồ Chí Minh

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc
-----o0o-----

KHOA: ĐIỆN – ĐIỆN TỬ

NHIỆM VỤ LUẬN ÁN TỐT NGHIỆP

BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

HỌ VÀ TÊN: NGUYỄN TẤN KHOA

MSSV: 40901245

NGÀNH: KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

Đầu đề luận án

THIẾT KẾ VÀ THỰC HIỆN BỘ ĐIỀU KHIỂN THÍCH NGHI TỰ CHỈNH CHO ĐỘNG CƠ DC

1-Nhiệm vụ (yêu cầu về nội dung và số liệu ban đầu)

.....

3-Ngày giao nhiệm vụ luận án:.....

4-Ngày hoàn thành nhiệm vụ:

5-Họ và tên người hướng dẫn

Phản hướng dẫn

1.....

2.....

3.....

Nội dung và yêu cầu LATN đã được thông qua Bộ môn

Ngày.....tháng 12 năm 2013

Chủ nhiệm bộ môn

(Ký và ghi rõ họ tên)

Người hướng dẫn chính

(Ký và ghi rõ họ tên)

PHẦN DÀNH CHO KHOA, BỘ MÔN

Người duyệt (Chấm sơ bộ):.....

Đơn vị:.....

Ngày bảo vệ:

Điểm tổng kết:.....

Nơi lưu trữ của luận án:.....

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

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

Ngày ____ tháng ____ năm ____

PHIẾU CHẤM BẢO VỆ LVTN

(Dành cho người hướng dẫn)

1. Họ và tên SV: NGUYỄN TẤN KHOA MSSV: 40901245
Chuyên ngành: Kỹ thuật điều khiển & Tự động hóa
2. Đề tài:

THIẾT KẾ VÀ THỰC HIỆN BỘ ĐIỀU KHIỂN THÍCH NGHI TỰ CHỈNH CHO ĐỘNG CƠ DC

3. Họ và tên người hướng dẫn: PGS TS. HUỖNH THÁI HOÀNG

4. Tổng quát về bản thuyết minh:

Số trang:

Số chương:

Số bảng số liệu:

Số hình vẽ:

Số tài liệu tham khảo:

Phần mềm tính toán:

Sản phẩm:

5. Tổng quát các bản vẽ:

Số bản vẽ: bản A1 bản A2 khổ khác:

Số bản vẽ tay: Số bản vẽ máy tính:

6. Những ưu điểm chính của luận án tốt nghiệp:

.....
.....
.....
.....

7. Những khuyết điểm chính của luận án tốt nghiệp

.....
.....
.....
.....

8. Đề nghị: Được bảo vệ ☐ Bổ sung thêm để bảo vệ ☐ Không được bảo vệ ☐.

9. Đánh giá chung (bằng chữ: *giỏi, khá, trung bình*).....Điểm:/10

Kí tên (ghi rõ họ tên)

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA- XÃ HỘI- CHỦ NGHĨA-VIỆT NAM

Độc lập – Tự do – Hạnh phúc

Ngày ____ tháng ____ năm ____

PHIẾU CHẤM BẢO VỆ LVTN

(Dành cho người phản biện)

1. Họ và tên SV: NGUYỄN TẤN KHOA MSSV:40901245

Chuyên ngành: Kỹ thuật điều khiển & Tự động hóa

2. Đề tài:

THIẾT KẾ VÀ THỰC HIỆN BỘ ĐIỀU KHIỂN THÍCH NGHI TỰ CHỈNH CHO ĐỘNG CƠ DC

3. Họ và tên người phản biện:

4. Tổng quát về bản thuyết minh:

Số trang:

Số chương:

Số bảng số liệu:

Số hình vẽ:

Số tài liệu tham khảo:

Phần mềm tính toán:

Sản phẩm:

5. Tổng quát các bản vẽ:

Số bản vẽ: bản A1 bản A2 khổ khác:

Số bản vẽ tay: Số bản vẽ máy tính:

6. Những ưu điểm chính của luận án tốt nghiệp:

.....
.....
.....
.....

7. Những khuyết điểm chính của luận án tốt nghiệp

.....
.....
.....
.....

8. Đề nghị: Được bảo vệ ☐ Bổ sung thêm để bảo vệ ☐ Không được bảo vệ ☐.

9. Đánh giá chung (bằng chữ: *giỏi, khá, trung bình*) Điểm:/10

Kí tên (ghi rõ họ tên)

[illegible]

[illegible]

LỜI MỞ ĐẦU

Quá trình làm luận văn tuy dài nhưng vẫn không đủ để một mình tôi có thể tối ưu hóa bộ điều khiển tự chỉnh động cơ DC *một cách hoàn hảo nhất*, nên tôi quyết định dành một thời gian trong quá trình nghiên cứu để biên soạn tài liệu cũng như download rất nhiều báo cáo khoa học về vấn đề này tại các hội nghị trên thế giới. Tất cả mục đích đều *nhằm truyền lại cho thế hệ những sinh viên thế hệ tiếp theo* để họ có thể tiếp tục rút ngắn khoảng chênh lệch lớn về kỹ thuật điều khiển và tự động hóa của Việt Nam với thế giới.

Toàn bộ mã nguồn và tài liệu sẽ được cung cấp miễn phí trên trang web của câu lạc bộ nghiên cứu khoa học Pay It Forward là www.payitforward.edu.vn bởi tác giả với hi vọng bộ điều khiển tự chỉnh sẽ thay thế hoàn bộ điều khiển PID truyền thống tại Việt Nam cũng như đạt hiệu năng một cách tuyệt đối trong lĩnh vực điều khiển chính xác một ngày không xa.

Sinh viên

Nguyễn Tấn Khoa

LỜI CẢM ƠN

Đầu tiên tôi xin cảm ơn giảng viên hướng dẫn, thầy Huỳnh Thái Hoàng vì những giúp đỡ tận tình nhất trong quá trình làm đồ án 1, đồ án 2 và cho đến khi kết thúc luận văn. Tôi cũng xin cảm ơn bạn bè trong nhóm G-force và những thành viên trong câu lạc bộ Pay It Forward vì đã đồng viên tôi về mặt tinh thần cũng như hỗ trợ xây dựng mô hình luận văn.

Sinh viên

Nguyễn Tấn Khoa

TÓM TẮT

Luận văn trình bày quá trình thực hiện, ứng dụng thực tế và kết quả so sánh của bộ điều khiển động cơ DC tự chỉnh cải tiến cùng giải thuật ước lượng bình phương tối thiểu đệ quy cùng giải thuật điều khiển theo mô hình chuẩn nhúng trên nền 2 vi điều khiển ARM[®]Cortex[™]M4 là STM32F4 Discovery của hãng ST và TIVAF4123 Launchpad của hãng Texas Intrument.

Với bộ điều khiển tự chỉnh, bất kì động cơ DC nào cũng được điều khiển chính xác và hiệu quả hơn so với những bộ điều khiển khác, điển hình là bộ điều khiển PID truyền thống hơn 100 năm tuổi.

Mặt khác, việc tận dụng hiệu năng và giá thành hợp lý của vi điều khiển ARM[®]Cortex[™]M4 sẽ mang bộ điều khiển gần hơn với những người dùng phổ thông và những lập trình viên mà vẫn đảm bảo chất lượng của hệ thống.

MỤC LỤC

LUẬN VĂN TỐT NGHIỆP	i
PHIẾU CHẤM BẢO VỆ LVTN	iii
PHIẾU CHẤM BẢO VỆ LVTN	iv
NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN	v
NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN	vi
LỜI MỞ ĐẦU	vii
LỜI CẢM ƠN	viii
TÓM TẮT	ix
MỤC LỤC	x
DANH SÁCH BẢNG BIỂU, HÌNH VẼ, CÁC TỪ VIẾT TẮT	xiii
CHƯƠNG 1: GIỚI THIỆU	1
1.1 TỔNG QUAN VỀ BỘ ĐIỀU KHIỂN TỰ CHỈNH	1
1.2 TIỀM NĂNG, Ý NGHĨA KHOA HỌC VÀ ỨNG DỤNG.....	2
1.3 MỤC TIÊU LUẬN VĂN VÀ PHƯƠNG PHÁP NGHIÊN CỨU	4
1.4 SƠ LƯỢC NỘI DUNG LUẬN VĂN	5
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	6
2.1 GIỚI THIỆU CHUNG VỀ HAI BỘ ĐIỀU KHIỂN PID VÀ STR.....	6
2.2 GIẢI THÍCH CHỨC NĂNG CÁC KHỐI TRONG GIẢI THUẬT STR	7
2.3 MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VỊ TRÍ VÀ VẬN TỐC ĐỘNG CƠ DC	8
2.3.1 MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VẬN TỐC ĐỘNG CƠ DC.....	8
2.3.2 MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VỊ TRÍ ĐỘNG CƠ DC.....	11
2.4 LÝ THUYẾT THIẾT KẾ BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN	12
2.5 CƠ SỞ TOÁN HỌC VÀ Ý NGHĨA THỰC TIỄN CỦA PHÉP ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU	13
2.6 PHƯƠNG PHÁP THIẾT KẾ LUẬT ĐIỀU KHIỂN THEO MÔ HÌNH MẪU	16
CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG	17
3.1 BỘ VI XỬ LÝ.....	18
3.1.1 VI ĐIỀU KHIỂN STM32F407 TRÊN KIT STM32F4 DISCOVERY	20
3.1.2 VI ĐIỀU KHIỂN TI TIVAF4123G TRÊN KIT TIVA LAUNCHPAD	21
3.2 MẠCH CẦU H	22
3.3 MẠCH SCHMITT TRIGGER.....	23
3.4 MÔ HÌNH ĐỘNG CƠ	26
CHƯƠNG 4: XÂY DỰNG GIẢI THUẬT	28
4.1 BỘ ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU ĐỆ QUY	28

4.1.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC	28
4.1.2 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ	30
4.2 KHỐI MÔ HÌNH CHUẨN	33
4.2.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC	33
4.2.2 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ	36
4.3 BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN	39
4.3.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC	39
4.3.2 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ	41
CHƯƠNG 5: NHỮNG CẢI TIẾN TRONG GIẢI THUẬT	44
5.1 CẢI TIẾN BỘ ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU ĐỆ QUY	44
5.1.1 HỆ SỐ QUÊN THAY ĐỔI THEO THÔNG SỐ CỦA HỆ THỐNG	44
5.1.2 GIẢI THUẬT ƯỚC LƯỢNG CÓ TÍNH ĐẾN ĐỘ TRỄ CỦA ĐỘNG CƠ	45
5.1.3 GIẢI THUẬT NHÂN CHUỖI MA TRẬN VỚI SỐ PHÉP TOÁN TỐI ƯU	48
5.1.4 GIẢI THUẬT THAY ĐỔI MA TRẬN KHỞI ĐẦU $P(0)$	49
5.1.5 GIẢI THUẬT THAY ĐỔI MA TRẬN THÔNG SỐ KHỞI ĐẦU $\theta(0)$	50
5.2 CẢI TIẾN KHỐI MÔ HÌNH CHUẨN	50
5.2.1 GIẢI THUẬT CHỐNG TRÀN DO LỖI CHIA CHO 0 TRONG KHỐI MÔ HÌNH CHUẨN	50
5.3 TỐI ƯU HÓA BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN	51
5.3.1 THÊM MỘT KHẤU TÍCH PHẦN VÀO KHỐI R	51
5.3.2 VÀI PHÉP THỬ VỚI NHIỀU DẠNG MÔ HÌNH CHUẨN KHÁC NHAU	56
5.4 CẢI TIẾN KIẾN TRÚC LẬP TRÌNH	59
5.4.1 SỬ DỤNG NGẮT TRONG TRUYỀN VÀ NHẬN USART	59
5.4.2 SỬ DỤNG PHƯƠNG PHÁP ĐO VẬN TỐC TỐI ƯU	59
5.4.3 SỬ DỤNG BỘ ENCODER ĐẾM BẰNG PHẦN CỨNG x4 SỐ XUNG	60
5.4.4 SỬ DỤNG HIỆU QUẢ NHỮNG LỆNH TIỀN XỬ LÝ KHI LẬP TRÌNH	60
CHƯƠNG 6: CÁC KẾT QUẢ THỰC NGHIỆM	61
6.1 ĐIỀU KHIỂN VẬN TỐC ĐỘNG CƠ	63
6.1.1 ĐÁNH GIÁ CHẤT LƯỢNG ĐÁP ỨNG VÀ TÍNH THÍCH NGHI CỦA CÁC BỘ ĐIỀU KHIỂN TRÊN CÙNG MỘT MÔ HÌNH	63
6.1.2 ĐÁNH GIÁ CHẤT LƯỢNG CỦA TÍN HIỆU ĐIỀU KHIỂN	69
6.2 ĐIỀU KHIỂN VỊ TRÍ ĐỘNG CƠ	72
6.2.1 ĐÁNH GIÁ CHẤT LƯỢNG CỦA NHỮNG TỐI ƯU HÓA TRONG GIẢI THUẬT STR	72
CHƯƠNG 7: ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN	74
7.1 VỀ PHẦN GIẢI THUẬT ƯỚC LƯỢNG	74
7.2 VỀ PHẦN BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN	75

7.3 VỀ KHỐI MÔ HÌNH CHUẨN	75
7.4 VỀ PHẦN CỨNG HOẶC CODE LẬP TRÌNH	75
7.5 KẾ HOẠCH PHÁT TRIỂN	75
PHỤ LỤC TÀI LIỆU THAM KHẢO	76

DANH SÁCH BẢNG BIỂU, HÌNH VẼ, CÁC TỪ VIẾT TẮT

Hình 1: Các robot, hệ thống công nghiệp có thể thay thế những bộ phận hỏng hóc bằng những loại động cơ khác nhau mà vẫn đảm bảo độ ổn định của sản phẩm.....	2
Hình 2: Các thang máy và robot chuyên chở có thể hoạt động mượt và trơn tru hơn bất chấp sự thay đổi của tải.....	3
Hình 3: Lĩnh vực điều khiển chính xác áp dụng giải thuật thích nghi sẽ mang lại lợi ích lớn cho phát triển công nghiệp tại Việt Nam.....	3
Hình 4: Ứng dụng trong điều khiển chính xác nhiệt độ lò nhiệt y tế.....	4
Hình 5: So sánh giữa bộ điều khiển PID và STR.....	6
Hình 6: Mô hình thông số vật lý của động cơ DC.....	8
Hình 7: Bộ điều khiển theo mô hình chuẩn.....	12
Hình 8: Phương pháp bình phương tối thiểu đệ quy.....	14
Hình 9: Bộ điều khiển theo mô hình mẫu.....	16
Hình 10: Sơ đồ khối phần cứng của hệ thống điều khiển động cơ.....	18
Hình 11: Giới thiệu kit vi điều khiển STM32F4 Discovery.....	20
Hình 12: Giới thiệu kit vi điều khiển TivaF4.....	21
Hình 13 Mạch cầu H MC33883.....	22
Hình 14: Tín hiệu sau khi đi qua mạch Schmitt Trigger.....	23
Hình 15: Sơ đồ mạch Schmitt Trigger.....	24
Hình 16: Schematic mạch Schmitt Trigger.....	24
Hình 17: Mạch Schmitt Trigger.....	25
Hình 18: Mô hình động cơ 1 Encoder 1000 xung/vòng, điện áp 24VDC.....	26
Hình 19: Mô hình động cơ 2 Encoder 200 xung/vòng, điện áp 12VDC.....	27
Hình 20: Sơ đồ khối quá trình lập trình nhúng trong vi điều khiển.....	28
Hình 21: Bộ ước lượng bình phương tối thiểu đệ.....	28
Hình 22: Khối mô hình chuẩn.....	33
Hình 23: Tạo mô hình mẫu điều khiển vận tốc với Matlab.....	35
Hình 24: Tạo mô hình mẫu điều khiển vị trí với Matlab.....	38
Hình 25: Bộ điều khiển theo mô hình chuẩn.....	39
Hình 26: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 0.2 thời gian lấy mẫu.....	57
Hình 27: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 0.2 thời gian lấy mẫu khi có tải thay đổi.....	57
Hình 28: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 2 thời gian lấy mẫu.....	58
Hình 29: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 2 thời gian lấy mẫu khi có tải thay đổi.....	58
Hình 30: Phương pháp đo vận tốc bằng Input Capture.....	59
Hình 31: Phương pháp đo vị trí bằng Quadrature Encoder Interface.....	60
Hình 32: Đáp ứng của bộ PID với hệ số chưa tốt khi chưa có sự thay đổi của tải (toàn cục).....	63
Hình 33: Đáp ứng của bộ PID với hệ số chưa tốt khi chưa có sự thay đổi của tải (phóng to).....	63
Hình 34: Đáp ứng của bộ PID với hệ số chưa tốt với tải thay đổi.....	64
Hình 35: Tín hiệu điều khiển của bộ PID với hệ số chưa tốt với tải thay đổi.....	64
Hình 36: Đáp ứng của bộ PID chọn đúng hệ số khi chưa có sự thay đổi của tải (toàn cục).....	65
Hình 37: Đáp ứng của bộ PID chọn đúng hệ số khi chưa có sự thay đổi của tải (phóng to).....	65

Hình 38: Đáp ứng của bộ PID chọn đúng hệ số với tải thay đổi.....	66
Hình 39: Tín hiệu điều khiển của bộ PID chọn đúng hệ số với tải thay đổi	66
Hình 40: Đáp ứng của bộ STR khi chưa có sự thay đổi của tải	67
Hình 41: Đáp ứng của bộ STR khi chưa có sự thay đổi của tải (phóng to).....	67
Hình 42: Đáp ứng của bộ STR với tải thay đổi	68
Hình 43: Tín hiệu điều khiển của bộ STR với tải thay đổi.....	68
Hình 44: Tín hiệu điều khiển của bộ PID với hệ số chưa tốt (phóng to).....	69
Hình 45: Tín hiệu điều khiển của bộ PID với hệ số chưa tốt (toàn cục)	69
Hình 46: Tín hiệu điều khiển của bộ PID chọn đúng hệ số (phóng to).....	70
Hình 47: Tín hiệu điều khiển của bộ PID chọn đúng hệ số (toàn cục).....	70
Hình 48: Tín hiệu điều khiển của bộ STR (phóng to)	71
Hình 49: Tín hiệu điều khiển của bộ STR (toàn cục).....	71
Hình 50: Điều khiển vị trí động cơ với giải thuật STR cơ bản, xảy ra hiện tượng rung lắc động cơ khi đạt xác lập.....	72
Hình 51: Tín hiệu điều khiển vị trí động cơ của bộ PID bị bão hòa và dao động mạnh	73
Hình 52: Điều khiển vị trí động cơ với giải thuật STR cải tiến, hệ số trễ bằng 1, loại bỏ hoàn toàn hiện tượng rung lắc	73
Hình 53: Tín hiệu điều khiển vị trí động cơ của bộ STR	74
Bảng 1: Giải thích những thông số vật lý động cơ	8
Bảng 2: Giải thích những thông số trong hàm truyền liên tục và rời rạc động cơ	10
Bảng 3: Giải thích các thông số bộ ước lượng điều khiển vận tốc	29
Bảng 4: Giải thích các thông số bộ ước lượng điều khiển vận tốc	32
Bảng 5: So sánh hai giải thuật nhân chuỗi ma trận	49
Bảng 6: Xác định các thông số PID bằng tay	62
Bảng 7: Xác định hệ số PID bằng Ziegler–Nichols	62

CHƯƠNG 1: GIỚI THIỆU

1.1 TỔNG QUAN VỀ BỘ ĐIỀU KHIỂN TỰ CHỈNH

Trong những năm trở lại đây, nền công nghiệp sản xuất phát triển vô cùng mạnh mẽ, các hệ thống bằng tay thông thường dần dần được thay thế hết bằng các hệ thống tự động tiên tiến. Điều này thúc đẩy lĩnh vực nghiên cứu về kỹ thuật điều khiển robot tự động phát triển mạnh một cách vượt bậc cả về lý thuyết lẫn ứng dụng. Nói đến robot, ta có thể thấy trên thế giới đã xuất hiện các loại robot đạt được tốc độ cực kỳ cao và vô cùng chính xác như các máy DEK gắn các linh kiện điện tử lên PCB, máy CNC dùng để đục khắc, tạo hình với quy mô lớn... Còn trong quân sự, các loại robot di động giám sát, rà phá bom mìn được sử dụng để tránh các tổn thất về con người được sử dụng. Robot hầu hết được sử dụng thay thế con người trong các môi trường có rủi ro cao, mang tính nguy hiểm như môi trường lòng đất, lò phản ứng hạt nhân, thao tác ngoài không gian hoặc dưới đáy biển. Lĩnh vực nghiên cứu robot là một lĩnh vực rộng lớn bao gồm nhiều phạm vi như vật lý học, điều khiển học, xử lý hình ảnh, cơ khí chính xác, kỹ thuật điện tử, kỹ thuật máy tính.

Nhắc đến robot, chúng ta không thể không nhắc đến những giải thuật điều khiển động cơ DC, điển hình là bộ điều khiển PID. Bộ điều khiển PID trong các thiết kế bộ điều tốc xuất hiện từ những năm 1890. Các bộ điều khiển PID sau đó được phát triển trong hệ thống lái tàu thủy tự động. Một trong những ví dụ sớm nhất của bộ điều khiển kiểu PID được phát triển bởi Elmer Sperry năm 1911, trong khi tác phẩm phân tích lý thuyết đầu tiên về bộ điều khiển PID được kỹ sư người Mỹ gốc Nga Nicolas Minorsky xuất bản, vào năm Minorsky 1922. [Theo Wikipedia]

Một trong những đề tài hấp dẫn của lý thuyết điều khiển học hiện đại là nghiên cứu và tối ưu giải thuật thích nghi, để sao cho chỉ với một giải thuật duy nhất có thể điều khiển mọi động cơ và hệ thống khác nhau. Tuy không có tài liệu cụ thể bộ điều khiển thích nghi đầu tiên được sử dụng vào năm nào nhưng theo một số thông tin trên internet thì vấn đề này thu hút nhiều sự chú ý của khoa học vào khoảng những năm 60, 70 của thế kỉ trước.

Quyển sách “K. J. Astrom and B. Wittenmark, Adaptive Control, Addison-Wesley, 1989, 2d ed. 1994” là một trong những tài liệu nổi tiếng nhất thế giới về lĩnh vực điều khiển thích nghi.

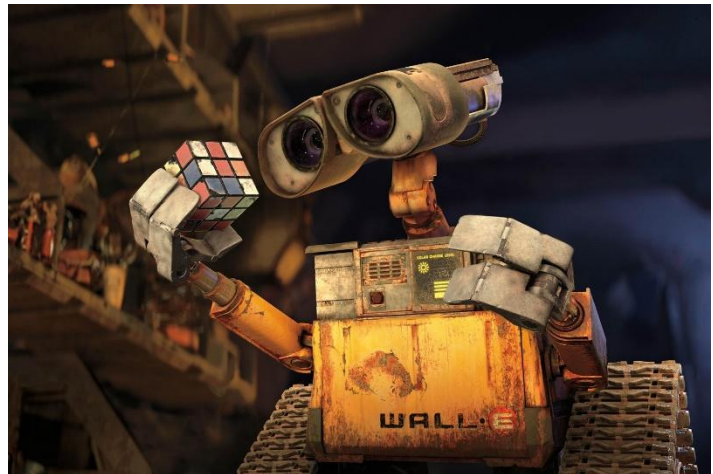
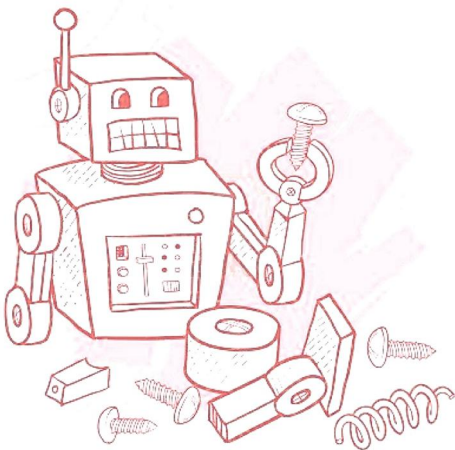
1.2 TIỀM NĂNG, Ý NGHĨA KHOA HỌC VÀ ỨNG DỤNG

Động cơ DC được sử dụng rộng rãi trong công nghiệp vì tính chính xác, tin cậy, ổn định, giá thành hợp lý. Giải thuật điều khiển PID được sử dụng điều khiển động cơ DC vì tính đơn giản nhưng vẫn đạt được hiệu quả tương đối trong những trường hợp cố định.

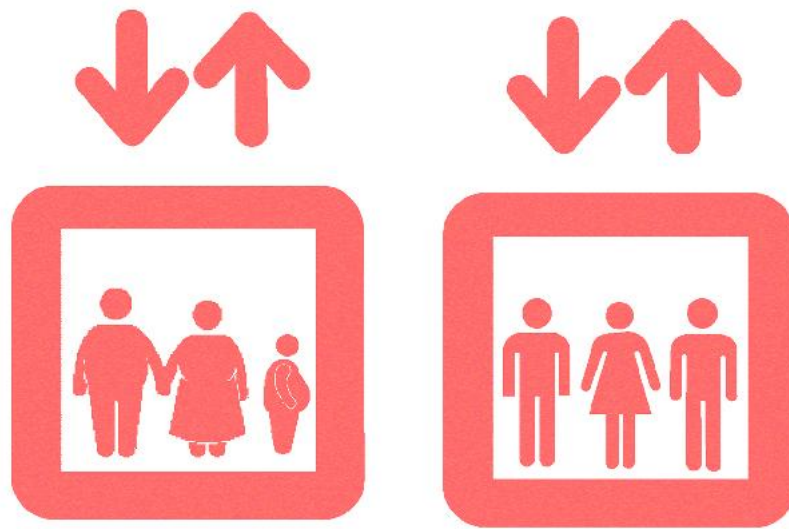
Tuy nhiên, để điều khiển hiệu quả bộ điều khiển PID, chúng ta cần xác định đúng hệ số K_p , K_i , K_d , vì thế trong những hệ thống thay đổi liên tục, việc sử dụng bộ điều khiển PID thể hiện nhiều nhược điểm.

Trong những năm gần đây, bộ điều khiển tự chỉnh tỏ ra rất hứa hẹn trong những hệ thống mà thông số chưa biết hoặc luôn thay đổi bởi khả năng tự thích nghi, cho đáp ứng nhanh và chính xác song song với sự phát triển của những bộ vi điều khiển hiệu năng cao.

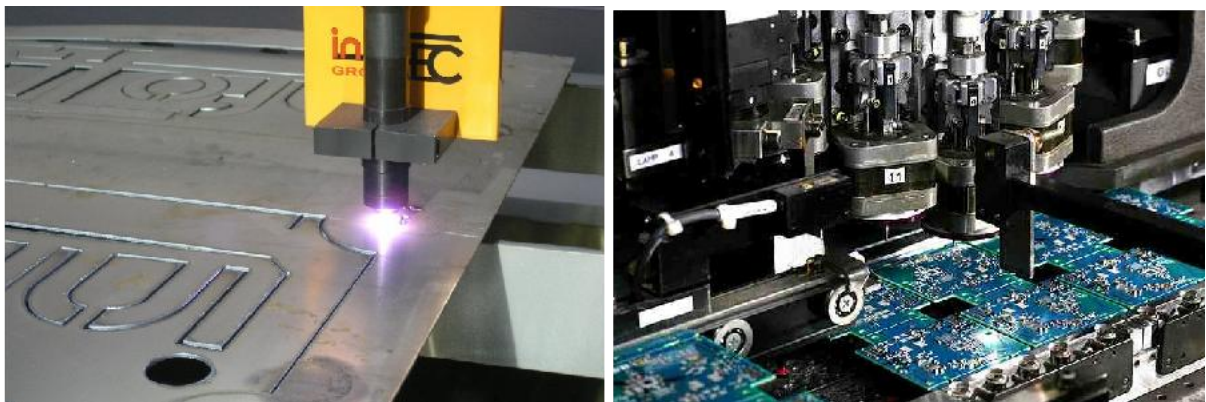
Ví dụ cụ thể:



Hình 1: Các robot, hệ thống công nghiệp có thể thay thế những bộ phận hỏng hóc bằng những loại động cơ khác nhau mà vẫn đảm bảo độ ổn định của sản phẩm.



Hình 2: Các thang máy và robot chuyên chở có thể hoạt động mượt và trơn tru hơn bất chấp sự thay đổi của tải.



Hình 3: Lĩnh vực điều khiển chính xác áp dụng giải thuật thích nghi sẽ mang lại lợi ích lớn cho phát triển công nghiệp tại Việt Nam



Hình 4: Ứng dụng trong điều khiển chính xác nhiệt độ lò nhiệt y tế

Trước khi luận văn này được công bố, đã có vài nghiên cứu về bộ điều khiển tự chỉnh như trong tài liệu [6], [7]. Tuy nhiên, giải thuật này được nhúng trong những kit vi xử lý DSP đắt tiền, khiến cho bộ điều khiển thú vị này không thể đến với những sinh viên có đam mê nghiên cứu trong lĩnh vực điều khiển thông minh. Vì thế, luận văn này được biên soạn nhằm mang giải thuật điều khiển thích nghi đến gần hơn với nền khoa học kỹ thuật còn non trẻ nhưng rất nhiệt huyết và tiềm năng tại Việt Nam.

1.3 MỤC TIÊU LUẬN VĂN VÀ PHƯƠNG PHÁP NGHIÊN CỨU

Mục tiêu của luận văn là tạo ra một giải thuật có hiệu năng và độ ổn định hơn bộ điều khiển PID truyền thống. Đồng thời giải thích cách thực hiện cũng như ý nghĩa của các công thức trong thuật toán một cách đơn giản nhất, để cho tất cả những sinh viên có thể áp dụng bộ điều khiển này trong sản phẩm của mình. Các bạn có thể tập trung phát triển những ứng dụng khác mà không cần phải lo lắng về vấn đề điều khiển động cơ nữa, nhờ đó mà ngành khoa học kỹ thuật Việt Nam có thể mở rộng ra nhiều lĩnh vực, mang tính chuyên môn hóa cao, không phải loay hoay mãi trong những vấn đề cũ nữa.

Phương pháp nghiên cứu của tác giả là làm với mô hình thực tế, đúc kết từ thực nghiệm. Do mô hình điều khiển động cơ đơn giản và khó có sai sót, các bạn cải thiện thuật toán từ thực tế sẽ hiểu rõ các vấn đề khi thực nghiệm hơn là làm với các phần mềm giả lập. Đồng thời, các bạn hãy chia sẻ giải thuật này cho những người bạn cùng làm đồ án, luận văn để họ sử dụng trong những sản phẩm của họ, nhờ đó chúng ta có một luồng ý kiến và góp ý từ những người dùng thực tế.

1.4 SƠ LƯỢC NỘI DUNG LUẬN VĂN

Luận văn gồm 7 chương và phụ lục những tài liệu tham khảo. Cụ thể như sau:

CHƯƠNG 1:

Giới thiệu chung, cung cấp cho người đọc một cái nhìn khái quát về các phương pháp điều khiển động cơ cũng như hiểu được vai trò to lớn của bộ điều khiển thích nghi trong ngành robot và điều khiển tự động.

CHƯƠNG 2:

Giải thích và chứng minh bằng toán học những cơ sở lý thuyết về mô hình rời rạc của động cơ DC, về giải thuật ước lượng bình phương tối thiểu đệ quy, để dễ dàng hiểu được giải thuật thích nghi trong chương 4.

CHƯƠNG 3:

Trình bày quá trình thiết kế phần cứng trong luận văn của tác giả, một số phương pháp làm ổn định phần cứng để tập trung hoàn thiện phần mềm.

CHƯƠNG 4:

Trình bày toàn bộ giải thuật cơ bản của các khối trong sơ đồ khối bộ điều khiển tự chỉnh. Bao gồm khối ước lượng thông số, khối mô hình mẫu, khối điều khiển thích nghi theo mô hình mẫu.

CHƯƠNG 5:

Trình bày những phiên bản cải tiến và tối ưu hóa giúp giải thuật ở chương 4 hoàn hảo và ổn định hơn của những nhà khoa học từng nghiên cứu về vấn đề này trên thế giới.

CHƯƠNG 6:

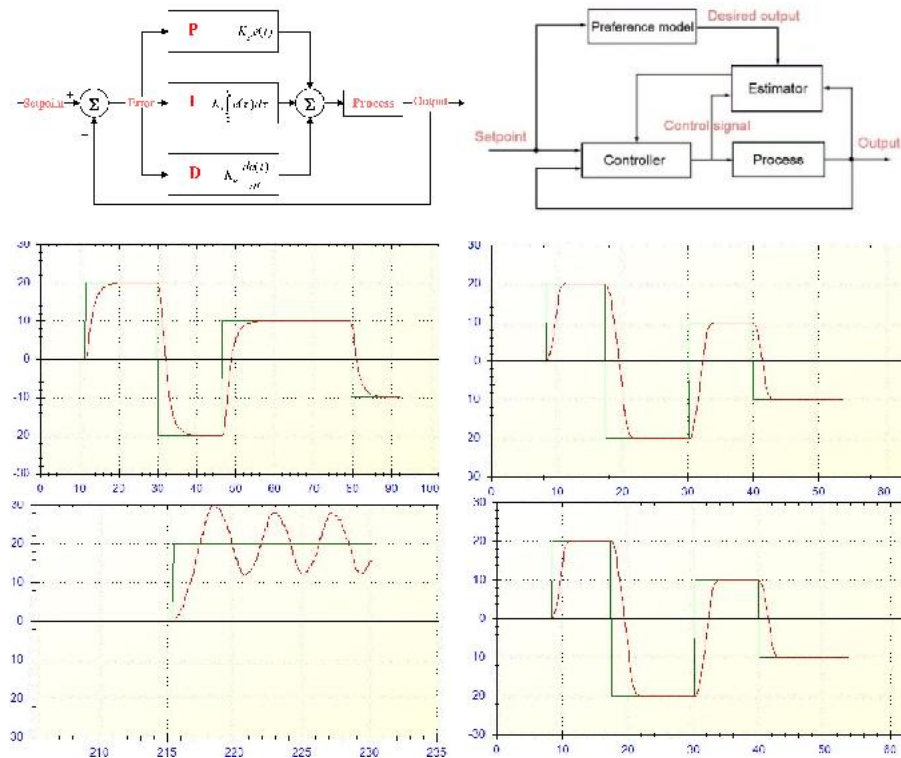
Kết quả thực nghiệm nhằm nhấn mạnh hiệu năng và tính thích nghi của bộ điều khiển tự chỉnh so với bộ PID truyền thống.

CHƯƠNG 7:

Trình bày những vấn đề chưa hoàn thiện và hướng giải quyết. Giúp những nghiên cứu sinh có hướng nghiên cứu và phát triển rõ ràng để hoàn thiện hóa đề tài này

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 GIỚI THIỆU CHUNG VỀ HAI BỘ ĐIỀU KHIỂN PID VÀ STR



Hình 5: So sánh giữa bộ điều khiển PID và STR

BỘ ĐIỀU KHIỂN PID

- + Đơn giản và hiệu quả trong những trường hợp cố định.
- Cần những thông số chính xác của mô hình hệ thống.
- Hiệu năng sẽ giảm nếu mô hình hệ thống thay đổi.

BỘ ĐIỀU KHIỂN TỰ CHỈNH:

- + Sử dụng đặc biệt hiệu quả trong trường hợp thông số hệ thống chưa biết hoặc thay đổi liên tục.
- + Cho ra đáp ứng động cơ giống nhau trong những trường hợp khác nhau.
- + Có khả năng học và thích nghi.
- Cần những vi điều khiển tương đối mạnh như Tiva™ C Series hay STM32F4 ARM Cortex-M4F.

2.2 GIẢI THÍCH CHỨC NĂNG CÁC KHỐI TRONG GIẢI THUẬT STR

BỘ ƯỚC LƯỢNG (ESTIMATOR):

Tín hiệu vào: Tín hiệu điều khiển điện áp động cơ, tín hiệu ra thực tế, thông số động cơ từ mô hình chuẩn.

Tín hiệu ra: Thông số ước lượng của hệ thống.

Chức năng:

1. Ước lượng trạng thái của động cơ thông qua những thông số đọc về từ encoder và tín hiệu tham khảo từ mô hình chuẩn.
2. Tính toán giá trị sai số ước lượng trong lần điều khiển tiếp theo.

BỘ ĐIỀU KHIỂN (CONTROLLER):

Tín hiệu vào: Thông số ước lượng của hệ thống, tín hiệu đặt, tín hiệu ra thực tế.

Tín hiệu ra: Tín hiệu điều khiển điện áp động cơ.

Chức năng:

Tính toán tín hiệu điều khiển dựa trên kết quả của bộ ước lượng sao cho động cơ có đáp ứng hoàn hảo giống mô hình mẫu.

KHỐI MÔ HÌNH CHUẨN (PREFERENCE MODEL):

Tín hiệu vào: Tín hiệu đặt.

Tín hiệu ra: Thông số động cơ từ mô hình chuẩn.

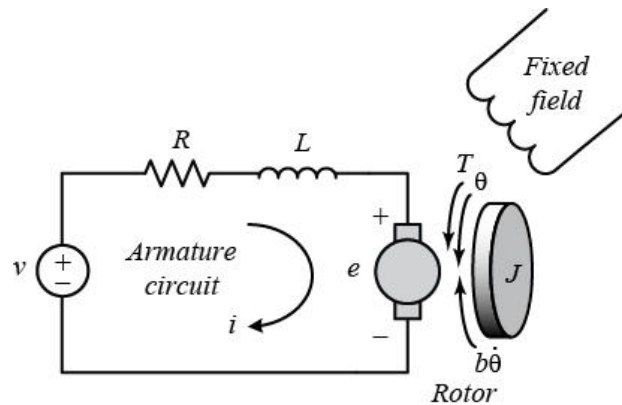
Chức năng:

1. Tính toán dựa trên những hàm của Matlab
2. Có đáp ứng hoàn hảo với vọt lố bằng không và thời gian xác lập nhanh nhất.

2.3 MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VỊ TRÍ VÀ VẬN TỐC ĐỘNG CƠ DC

2.3.1 MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VẬN TỐC ĐỘNG CƠ DC

Diễn giải toán học quá trình xác định hàm truyền vận tốc liên tục của động cơ DC:



Hình 6: Mô hình thông số vật lý của động cơ DC

Trong diễn giải này, chúng ta giả sử đầu vào của hệ thống là nguồn điện áp (Volt) áp dụng cho phần ứng của động cơ, trong khi đầu ra là tốc độ quay của trục phản cảm động cơ $d(\theta)/dt$. Các cánh quạt và trục được giả định là không bị trượt. Chúng tôi tiếp tục giả định rằng mô hình ma sát nhớt, tức là, mô-men xoắn ma sát tỉ lệ với trục vận tốc góc. Các thông số vật lý của động cơ được thể hiện dưới đây:

Bảng 1: Giải thích những thông số vật lý động cơ

Kí hiệu	Ý nghĩa	Thứ nguyên
J	Moment quán tính của rotor	$0,01 \text{ kg.m}^2$
b	Động cơ nhớt ma sát liên tục	$0,1 \text{ NMS}$
K_e	Điện động lực không đổi	$0,01 \text{ V / rad / giây}$

Kt	Động cơ mô-men xoắn không đổi	0,01 Nm / Amp
(R)	Điện kháng	1 Ohm
(L)	Điện cảm	0,5 H

Nhìn chung, mô-men xoắn được tạo ra bởi một động cơ DC tỷ lệ thuận với phản ứng hiện tại và sức mạnh của từ trường. Trong ví dụ này, chúng ta sẽ giả định rằng từ trường là không đổi và, do đó, mô-men xoắn động cơ chỉ có tỷ lệ thuận với điện áp phản ứng bởi HẰNG SỐ Kt không đổi như trong phương trình dưới đây. Điều khiển động cơ DC chính là điều khiển phản ứng của động cơ ấy:

$$T = K_t i$$

E tỷ lệ thuận với đạo hàm vận tốc góc của trục động cơ với hằng số Ke không đổi:

$$e = K_e \dot{\theta}$$

Trong hệ SI, mô-men xoắn động cơ và thông số emf bằng nhau, có nghĩa là, $K_t = K_e$, do đó, chúng tôi sẽ sử dụng K đại diện cho cả hai mô-men xoắn động cơ không đổi và thông số emf là hằng số.

Từ hình trên, chúng ta có thể lấy được các phương trình sau đây dựa trên định luật 2 Newton và định luật áp Kirchhoff:

$$J\ddot{\theta} + b\dot{\theta} = K i$$

$$L \frac{di}{dt} + Ri = V - K \dot{\theta}$$

Tính toán hàm truyền liên tục vận tốc động cơ

Áp dụng biến đổi Laplace, các phương trình mô hình trên có thể được viết lại theo biến s .

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V(s) - Ks \Theta(s)$$

Tìm hàm truyền vòng hở của động cơ bằng vài phép biến đổi đơn giản, ta được hàm truyền sau đây, trong đó tốc độ quay được coi là đầu ra và điện áp phần ứng được coi là đầu vào.

$$P(s) = \frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \left[\frac{rad/sec}{V} \right]$$

Từ hàm truyền liên tục vận tốc động cơ như trên, dựa vào lý thuyết rời rạc hóa, chúng ta thu được mô hình rời rạc vận tốc động cơ như sau:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} u(z)$$
$$y_m(z) = \frac{Bm}{Am} u_c(z) = \frac{b_{1m} z + b_{2m}}{z^2 + a_{1m} z + a_{2m}} u_c(z)$$

Bảng 2: Giải thích những thông số trong hàm truyền liên tục và rời rạc động cơ

Thông số	Giải thích
$u_c(t), u_c(k)$	Tín hiệu vận tốc hoặc vị trí mong muốn theo dạng liên tục và dạng rời rạc
$u(t), u(k)$	Tín hiệu điện áp ra từ bộ điều khiển theo dạng liên tục và dạng rời rạc
$y(t), y(k)$	Tín hiệu vận tốc hoặc vị trí thực tế theo dạng liên tục và dạng rời rạc
$y_m(t), y_m(k)$	Tín hiệu vận tốc hoặc vị trí từ mô hình mẫu theo dạng liên tục và dạng rời rạc

a1, a2, a3, b1, b2, b3	Thông số thực tế của mô hình toán học động cơ
a1 _m , a2 _m , a3 _m , b1 _m , b2 _m , b3 _m	Thông số mô hình mẫu của mô hình toán học động cơ

Tiếp theo phần này, Chương 4.2 sẽ đưa ra phương pháp để rời rạc hóa, xác định các thông số của mô hình mẫu nhờ vào hàm truyền vận tốc và vị trí đã được xác định như trên.

2.3.2 MÔ HÌNH TOÁN HỌC ĐIỀU KHIỂN VỊ TRÍ ĐỘNG CƠ DC

Diễn giải toán học quá trình xác định hàm truyền vị trí liên tục của động cơ DC:

Để tìm được mô hình điều khiển vị trí, ta thêm vào đó một khâu tích phân, tức là nhân s vào mẫu số do vận tốc là tích phân của vị trí.

$$\frac{\Theta(s)}{V(s)} = \frac{K}{s[(Js + b)(Ls + R) + K^2]} \left[\frac{rad}{V} \right]$$

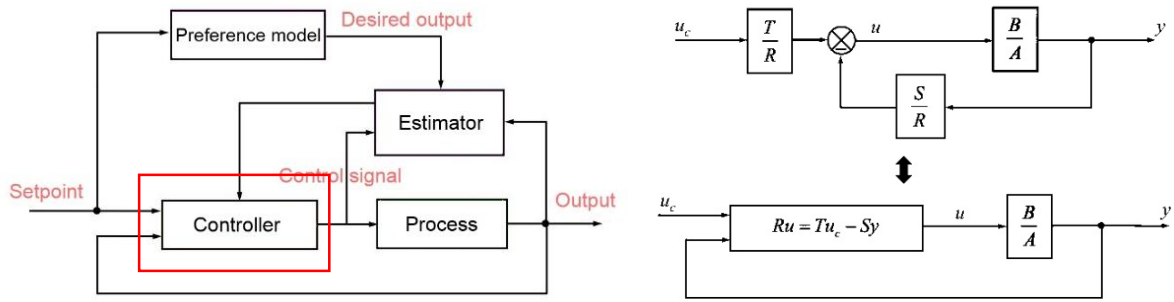
Từ hàm truyền liên tục vị trí động cơ như trên, dựa vào lý thuyết rời rạc hóa, chúng ta thu được mô hình rời rạc vị trí động cơ như sau:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3} u(z)$$

$$y_m(z) = \frac{Bm}{Am} u_c(z) = \frac{b_1 m z^2 + b_2 m z + b_3 m}{z^3 + a_1 m z^2 + a_2 m z + a_3 m} u_c(z)$$

Tiếp theo phần này, Chương 4.2 sẽ đưa ra phương pháp để rời rạc hóa, xác định các thông số trong hàm truyền vận tốc và vị trí động cơ từ hàm truyền liên tục đã được xác định như trên.

2.4 LÝ THUYẾT THIẾT KẾ BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN



Hình 7: Bộ điều khiển theo mô hình chuẩn

Lý thuyết

1. Bài toán đặt ra là thiết kế R, T, S để đáp ứng của hệ kín $\frac{B}{A}$ bám hoàn hảo theo mô hình chuẩn $\frac{B_m}{A_m}$.
2. So sánh hàm truyền hệ kín với mô hình chuẩn:

$$y(t) = \frac{BT}{AR + BS} u_c(t)$$

$$y_m(t) = \frac{B_m}{A_m} u_c(t)$$

3. Để đạt được đáp ứng vòng kín mong muốn, cần có điều kiện:
 - Phương trình đặc trưng của hệ kín phải có các cực trùng với các cực của mô hình chuẩn tức là $AR + BS$ phải chia hết cho các cực của mô hình chuẩn, tức là $AR + BS$ phải chia hết cho A_m .
 - Các zero nằm bên trái mặt phẳng phức của B phải được triệt tiêu bởi các cực của hệ kín. Giả sử có thể phân tích $B = B^+ B^-$ (với B^+ gồm các zero bên trái mặt phẳng phức), chúng ta cần điều kiện $AR + BS$ chia hết cho B^+ .
4. Từ đó ta có phương trình Diophantine:

$$AR + BS = A_0 A_m B^+$$

5. Để có thể khử B^+ , R phải có dạng:

$$R = R_1 B^+$$

$$\Rightarrow AR_1 B^+ + BS = A_0 A_m B^+$$

$$\Rightarrow AR_1 + BS = A_0 A_m$$

6. Với các điều kiện trên hàm truyền hệ kín trở thành:

$$y(t) = \frac{B^+ B^- T}{A_0 A_m B^+} u_c(t)$$

7. Rút gọn và so sánh với mô hình mẫu, ta có:

$$y(t) = \frac{B^- T}{A_0 A_m} u_c(t)$$

$$y_m(t) = \frac{B_m}{A_m} u_c(t)$$

8. Điều kiện cần để hàm truyền hệ kín đúng bằng mô hình mẫu là:

$$B_m = B^- B'_m$$

$$T = A_0 B'_m$$

9. Điều kiện đảm bảo tồn tại lời giải bài toán điều khiển theo mô hình chuẩn là:

$$\text{Bậc } A_0 \geq 2 \text{ Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1$$

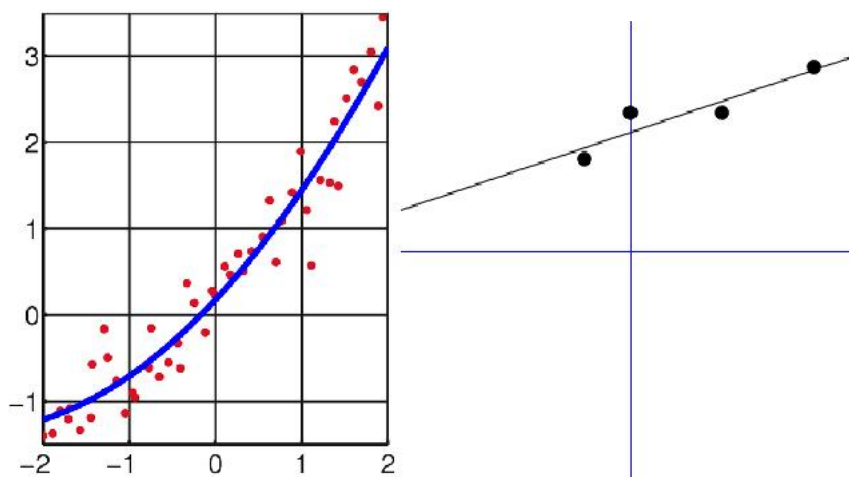
$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

10. Phương pháp đơn giản tìm nghiệm phương trình Diophantine:

- Chọn bậc của đa thức R và S phù hợp.
- Cân bằng các hệ số của phương trình Diophantine sẽ tìm được các hệ số của R và S.

2.5 CƠ SỞ TOÁN HỌC VÀ Ý NGHĨA THỰC TIỄN CỦA PHÉP ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU

Trong toán học, phương pháp bình phương tối thiểu, còn gọi là bình phương nhỏ nhất hay bình phương trung bình tối thiểu, là một phương pháp tối ưu hóa để lựa chọn một đường khớp nhất cho một dải dữ liệu ứng với cực trị của tổng các sai số (error) giữa đường khớp và dữ liệu. Phương pháp này giả định các sai số (error) của phép đo đạc dữ liệu phân phối ngẫu nhiên.



Hình 8: Phương pháp bình phương tối thiểu đệ quy

Bình phương tối thiểu tuyến tính là một kỹ thuật trong ngành tối ưu toán học để tìm một nghiệm gần đúng cho một hệ phương trình tuyến tính không có nghiệm chính xác. Điều này thường xảy ra khi số phương trình là (m) lớn hơn số biến (n).

Theo từ ngữ toán học, chúng ta muốn tìm nghiệm của "phương trình":

$$A\mathbf{x} \approx \mathbf{b}$$

với A là một ma trận cỡ m-nhân-n (với $m > n$) và x và b theo thứ tự đó là vectơ cột với n- và m-hàng. Một cách chính xác hơn, ta muốn làm tối thiểu chuẩn Euclidean bình phương của phần dư $A\mathbf{x} - \mathbf{b}$.

Trong trường hợp đó, nghiệm của hệ phương trình tuyến tính là duy nhất và được cho bởi:

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b}.$$

Ví dụ

Cho các điểm (0, 3), (2, 3), (4, 4), (-1, 2). Chúng ta tìm một lời giải có dạng $\alpha x + \beta = y$, nghĩa là:

$$\begin{pmatrix} x & 1 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = y$$

Chúng ta sau đó có thể lập ma trận A:

$$A = \begin{pmatrix} 0 & 1 \\ 2 & 1 \\ 4 & 1 \\ -1 & 1 \end{pmatrix}$$

$$A^T = \begin{pmatrix} 0 & 2 & 4 & -1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

$$A^T A = \begin{pmatrix} 21 & 5 \\ 5 & 4 \end{pmatrix}$$

và vector **b**:

$$\mathbf{b} = \begin{pmatrix} 3 \\ 3 \\ 4 \\ 2 \end{pmatrix}$$

và sau đó:

$$A^T \mathbf{b} = \begin{pmatrix} 20 \\ 12 \end{pmatrix}$$

Hình vẽ các điểm và lời giải.

$$A^T A \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = A^T \mathbf{b}$$

$$\begin{pmatrix} 21 & 5 \\ 5 & 4 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} 20 \\ 12 \end{pmatrix}$$

Sau đó,

$$(A^T A)^{-1} = \frac{1}{59} \begin{pmatrix} 4 & -5 \\ -5 & 21 \end{pmatrix}$$

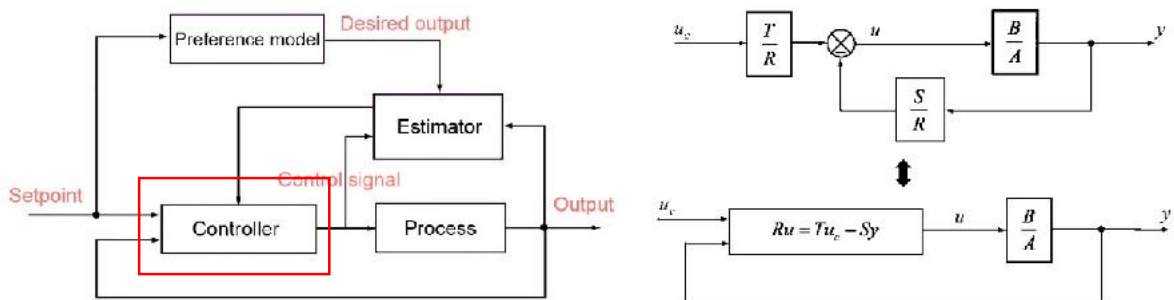
và

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \frac{1}{59} \begin{pmatrix} 4 & -5 \\ -5 & 21 \end{pmatrix} \begin{pmatrix} 20 \\ 12 \end{pmatrix} = \begin{pmatrix} 20/59 \\ 152/59 \end{pmatrix}$$

và đường thẳng tốt nhất là $(20/59)x + 152/59 = y$.

Bây giờ chúng ta đối chiếu với giải thuật bên dưới là đủ để hiểu được nguyên tắc hoạt động của giải thuật bình phương tối thiểu đệ quy trong ước lượng thông số động cơ. Giải thuật dựa vào một số lượng thay đổi đáp ứng đầu ra và đầu vào của những lần lấy mẫu trước đó để tìm ra các thông số $a_1, a_2, a_3, b_1, b_2, b_3$ thể hiện chính xác mô hình toán học động cơ ngay tại thời điểm tính toán. Từ đó mà bộ điều khiển theo mô hình chuẩn tính toán giá trị điện áp phù hợp để động cơ bám theo đáp ứng mong muốn.

2.6 PHƯƠNG PHÁP THIẾT KẾ LUẬT ĐIỀU KHIỂN THEO MÔ HÌNH MẪU



Hình 9: Bộ điều khiển theo mô hình mẫu

Trình tự giải

Đối tượng:

$$y(t) = \frac{B}{A}u(t)$$

Mô hình mẫu:

$$y_m(t) = \frac{B_m}{A_m}u_c(t)$$

Bước 1: Phân tích B dưới dạng $B = B^+ B^-$

Bước 2: Kiểm tra mô hình mẫu có thỏa mãn điều kiện cần để tồn tại lời giải hay không

$$B_m = B^- B'_m$$

$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

Bước 3: Chọn bậc của A_0 thỏa mãn điều kiện tồn tại lời giải

$$\text{Bậc } A_0 \geq 2 \text{ Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1$$

Bước 4: Chọn bậc của S và R_1

$$\text{Bậc } R_1 = \text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } A$$

$$\text{Bậc } S = \min\{[\text{Bậc } R_1 + \text{Bậc } B^+], [\text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } B^-]\}$$

Bước 5: Tính S và R_1 bằng cách giải phương trình:

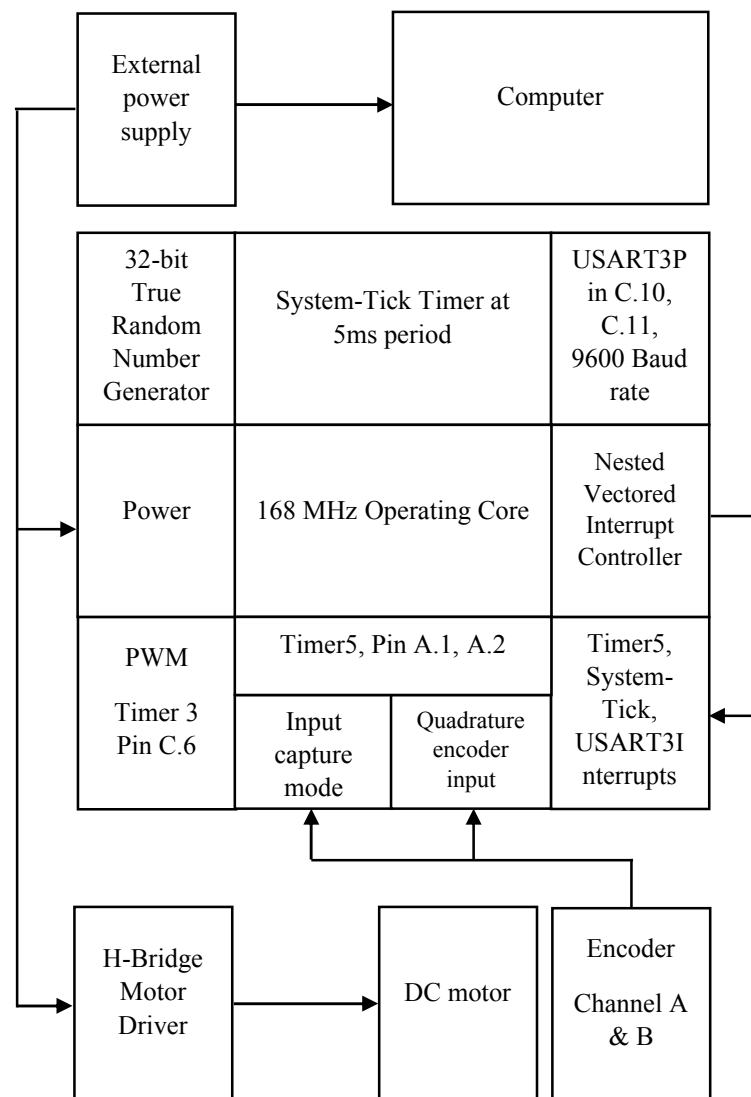
$$AR_1 + BS = A_0A_m$$

Bước 6: Tính R và T

$$R = R_1B^+$$

$$T = A_0B'_m$$

CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG PHẦN CỨNG



Hình 10: Sơ đồ khối phần cứng của hệ thống điều khiển động cơ

3.1 BỘ VI XỬ LÝ

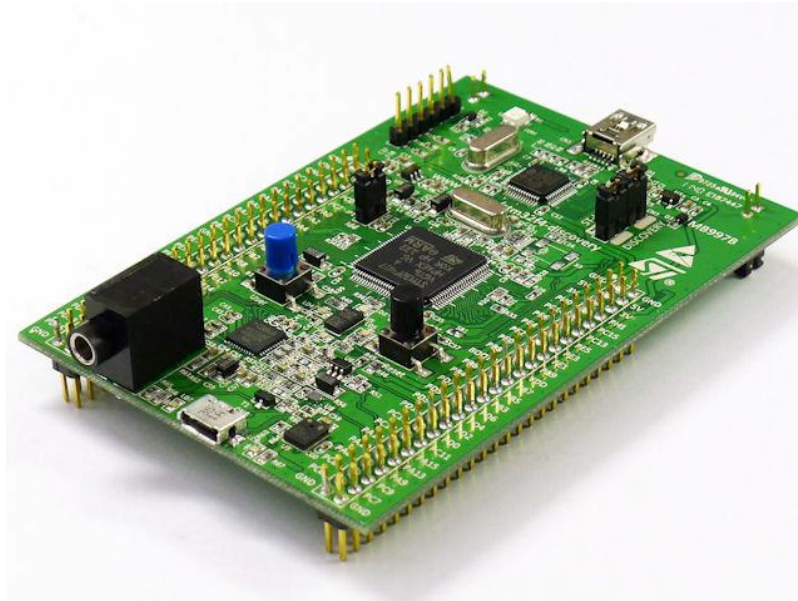
Chúng ta nên dùng những dòng chip ARM®Cortex™M vì những dòng chip này có xung nhịp tương đối từ 80MHz đến 500Mhz, cùng bộ tính toán số thực có sẵn trong chip giúp tính toán nhanh giải thuật ước lượng nhân chia nhiều ma trận số thực trong ít nhất là 5ms, chu kì trung bình để có thể điều khiển mượt động cơ DC mà những dòng chip bình thường như 8051, PIC16, Arduino không thể làm được điều đó. Mặt khác, chip ARM hỗ trợ thư viện rất mạnh mẽ với giá cả cạnh tranh khiến chúng ta đơn giản hóa phần nào quá trình thực hiện bộ điều khiển tự chỉnh.

1. Những module cần có của vi điều khiển để ứng dụng bộ điều khiển thích nghi gồm có:
Module System tick hoặc timer để tạo ra chính xác chu kì điều khiển động cơ là 5ms hoặc 10ms.
2. Ngoài ra cần phải có 1 bộ PWM để điều khiển điện áp động cơ.
3. Một bộ đọc Encoder Quadrature bằng phần cứng.
4. Một chân GPIO để điều khiển hướng động cơ.
5. Một module UART để truyền tín hiệu hai chiều với máy tính hoặc với module khác.

Vì vậy, hai vi điều khiển dưới đây được sử dụng có đầy đủ tính năng và giá thành hợp lý để thực hiện bộ điều khiển tự chỉnh cho động cơ DC:

1. Vi điều khiển STM32F4 (hình 8)
2. Vi điều khiển TIVAF4 (hình 9)

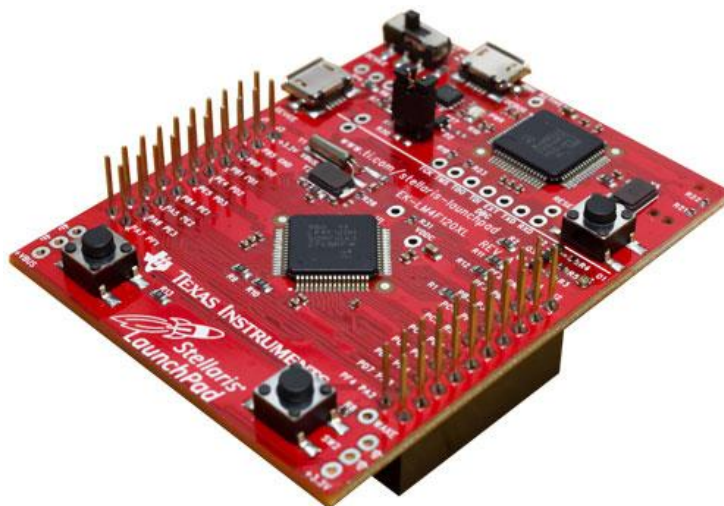
3.1.1 VI ĐIỀU KHIỂN STM32F407 TRÊN KIT STM32F4 DISCOVERY



1. Vi điều khiển STM32F407VGT6 với tính năng 32-bit ARM Cortex-M4F lõi, 1 MB Flash, 192 KB bộ nhớ RAM trong một gói LQFP100
2. On-board ST-LINK/V2 với chế độ lựa chọn chuyển sang sử dụng các bộ như một ST-LINK/V2 độc lập (với kết nối SWD cho lập trình và gỡ lỗi)
3. Điện nguồn cung cấp: 3 V và 5 V
4. LIS302DL, cảm biến chuyển động ST MEMS, 3 trục đầu ra kỹ thuật số gia tốc kế
5. MP45DT02, ST MEMS cảm biến âm thanh, microphone kỹ thuật số omni-directional
6. CS43L22, DAC âm thanh tích hợp trình điều khiển lớp D loa
7. LD1 (màu đỏ / xanh lá cây) để giao tiếp USB, LD2 (màu đỏ) cho 3,3 V điện trên
8. Bốn đèn LED sử dụng, LD3 (màu cam), LD4 (màu xanh), LD5 (màu đỏ) và LD6 (màu xanh)
9. 2 USB OTG đèn LED LD7 (màu xanh) và VBus LD8 (màu đỏ) quá dòng
10. Hai nút bấm (người sử dụng và reset)
11. USB OTG FS với cổng nối micro-AB
12. Header mở rộng cho tất cả các LQFP100 I / O để kết nối nhanh chóng và dễ dàng tạo mẫu bảng thăm dò
13. DSP ứng dụng xử lý tính toán số học

Hình 11: Giới thiệu kit vi điều khiển STM32F4 Discovery

3.1.2 VI ĐIỀU KHIỂN TI TIVAF4123G TRÊN KIT TIVA LAUNCHPAD



1. 32-bit ARM Cortex™® -M4 80 - MHz xử lý lõi với hệ thống Timer (SysTick) , tích hợp lồng nhau vector điều khiển ngắt (NVIC) , Wake-Up điều khiển ngắt (WIC), bộ nhớ Kiểm (MPU) , Floating- Point Unit (FPU)
2. On-chip bộ nhớ , có 256 KB, 32 KB SRAM ; ROM nội bộ nạp với TivaWare™ cho phần mềm C Series ; 2KB EEPROM
3. Tám UARTs, bốn Giao diện nối tiếp mô-đun (SSI) đồng bộ, hỗ trợ hoạt động cho Freescale SPI, Microwire , hoặc Texas công cụ giao diện nối tiếp đồng bộ , bốn liên Vi mạch (I2C) mô-đun, cung cấp tiêu chuẩn (100 Kbps) và Fast (400 Kbps)
4. Hỗ trợ tương tự, có tính năng: hai 12-bit Analog-to- kỹ thuật số chuyển đổi (ADC) với 12 kênh đầu vào tương tự và một tỷ lệ mẫu của một triệu mẫu / giây
5. Điều khiển chuyển động tiên tiến, có tính năng: tám Độ rộng xung điều chế (PWM) khối máy phát điện , mỗi một 16 -bit truy cập , hai bộ so sánh PWM, một máy phát tín hiệu PWM, một máy phát điện chết -band, và chọn ngắt / ADC- kích hoạt , hai PWM lỗi đầu vào để thúc đẩy tắt máy có độ trễ thấp , hai Quadrature (QEI) mô-đun, với vị trí tích hợp để theo dõi vị trí bộ mã hóa và chụp tốc độ sử dụng được xây dựng trong bộ đếm thời gian.
6. 12 16/32-bit và 12 32/64-bit Capture So sánh PWM.
7. Lên đến 43 GPIO (tùy thuộc vào cấu hình), điều khiển lập trình cho ngắt GPIO.
8. Nhiều nguồn đồng hồ cho đồng hồ hệ thống vi điều khiển : chính xác Oscillator (PIOSC) , chính Oscillator (MOSC) , 32,768 kHz và nội bộ 30 kHz Oscillator.

Hình 12: Giới thiệu kit vi điều khiển TivaF4

3.2 MẠCH CẦU H

Tính năng kỹ thuật board mạch công suất KHT-H_BRIDGE_DCMOTOR_MC33883

1. Phân công suất dùng 4 POWER MOFETs IRF3205 có thông số : $V_{DS} = 55V$, $R_{DS(on)} = 8.0m\Omega$, $I_D = 110A$
2. Điện áp cấp ngõ vào từ +12V đến +40VDC.
3. Sử dụng opto xung 6N137, và IC lái MC33883 nên tần số PWM có thể đáp ứng lên đến 100 KHz.
4. Ngõ vào opto cách ly PWM+, PWM-, DIR+, DIR-.
5. Sử dụng IC kích FET chuyên dụng MC33883 của Freescale, nên duty PWM đạt được 100%, ưu điểm hơn so với các IC kích FET khác.
6. Có Led báo nguồn, Led báo chiều động cơ và PWM.
7. Board có 2 nút nhấn DIR và PWM thích hợp cho việc test board bằng tay.
8. Board được thiết kế nhỏ gọn, có socket bắt trên nhôm rail.



Hình 13 Mạch cầu H MC33883

Cầu H điều khiển động cơ với thời gian delay ngắn hơn từ delay micro giây xuống còn nano giây, giúp cho giải thuật điều khiển gần hơn với lý thuyết mô phỏng, chính xác và hiệu quả hơn.

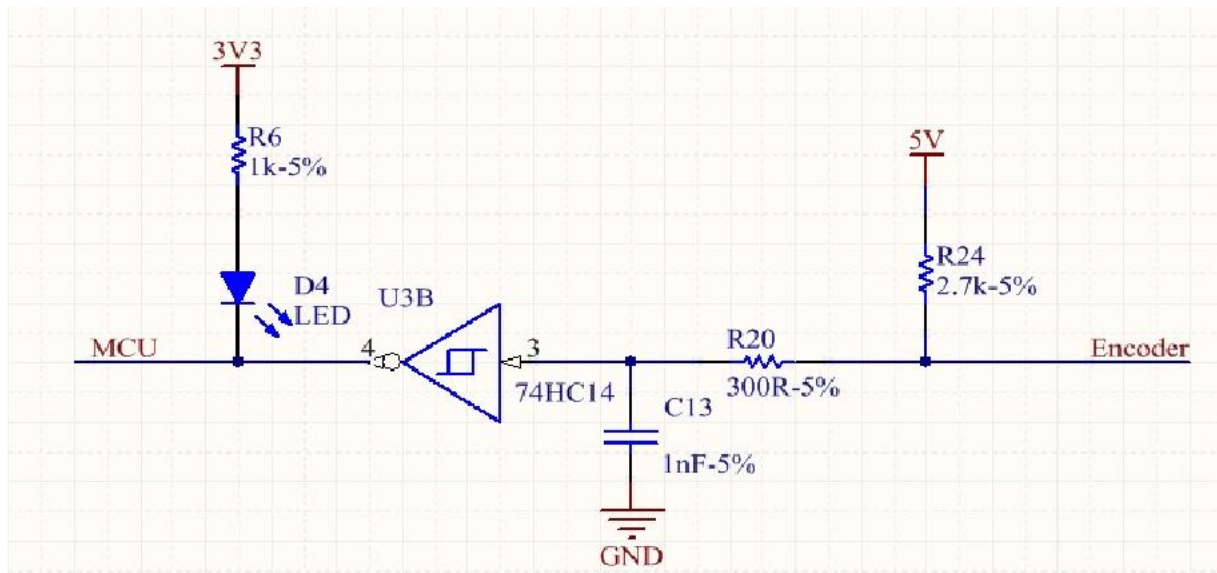
3.3 MẠCH SCHMITT TRIGGER

Schmitt Trigger là mạch so sánh: khi điện áp đầu vào lớn hơn ngưỡng trên thì ngõ ra là cao (nếu ST không đảo) hoặc thấp (nếu ST đảo). Ngõ ra sẽ giữ nguyên mức nếu điện áp nằm giữa ngưỡng trên và ngưỡng dưới. Khi điện áp vào thấp hơn ngưỡng dưới thì ngõ ra sẽ thay đổi.



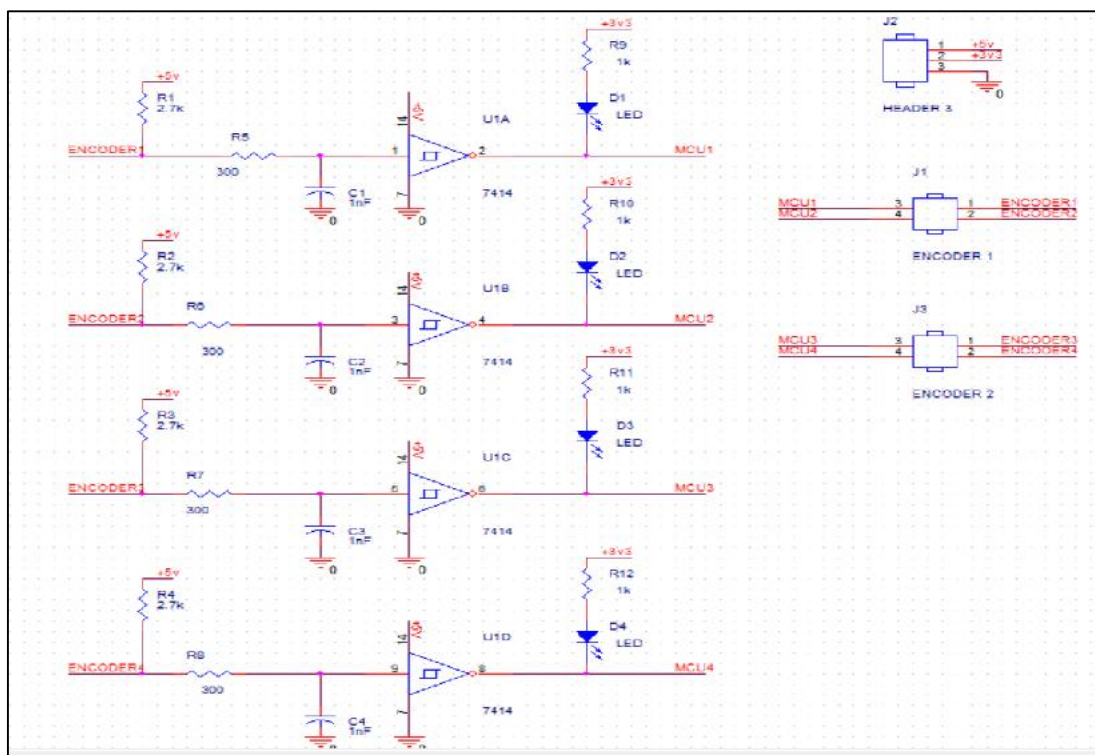
Hình 14: Tín hiệu sau khi đi qua mạch Schmitt Trigger

Mục đích của mạch này là dùng để lọc tín hiệu ngõ ra đọc từ encoder về MCU. Do quá trình xử lý tín hiệu, phần động lực điều khiển cầu H sẽ khiến cho tín hiệu encoder bị nhiễu dẫn tới việc đo đặc vị trí các thông số không được chính xác. Vì thế tôi xin lấy bản thiết kế từ nhóm bạn Đặng Duy Thành – MSSV 40902457 mạch Schmitt Trigger theo sơ đồ sau:

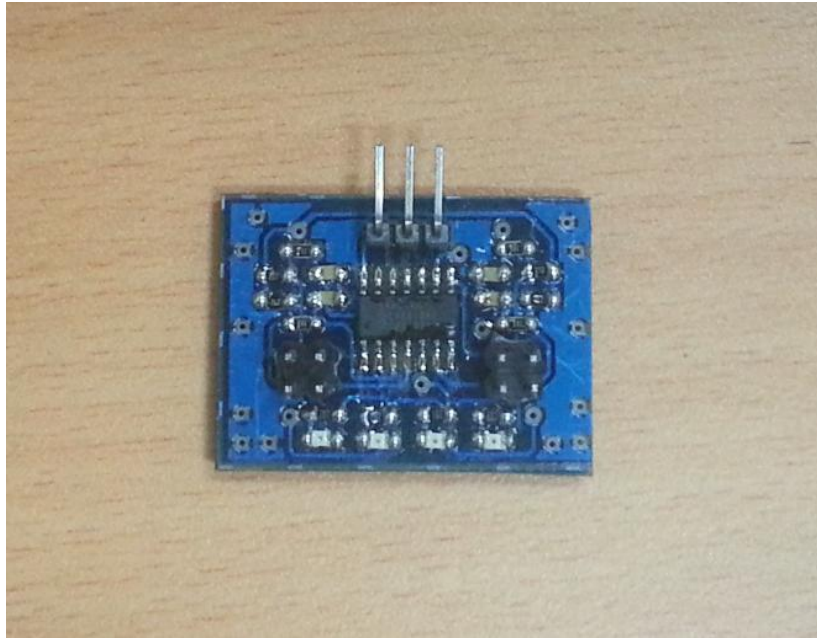


Hình 15: Sơ đồ mạch Schmitt Trigger

Đối với sơ đồ trên, một tín hiệu từ 1 chân encoder sẽ đi qua bộ đệm là con IC 74HC14 với ngõ ra đảo. Sau khi xử lý xong, tín hiệu sẽ được dẫn tới MCU một cách chính xác. Do mỗi encoder hay động cơ đều có 2 chân xung nên yêu cầu của luận văn này thì cần tới 4 sơ đồ như trên. Vì thế nhóm chúng em đã thi công phần mạch để có thể điều khiển được con lắc một cách chính xác

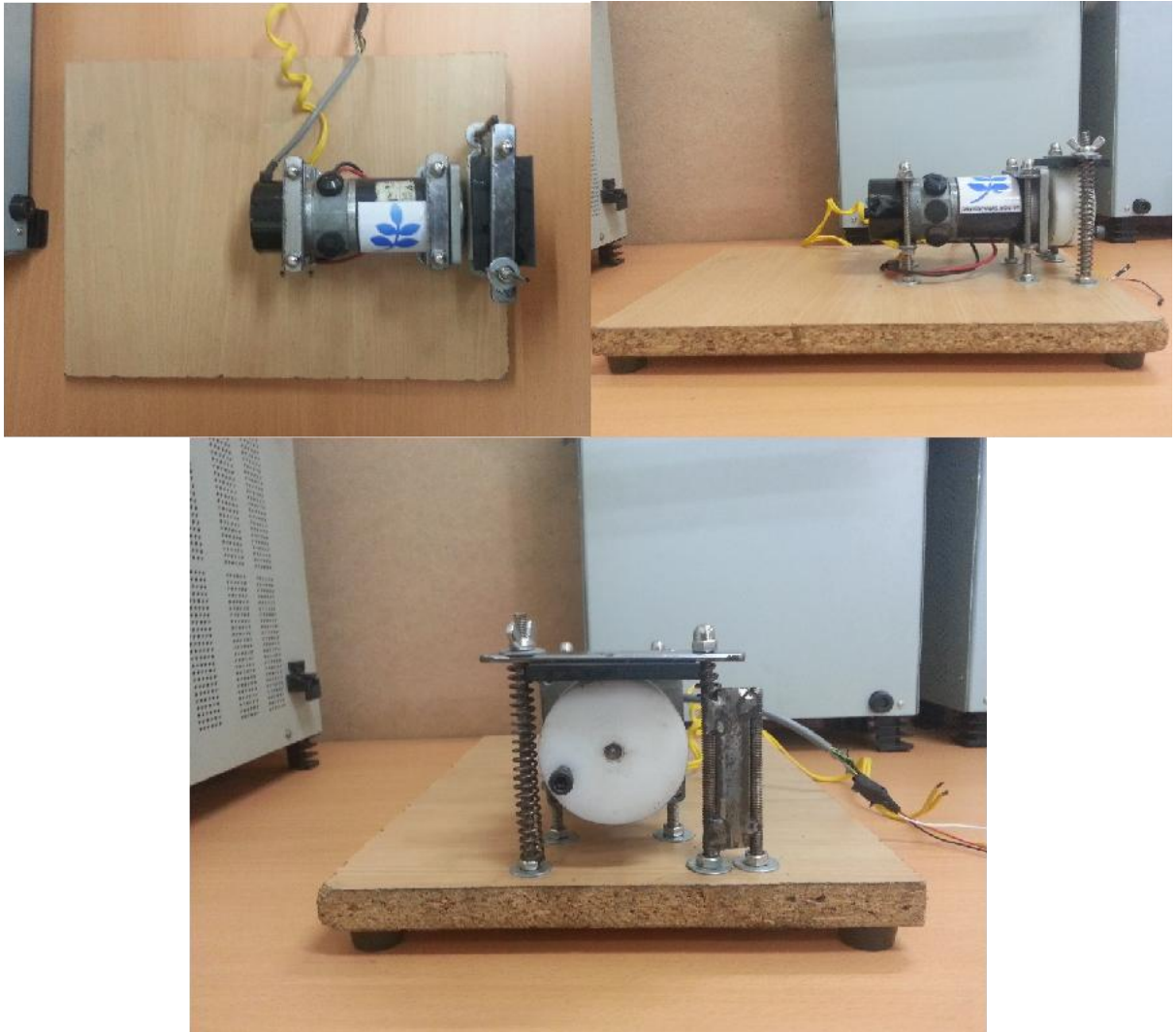


Hình 16: Schematic mạch Schmitt Trigger

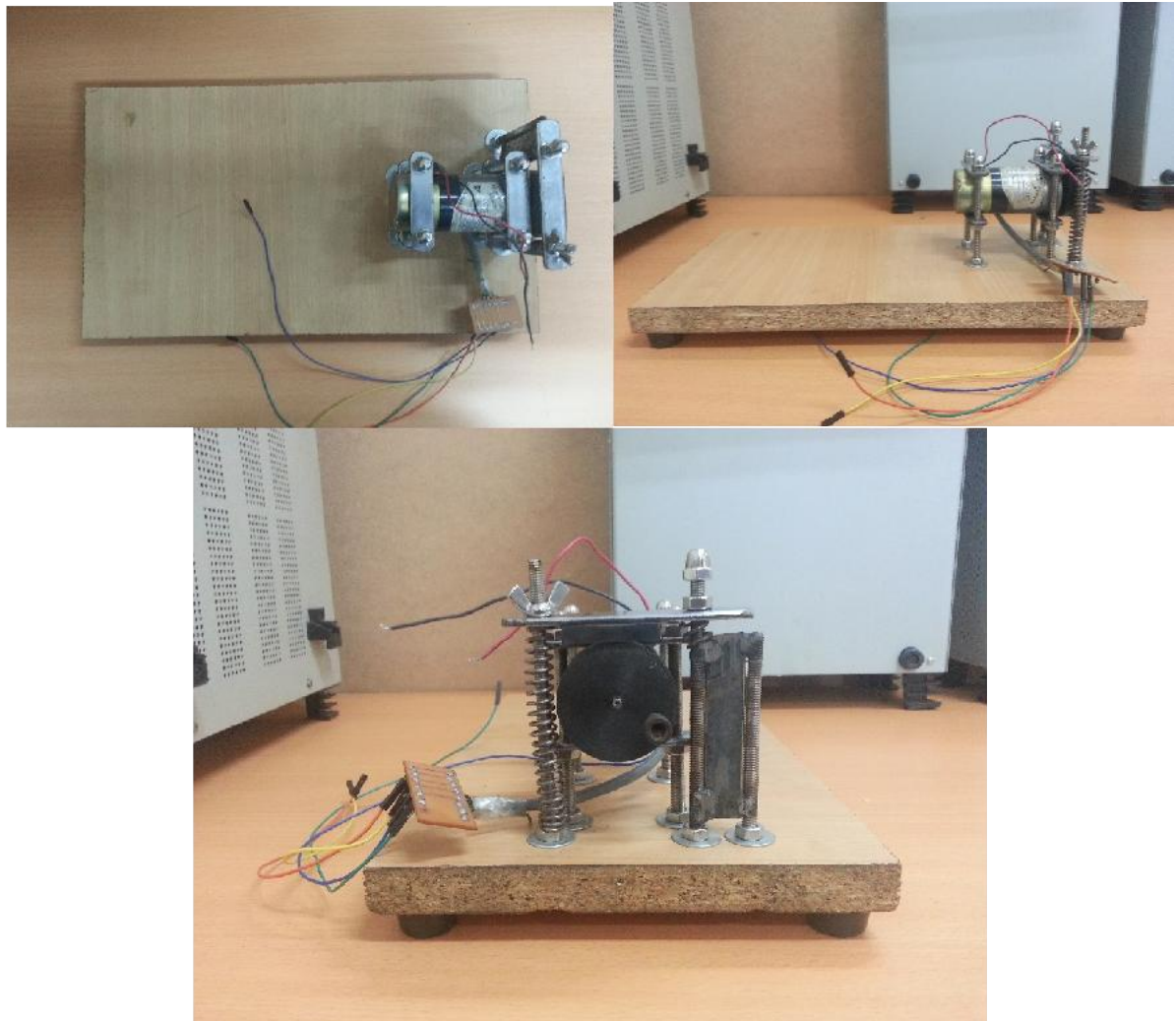


Hình 17: Mạch Schmitt Trigger

3.4 MÔ HÌNH ĐỘNG CƠ

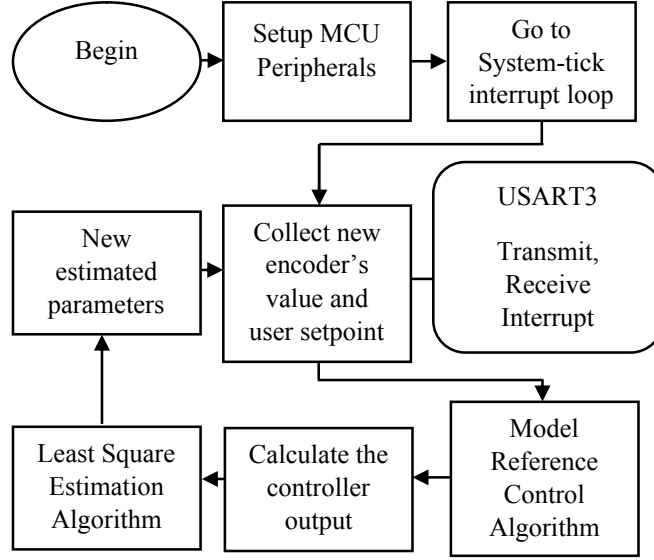


Hình 18: Mô hình động cơ 1 Encoder 1000 xung/vòng, điện áp 24VDC



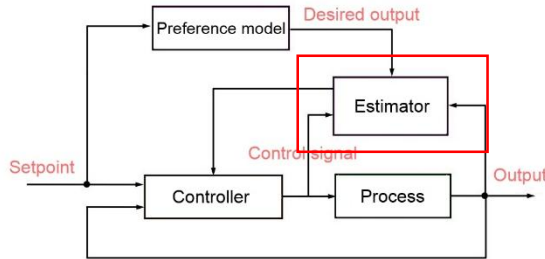
Hình 19: Mô hình động cơ 2 Encoder 200 xung/vòng, điện áp 12VDC

CHƯƠNG 4: XÂY DỰNG GIẢI THUẬT



Hình 20: Sơ đồ khối quá trình lập trình nhúng trong vi điều khiển

4.1 BỘ ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU ĐỆ QUY



Hình 21: Bộ ước lượng bình phương tối thiểu đệ

4.1.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

Ý nghĩa của giải thuật này là xác định được vận tốc ước lượng của động cơ là $\hat{y}(k)$ thông qua các thông số thu về từ encoder và setpoint trong ma trận $\varphi^T(k)$ và thông số ước lượng mô hình rời rạc trong ma trận $\hat{\theta}(k)$. Thông qua sự chênh lệch giữa vận tốc ước lượng và vận tốc thực tế, luật điều khiển sẽ đưa ra output phù hợp trong mục 4.3

$$\hat{y}(k) = \varphi^T(k)\hat{\theta}(k)$$

$$\begin{cases} y(k) + a_1y(k-1) + a_2y(k-2) = b_1u(k-1) + b_2u(k-2) \\ \theta(k) = [a_1(k) \ a_2(k) \ b_1(k) \ b_2(k)]^T \\ \varphi(k) = [-y(k-1) \ -y(k-2) \ u(k-1) \ u(k-2)]^T \end{cases}$$

Bước 1: Khởi tạo ma trận thông số khởi đầu $\theta(0)$ và ma trận covariance $P(0)$

$$\theta(0) = \text{random}(4,1)$$

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

Bước 2: Cập nhật ma trận trạng thái hệ thống $\varphi(k)$

$$\varphi(k) = [-y(k-1) - y(k-2) \quad u(k-1) \quad u(k-2)]^T$$

Bước 3: Tính toán ma trận gain Kalman $L(k)$ mới theo công thức:

$$L(k) = \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)}$$

Bước 4: Tính toán ma trận sai số ước lượng $\epsilon(k)$:

$$\epsilon(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)$$

Bước 5: Tính toán ma trận trạng thái hệ thống $\varphi(k)$ mới và lấy các hệ số ước lượng trạng thái động cơ

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)\epsilon(k)$$

$$\hat{\theta}(k) = [\hat{a}_1(k) \quad \hat{a}_2(k) \quad \hat{b}_1(k) \quad \hat{b}_2(k)]^T$$

Bước 6: Tính toán ma trận covariance P mới theo công thức

$$P(k) = \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi^T(k)\varphi(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \right] = \frac{1}{\lambda} [P(k-1) - L(k)\varphi(k)P(k-1)]$$

Bước 7: Quay về Bước 2, bước qua lần lấy mẫu và điều khiển mới

Bảng 3: Giải thích các thông số bộ ước lượng điều khiển vận tốc

Thông số	Giải thích
----------	------------

$u(k)$	Tín hiệu điện áp ra từ bộ điều khiển theo dạng rời rạc
$y(k)$	Tín hiệu vận tốc hoặc vị trí thực tế theo dạng rời rạc
Ma trận $\theta(k)$	Ma trận trạng thái hệ thống
Ma trận $\hat{\theta}(k)$	Ma trận trạng thái ước lượng thái hệ thống bởi bộ ước lượng
Ma trận $\varphi(k)$	Ma trận $\varphi(k)$ gồm tín hiệu điện áp ra từ bộ điều khiển, tín hiệu vận tốc hoặc vị trí thực tế, ở lần lấy mẫu trước
$\varepsilon(k)$	Sai số so với thực tế mà giải thuật ước lượng dự đoán
Ma trận $L(k)$	Ma trận $L(k)$ còn gọi là ma trận gain Kalman kết hợp với sai số dự đoán $\varepsilon(k)$ để tính toán ra thông số ước lượng của hệ thống
Ma trận $P(k)$	Ma trận $P(k)$ còn gọi là ma trận covariance, là cốt lõi của giải thuật ước lượng bình phương tối thiểu đệ quy
λ	Hằng số quên từ 0.95 đến 0.998 tùy hệ thống
Ma trận $\hat{\theta}(0)$	Chứa các hệ số khi bắt đầu giải thuật, chọn random từ 0 đến 1
Ma trận $\hat{P}(0)$	Chứa các hệ số khi bắt đầu giải thuật, chọn ma trận đơn vị I có đường chéo bằng 1

4.1.2 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ

Ý nghĩa của giải thuật này là xác định được vị trí ước lượng của động cơ là $\hat{y}(k)$ thông qua các thông số thu về từ encoder và setpoint trong ma trận $\varphi^T(k)$ và thông số ước lượng mô

hình rời rạc trong ma trận $\hat{\theta}(k)$. Thông qua sự chênh lệch giữa vận tốc ước lượng và vận tốc thực tế, luật điều khiển sẽ đưa ra output phù hợp trong mục 4.3

$$\hat{y}(k) = \varphi^T(k)\hat{\theta}(k)$$

$$\begin{cases} y(k) + a_1y(k-1) + a_2y(k-2) + a_3y(k-3) = b_1u(k-1) + b_2u(k-2) + b_3u(k-3) \\ \theta(k) = [a_1(k) \ a_2(k) \ a_3(k) \ b_1(k) \ b_2(k) \ b_3(k)]^T \\ \varphi(k) = [-y(k-1) \ -y(k-2) \ -y(k-3) \ u(k-1) \ u(k-2) \ u(k-3)]^T \end{cases}$$

Bước 1: Khởi tạo ma trận thông số khởi đầu $\theta(0)$ và ma trận covariance $P(0)$

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

$$\theta(0) = \text{random}(6,1)$$

Bước 2: Cập nhật ma trận trạng thái hệ thống $\varphi(k)$

$$\varphi(k) = [-y(k-1) \ -y(k-2) \ -y(k-3) \ u(k-1) \ u(k-2) \ u(k-3)]^T$$

Bước 3: Tính toán ma trận gain Kalman $L(k)$ mới theo công thức:

$$L(k) = \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)}$$

Bước 4: Tính toán ma trận sai số ước lượng $\epsilon(k)$:

$$\epsilon(k) = y(k) - \varphi^T(k)\hat{\theta}(k-1)$$

Bước 5: Tính toán ma trận trạng thái hệ thống $\varphi(k)$ mới và lấy các hệ số ước lượng trạng thái động cơ

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)\epsilon(k)$$

$$\hat{\theta}(k) = [\hat{a}_1(k) \ \hat{a}_2(k) \ \hat{a}_3(k) \ \hat{b}_1(k) \ \hat{b}_2(k) \ \hat{b}_3(k)]^T$$

Bước 6: Tính toán ma trận covariance P mới theo công thức

$$P(k) = \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi^T(k)\varphi(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \right] = \frac{1}{\lambda} [P(k-1) - L(k)\varphi(k)P(k-1)]$$

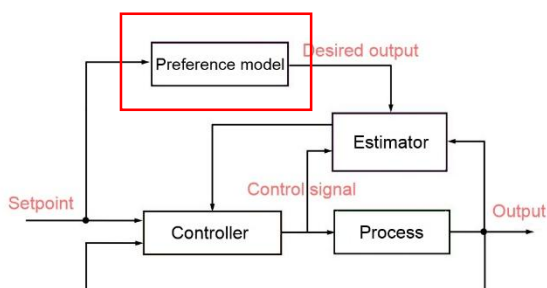
Bước 7: Quay về Bước 2, bước qua lần lấy mẫu và điều khiển mới

Bảng 4: Giải thích các thông số bộ ước lượng điều khiển vận tốc

Thông số	Giải thích
$u(k)$	Tín hiệu điện áp ra từ bộ điều khiển theo dạng rời rạc
$y(k)$	Tín hiệu vận tốc hoặc vị trí thực tế theo dạng rời rạc
Ma trận $\theta(k)$	Ma trận trạng thái hệ thống
Ma trận $\hat{\theta}(k)$	Ma trận trạng thái ước lượng thái hệ thống bởi bộ ước lượng
Ma trận $\varphi(k)$	Ma trận $\varphi(k)$ gồm tín hiệu điện áp ra từ bộ điều khiển, tín hiệu vận tốc hoặc vị trí thực tế, ở lần lấy mẫu trước
$\varepsilon(k)$	Sai số so với thực tế mà giải thuật ước lượng dự đoán
Ma trận $L(k)$	Ma trận $L(k)$ còn gọi là ma trận gain Kalman kết hợp với sai số dự đoán $\varepsilon(k)$ để tính toán ra thông số ước lượng của hệ thống
Ma trận $P(k)$	Ma trận $P(k)$ còn gọi là ma trận covariance, là cốt lõi của giải thuật ước lượng bình phương tối thiểu đệ quy
λ	Hằng số quên từ 0.95 đến 0.998 tùy hệ thống
Ma trận $\hat{\theta}(0)$	Chứa các hệ số khi bắt đầu giải thuật, chọn random từ 0 đến 1

Ma trận $\hat{P}(0)$	Chứa các hệ số khi bắt đầu giải thuật, chọn ma trận đơn vị I có đường chéo bằng 1
----------------------	---

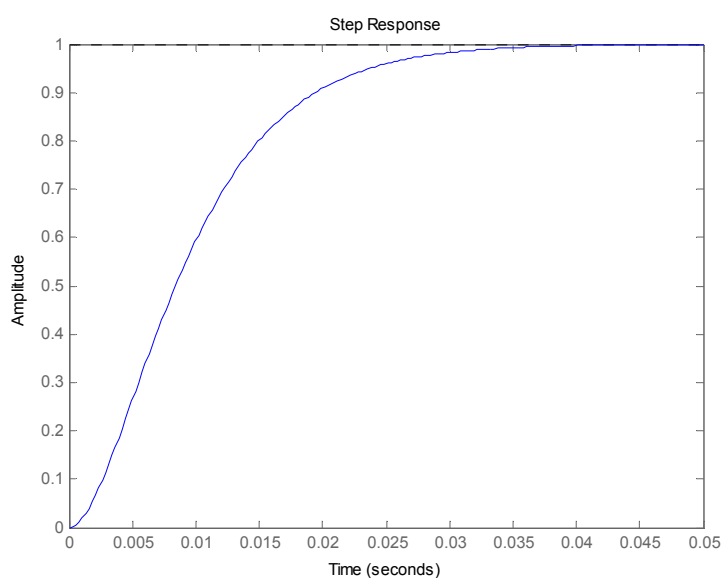
4.2 KHỐI MÔ HÌNH CHUẨN



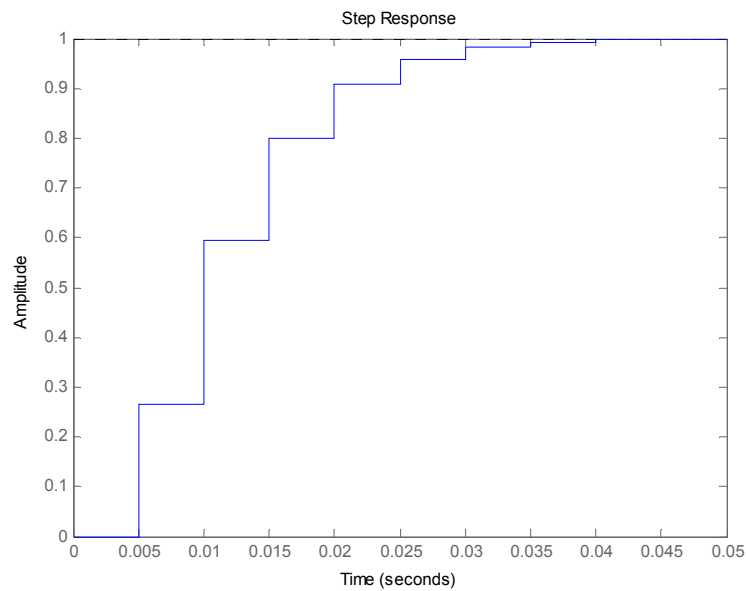
Hình 22: Khối mô hình chuẩn

4.2.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

$$\frac{Y_m(s)}{U_c(s)} = G(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2}$$



$$\frac{Y_m(z)}{U_c(z)} = G_d(z) = \frac{b_{1m}z + b_{2m}}{z^2 + a_{1m}z + a_{2m}}$$



Bước 1: Thiết kế mô hình chuẩn:

- Độ vọt lố (Percentage of Overshoot) là 0%:

$$\begin{cases} POT = e^{\frac{-\xi * \pi}{\sqrt{1-\xi^2}}} \\ POT = 0\% \end{cases}$$

$$\Rightarrow \xi = 1$$

- Thời gian xác lập T_{qd} theo thực nghiệm bằng 0.5 đến 0.1 chu kì lấy mẫu sẽ cho kết quả tốt nhất

$$\begin{cases} T_{qd} = \frac{4}{\omega_n * \xi} \\ T_{qd} = 0.0005 \end{cases}$$

$$\Rightarrow \omega_n = 8000$$

Bước 2: Tính toán các hệ số mô hình chuẩn $a1_m$, $a2_m$, $b1_m$, $b2_m$ bằng hàm Matlab:

```
>> G = tf([8000*8000],conv([1 8000*2 8000*8000],[1]))
```


Command Window

New to MATLAB? Watch this [Video](#), see [Lemos](#), or read [Getting started](#).

```

3 -

      6.4e11

      s^2 + 80000 s + 1.6e11 /

Continuous-time transfer function.

>> G = tf ([8000*8000],conv([1 8000*2 8000*8000],[1 ]))

G =

      6.4e07
      -----
      s^2 + 16000 s + 6.4e07

Continuous-time transfer function.

>> Gd = c2d(G, 0.005)

Gd =

      z + 1.657e-16
      -----
      z^2 - 8.497e-18 z + 1.805e-35

Sample time: 0.005 seconds
Discrete-time transfer function.

fx >>

```

Workspace

Select data to plot:

Name	Value	More
G	<1x1 tf>	
Gd	<1x1 tf>	

Command History

```

Gd = c2d(G, 0.005)
Gd = c2d(G, 0.005)
G1 = tf ([8000*8000],[1 8000*2 8000*8000])
G1d = c2d(G1, 0.005)
%-- 11/7/2013 5:25 PM --%
C = tf ([4000*4000*40000],c
G = tf ([8000*8000*80000],c
Gd = c2d(G, 0.005)
%-- 11/18/2013 4:17 PM --%
G = tf ([4000*4000*40000],c
G = tf ([4000*4000*40000],c
C = tf ([8000*8000],conv([1
Gd = c2d(G, 0.005)

```

Variable Editor - Gd

File Edit View Graphics Debug Desktop Window Help

Stack: Base No valid plots for: Gd.num(1,1)

Gd <1x1 tf>

Property	Value	Min	Max
num	<1x1 cell>		
den	<1x1 cell>		
Variable	's'		
InputDelay	0	0	0
OutputDelay	0	0	0
Ts	0.0050	0.0050	0.0050
TimeUnit	'seconds'		
InputName	<1x1 cell>		
InputUnit	<1x1 cell>		
InputGroup	<1x1 struct>		
OutputName	<1x1 cell>		
OutputUnit	<1x1 cell>		
OutputGroup	<1x1 struct>		
Name	'Gd'		
Notes	<1x1 cell>		
UseCase	1		

Variable Editor - Gd.num(1,1)

File Edit View Graphics Debug Desktop Window Help

Stack: Base No valid plots for: Gd.num(1,1)

Gd.num(1,1) <1x3 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0	1	1.6569e-16															

Variable Editor - Gd.den(1,1)

File Edit View Graphics Debug Desktop Window Help

Stack: Base No valid plots for: Gd.den(1,1)

Gd.den(1,1) <1x3 double>

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1	-8.4967e-18	1.8049e-35															

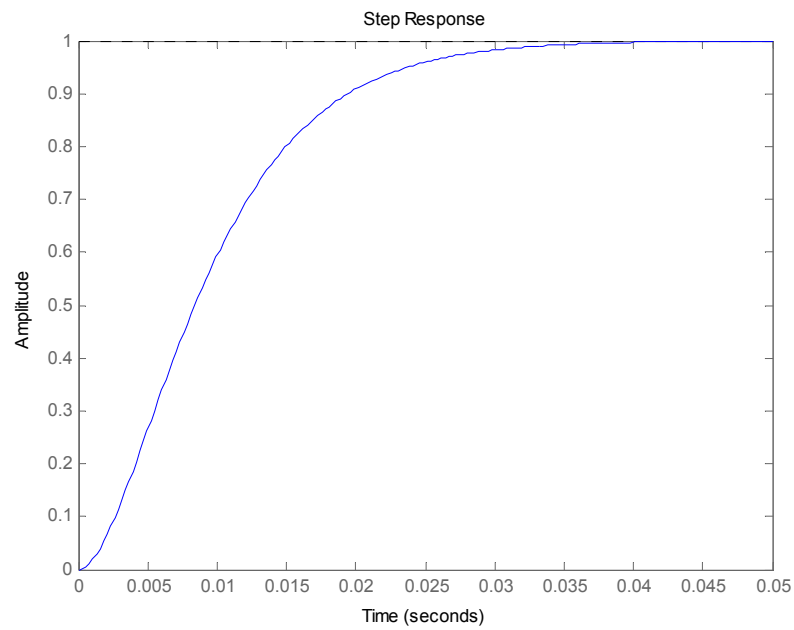
Hình 23: Tạo mô hình mẫu điều khiển vận tốc với Matlab

$$\frac{Y_m(z)}{U_c(z)} = G_d(z) = \frac{b_{1m}z + b_{2m}}{z^2 + a_{1m}z + a_{2m}}$$

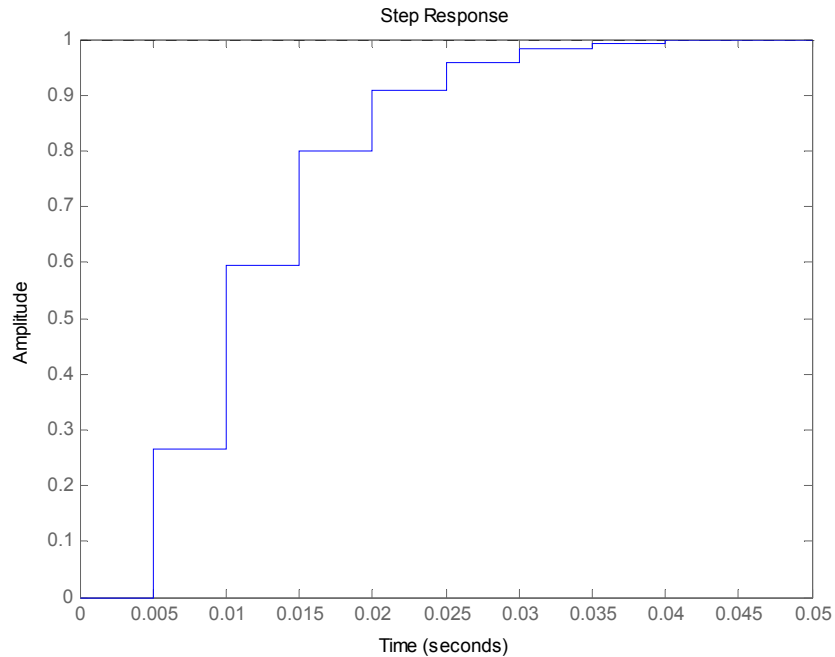
$$\rightarrow \frac{z + 1.65685815956384e - 16}{z^2 - 8.49670851058287e - 18z + 1.80485138785099e - 35}$$

4.2.2 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ

$$\frac{Y_m(s)}{U_c(s)} = G(s) = \frac{10 * \omega_n^3}{(s - 10\omega_n)(s^2 + 2\xi\omega_n s + \omega_n^2)}$$



$$\frac{Y_m(z)}{U_c(z)} = G_d(z) = \frac{b_1 m z^2 + b_2 m z + b_3 m}{z^3 + a_1 m z^2 + a_2 m z + a_3 m}$$



Bước 1: Thiết kế mô hình chuẩn:

- Độ vọt lô (Percentage of Overshoot) là 0%:

$$\begin{cases} POT = e^{\frac{-\xi * \pi}{\sqrt{1-\xi^2}}} \\ POT = 0\% \end{cases}$$

$$\Rightarrow \xi = 1$$

- Thời gian xác lập T_{qd} theo thực nghiệm bằng 0.5 đến 0.1 chu kì lấy mẫu sẽ cho kết quả tốt nhất

$$\begin{cases} T_{qd} = \frac{4}{\omega_n * \xi} \\ T_{qd} = 0.001 \end{cases}$$

$$\Rightarrow \omega_n = 4000$$

- Vì mô hình vị trí, điện áp động cơ có bậc ở mẫu là 3 nên cực thứ 3 ta chọn bằng 10 lần phần thực của cặp nghiệm phức quyết định để mô hình mẫu có đáp ứng tốt với đầu vào hàm nấc

Bước 2: Tính toán các hệ số mô hình chuẩn $a1_m, a2_m, b1_m, b2_m$ bằng hàm Matlab:

>> G = tf ([4000*4000*40000],conv([1 4000*2 4000*4000],[1 40000]))

The Command Window shows the following commands and outputs:

```
>> G = tf ( [4000*4000*40000],conv([1 4000*2 4000*4000],[1 40000]))

G =

      6.4e11
      -----
      s^3 + 48000 s^2 + 3.36e08 s + 6.4e11

Continuous-time transfer function.

>> Gd = c2d(G, 0.005)

Gd =

      z^2 + 4.372e-08 z + 5.245e-20
      -----
      z^3 - 4.122e-09 z^2 + 4.248e-18 z + 3.622e-41

Sample time: 0.005 seconds
Discrete-time transfer function.
```

The Workspace shows variables G and Gd, both of type 1x1 tf.

The Command History shows the sequence of commands entered:

```
G = tf ( [4000*4000*40000],c
G = tf ( [4000*4000*40000],c
G = tf ( [8000*8000],conv([1
Gd = c2d(G, 0.005)
roots([1 8000*2 8000*8000])
G = tf ( [4000*4000*40000],c
Gd = c2d(G, 0.005)
clr
clrc
G = tf ( [4000*4000*40000],c
Gd = c2d(G, 0.005)
```

Property	Value	Min	Max
num	<1x1 cell>		
den	<1x1 cell>		
Variable	'z'		
ioDelay	0	0	0
InputDelay	0	0	0
OutputDelay	0	0	0
Ts	0.0050	0.0050	0.0050
TimeUnit	'seconds'		
InputName	<1x1 cell>		
InputUnit	<1x1 cell>		
InputGroup	<1x1 struct>		
OutputName	<1x1 cell>		
OutputUnit	<1x1 cell>		
OutputGroup	<1x1 struct>		
Name	''		
Notes	<0x0 cell>		
UserData	[]		

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	1.0000	4.3717e-08	5.2449e-20														

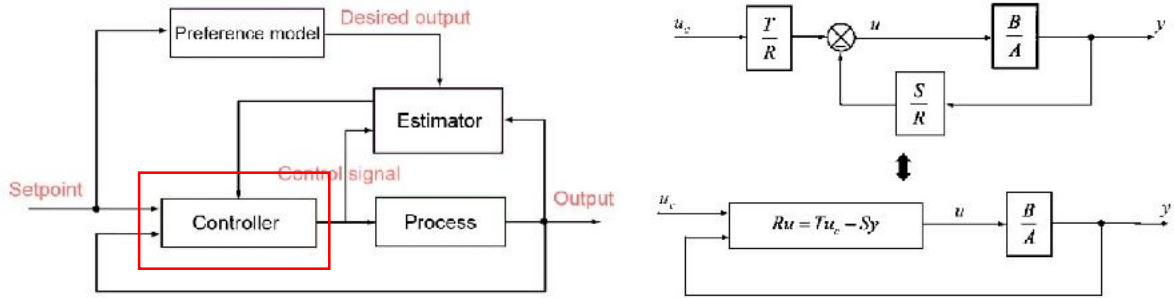
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-4.1223e-09	4.2484e-18	3.6221e-41														

Hình 24: Tạo mô hình mẫu điều khiển vị trí với Matlab

$$\frac{Y_m(z)}{U_c(z)} = G_d(z) = \frac{b_1 m z^2 + b_2 m z + b_3 m}{z^3 + a_1 m z^2 + a_2 m z + a_3 m}$$

$$\rightarrow \frac{0.999999999999985z^2 + 1.84514941606361e - 16z + 2.22821158995961e - 37}{z^3 - 8.49670851058608e - 18z^2 + 1.80485138784642e - 35z + 1.42296466898492e - 66}$$

4.3 BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN



Hình 25: Bộ điều khiển theo mô hình chuẩn

4.3.1 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

Đối tượng:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2}$$

Mô hình mẫu:

$$y_m(z) = \frac{B_m}{A_m} u_c(z) = \frac{b_{1m} z + b_{2m}}{z^2 + a_{1m} z + a_{2m}}$$

Bước 1: Phân tích B dưới dạng $B = B^+ B^-$

$$B = B^+ B^-$$

$$\begin{cases} B^+ = b_1 z + b_2 \\ B^- = 1 \end{cases}$$

Bước 2: Kiểm tra mô hình mẫu có thỏa mãn điều kiện cần để tồn tại lời giải hay không

$$B_m = B^- B'_m$$

$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

$$B_m = B'_m B^-$$

$$\Rightarrow B'_m = B_m = b_{1m}z + b_{2m}$$

$$[A_m] - [B_m] \geq [A] - [B] \quad (2 - 1 \geq 2 - 1)$$

Bước 3: Chọn bậc của A_0 thỏa mãn điều kiện tồn tại lời giải

$$\text{Bậc } A_0 \geq 2 \text{Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1$$

$$[A_0] = 2[A] - [A_m] - [B^+] - 1 = 2 * 2 - 2 - 1 - 1 = 0$$

$$\Rightarrow A_0 = 1$$

Bước 4: Chọn bậc của S và R_1

$$\text{Bậc } R_1 = \text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } A$$

$$\text{Bậc } S = \min\{[\text{Bậc } R_1 + \text{Bậc } B^+], [\text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } B^-]\}$$

$$[R_1] = [A_0] + [A_m] - [A] = 0 + 2 - 2 = 0$$

$$[S] = \min\{[R_1] + [B^+], [A_0] + [A_m] - [B^-]\}$$

$$= \min\{0 + 1, 0 + 2 - 0\} = 1$$

$$\begin{cases} R_1 = r_0 \\ S = s_0z + s_1 \end{cases}$$

Bước 5: Tính S và R_1 bằng cách giải phương trình:

$$AR_1 + BS = A_0A_m$$

$$AR_1 + B^-S = A_0A_m$$

$$\Rightarrow (z^2 + a_1z + a_2)r_0 + (s_0z + s_1) = z^2 + a_{1m}z + a_{2m}$$

$$\begin{cases} r_0 = 1 \\ s_0 = a_{1m} - a_1 \\ s_1 = a_{2m} - a_2 \end{cases}$$

$$R_1 = 1$$

Bước 6: Tính R và T

$$R = R_1 B^+$$

$$T = A_0 B'_m$$

$$S = (a_{1m} - a_1)z + (a_{2m} - a_2)$$

$$R = B^+ R_1 = b_1 z + b_2$$

$$T = A_0 B'_m = b_{1m} z + b_{2m}$$

Luật điều khiển

$$Ru = Tu_c - Sy$$

$$(b_1 z + b_2)u = (b_{1m} z + b_{2m})u_c - [(a_{1m} - a_1)z + (a_{2m} - a_2)]y$$

$$(b_1 + b_2 z^{-1})u = (b_{1m} + b_{2m} z^{-1})u_c - [(a_{1m} - a_1) + (a_{2m} - a_2)z^{-1}]y$$

$$u = \left(\frac{b_{1m}}{b_1} + \frac{b_{2m}}{b_1} z^{-1} \right) u_c - \left[\left(\frac{a_{1m}}{b_1} - \frac{a_1}{b_1} \right) + \left(\frac{a_{2m}}{b_1} - \frac{a_2}{b_1} \right) z^{-1} \right] y - \left(\frac{b_2}{b_1} z^{-1} \right) u$$

4.3.2 GIẢI THUẬT ƯỚC LƯỢNG TRONG TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ

Đối tượng:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}$$

Mô hình mẫu:

$$y_m(z) = \frac{Bm}{Am} u_c(z) = \frac{b_1 m z^2 + b_2 m z + b_3 m}{z^3 + a_1 m z^2 + a_2 m z + a_3 m}$$

Bước 1: Phân tích B dưới dạng $B = B^+ B^-$

$$B = B^+ B^-$$

$$\begin{cases} B^+ = b_1 z^2 + b_2 z + b_3 \\ B^- = 1 \end{cases}$$

Bước 2: Kiểm tra mô hình mẫu có thỏa mãn điều kiện cần để tồn tại lời giải hay không

$$B_m = B^- B'_m$$

$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

$$B_m = B'_m B^-$$

$$\Rightarrow B'_m = B_m = b_1 m z^2 + b_2 m z + b_3 m$$

$$[A_m] - [B_m] \geq [A] - [B] \quad (3 - 2 \geq 3 - 2)$$

Bước 3: Chọn bậc của A_0 thỏa mãn điều kiện tồn tại lời giải

$$\text{Bậc } A_0 \geq 2 \text{ Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1$$

$$[A_0] = 2[A] - [A_m] - [B^+] - 1$$

$$= 2 * 3 - 3 - 2 - 1 = 0$$

$$\Rightarrow A_0 = 1$$

Bước 4: Chọn bậc của S và R_1

$$\text{Bậc } R_1 = \text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } A$$

$$\text{Bậc } S = \min\{[\text{Bậc } R_1 + \text{Bậc } B^+], [\text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } B^-]\}$$

$$[R_1] = [A_0] + [A_m] - [A] = 0 + 3 - 3 = 0$$

$$[S] = \min\{[R_1] + [B^+], [A_0] + [A_m] - [B^-]\} = \min\{0 + 2, 0 + 3 - 0\} = 2$$

$$\begin{cases} R_1 = r_0 \\ S = s_0 z^2 + s_1 z + s_2 \end{cases}$$

Bước 5: Tính S và R_1 bằng cách giải phương trình:

$$AR_1 + BS = A_0 A_m$$

$$AR_1 + B^- S = A_0 A_m$$

$$\Rightarrow (z^3 + a_1 z^2 + a_2 z + a_3) r_0 + (s_0 z^2 + s_1 z + s_2) = (z^3 + a_1 m z^2 + a_2 m z + a_3 m)$$

$$r_0 = 1$$

$$\begin{cases} s_0 = a_{1m} - a_1 \\ s_1 = a_{2m} - a_2 \\ s_2 = a_{3m} - a_3 \end{cases}$$

$$R_1 = 1$$

Bước 6: Tính R và T

$$R = R_1 B^+$$

$$T = A_0 B'_m$$

$$S = (a_{1m} - a_1) z^2 + (a_{2m} - a_2) z + (a_{3m} - a_3)$$

$$R = B^+ R_1 = b_1 z^2 + b_2 z + b_3$$

$$T = A_0 B'_m = b_1 m z^2 + b_2 m z + b_3 m$$

Luật điều khiển

$$Ru = Tu_c - Sy$$

$$\begin{aligned} (b_1 z^2 + b_2 z + b_3) u \\ = (b_1 m z^2 + b_2 m z + b_3 m) u_c - [(a_{1m} - a_1) z^2 + (a_{2m} - a_2) z + (a_{3m} - a_3)] y \end{aligned}$$

$$\begin{aligned}
& (b_1 + b_2 z^{-1} + b_3 z^{-2})u \\
& = (b_{1m} + b_{2m} z^{-1} + b_{3m} z^{-2})u_c - [(a_{1m} - a_1) + (a_{2m} - a_2)z^{-1} - (a_{3m} \\
& - a_3)z^{-2}]y
\end{aligned}$$

$$\begin{aligned}
u = & \left(\frac{b_{1m}}{b_1} + \frac{b_{2m}}{b_1} z^{-1} + \frac{b_{3m}}{b_1} z^{-2} \right) u_c \\
& - \left[\left(\frac{a_{1m}}{b_1} - \frac{a_1}{b_1} \right) + \left(\frac{a_{2m}}{b_1} - \frac{a_2}{b_1} \right) z^{-1} - \left(\frac{a_{3m}}{b_1} - \frac{a_3}{b_1} \right) z^{-2} \right] y - \left(\frac{b_2}{b_1} z^{-1} \right. \\
& \left. + \frac{b_3}{b_1} z^{-2} \right) u
\end{aligned}$$

CHƯƠNG 5: NHỮNG CẢI TIẾN TRONG GIẢI THUẬT

5.1 CẢI TIẾN BỘ ƯỚC LƯỢNG BÌNH PHƯƠNG TỐI THIỂU ĐỆ QUY

5.1.1 HỆ SỐ QUÊN THAY ĐỔI THEO THÔNG SỐ CỦA HỆ THỐNG

Hệ số quên trong công thức bộ ước lượng cơ bản có hệ số λ là một hằng số. Trước hết, chúng ta cần làm rõ chức năng vô cùng quan trọng của hệ số quên trong giải thuật điều khiển này:

Hệ số quên làm cho những thông số của ma trận covariance P không trở nên quá nhỏ làm mất tính thích nghi. Sự đặc trưng của tính thích nghi thể hiện qua thông số N gọi là độ dài của bộ nhớ thích nghi:

$$N = \frac{1}{1 - \lambda}$$

Hệ thống giải thuật thích nghi dùng những thông số từ N mẫu trước đó để tính toán và ước lượng thông số hiện tại của động cơ. Bởi vì đây là giải thuật hồi quy, nó không thể thích nghi và tính toán tất cả những thông số của động cơ từ lúc hoạt động đến vô cùng, đơn giản vì không một vi xử lý nào có thể tính toán số lượng phép tính tăng tuyến tính đến vô cực cả.

Ở bảng thông tin thông số trước, giải thuật cơ bản khuyến nghị hệ số λ là hằng số từ 0.95 đến 0.998 tùy hệ thống tức là giải thuật sẽ thích nghi từ 20 đến 500 thông số trước đó của động cơ, điều này có hai nhược điểm:

1. Chúng ta không biết hệ thống nào biến đổi nhanh chậm ra sao mà chọn được số lượng mẫu thích nghi phù hợp, có những hệ thống thay đổi nhanh và liên tục theo thời gian (ví dụ như động cơ cánh tay máy gấp vật thể liên tục, hoặc có nhiễu mạnh trong môi

trường) trong khi có những hệ thống thông số thay đổi rất ít (động cơ thang máy, xe lăn cho người già). Điều này làm giảm tính thích nghi của bộ điều khiển được gọi là thích nghi.

2. Khi những thông số của hệ thống thay đổi không quá nhiều (như trong trường hợp động cơ đạt xác lập), ma trận P có xu hướng ít thay đổi trong khi sau mỗi chu kỳ lấy mẫu, ma trận P lại được chia cho hệ số quên λ bé hơn 1, điều này dẫn đến một lúc nào đó, giá trị của P đạt đến vô cùng và hệ thống trở nên bất ổn định.

$$P(k) = \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi^T(k)\varphi(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \right] = \frac{1}{\lambda} [P(k-1) - L(k)\varphi(k)P(k-1)]$$

Để giải quyết vấn đề này, hệ số λ cần được tự động điều chỉnh theo một giải thuật mới:

$$\lambda(k) = \begin{cases} 1 - \frac{[1 - \varphi^T(k-1)L(k)]\epsilon(k)^2}{\Sigma_0} & \text{Nếu } \lambda(k) > \lambda(k)_{\min} \\ \lambda(k)_{\min} & \text{Nếu } \lambda(k) < \lambda(k)_{\min} \end{cases}$$

Với $\lambda(k)_{\min}$ từ 0.9 đến 0.95, chúng ta phải đặt hệ số $\lambda(k)_{\min}$ vì theo giải thuật này, đôi lúc $\lambda(k)$ xuống đến âm hoặc quá bé nên không mang tính thích nghi. Và Σ_0 là một hằng số vừa đủ lớn dựa vào thực nghiệm, Σ_0 quy định tốc độ thay đổi của $\lambda(k)$.

Tóm lại, giải thuật này làm biến mất hiện tượng đôi lúc động cơ bị giật khi các thông số thu thập được thay đổi không đáng kể.

5.1.2 GIẢI THUẬT ƯỚC LƯỢNG CÓ TÍNH ĐẾN ĐỘ TRỄ CỦA ĐỘNG CƠ

Như ta đã biết, có một sự khác nhau không hề nhỏ giữa mô hình thực tế và mô hình xây dựng trong lý thuyết. Một trong những tác nhân khác biệt đó chính là độ trễ động cơ, là thời gian mà sự thay đổi điện áp bắt đầu ảnh hưởng đến vận tốc hoặc vị trí của động cơ.

Trong mô hình thực tế, ngoài sự trễ gây ra bởi phản cảm và phản ứng của động cơ, chúng ta còn gặp trễ gây ra bởi các thiết bị điều khiển như mạch cầu H, sự truyền nhận tín hiệu điều khiển giữa vi điều khiển và máy vi tính. Chính vì đó, cần thay đổi giải thuật gốc để giải quyết tác nhân này, làm bộ điều khiển được tốt hơn.

Khoảng thời gian giữa lúc điện áp thay đổi đến lúc động cơ đáp ứng gọi là d , d được thể hiện rõ qua công thức sau đây:

$$y(k) = \varphi^T(k-d)\theta(k) \text{ (ở giải thuật gốc: } d=0\text{)}$$

Lúc này, toàn bộ công thức ước lượng được viết lại như sau, theo kinh nghiệm thực tế của tác giả, hệ số d sẽ làm điều khiển vị trí rất hoàn hảo, d nên chọn từ 1, 2 là hợp lý, lớn hơn hoặc bé hơn con số này sẽ gây rung lắc cho động cơ khi xác lập.

5.1.2A TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

Ý nghĩa của giải thuật này là xác định được vận tốc ước lượng của động cơ là $\hat{y}(k)$ thông qua các thông số thu về từ encoder và setpoint trong ma trận $\varphi^T(k)$ và thông số ước lượng mô hình rời rạc trong ma trận $\hat{\theta}(k)$. Thông qua sự chênh lệch giữa vận tốc ước lượng và vận tốc thực tế, luật điều khiển sẽ đưa ra output phù hợp trong mục 4.3

$$\hat{y}(k) = \varphi^T(k-d)\hat{\theta}(k)$$

$$\begin{cases} y(k) + a_1y(k-1) + a_2y(k-2) = b_1u(k-1) + b_2u(k-2) \\ \theta(k) = [a_1(k) \ a_2(k) \ b_1(k) \ b_2(k)]^T \\ \varphi(k) = [-y(k-1) - y(k-2) \ u(k-1) \ u(k-2)]^T \end{cases}$$

Bước 1: Khởi tạo ma trận thông số khởi đầu $\theta(0)$ và ma trận covariance $P(0)$

$$\theta(0) = \text{random}(4,1)$$

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

Bước 2: Cập nhật ma trận trạng thái hệ thống $\varphi(k)$

$$\varphi(k) = [-y(k-1) - y(k-2) \ u(k-1) \ u(k-2)]^T$$

Bước 3: Tính toán ma trận gain Kalman $L(k)$ mới theo công thức:

$$L(k) = \frac{P(k-1)\varphi(k-d)}{\lambda + \varphi^T(k-d)P(k-1)\varphi(k-d)}$$

Bước 4: Tính toán ma trận sai số ước lượng $\epsilon(k)$:

$$\epsilon(k) = y(k) - \varphi^T(k-d)\hat{\theta}(k-1)$$

Bước 5: Tính toán ma trận trạng thái hệ thống $\varphi(k)$ mới và lấy các hệ số ước lượng trạng thái động cơ

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)\epsilon(k)$$

$$\hat{\theta}(k) = [\hat{a}_1(k) \ \hat{a}_2(k) \ \hat{b}_1(k) \ \hat{b}_2(k)]^T$$

Bước 6: Tính toán ma trận covariance P mới theo công thức

$$\begin{aligned} P(k) &= \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi^T(k-d)\varphi(k-d)P(k-1)}{\lambda + \varphi^T(k-d)P(k-1)\varphi(k-d)} \right] \\ &= \frac{1}{\lambda} [P(k-1) - L(k)\varphi(k-d)P(k-1)] \end{aligned}$$

Bước 7: Quay về Bước 2, bước qua lần lấy mẫu và điều khiển mới

5.1.2B TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ

Ý nghĩa của giải thuật này là xác định được vị trí ước lượng của động cơ là $\hat{y}(k)$ thông qua các thông số thu về từ encoder và setpoint trong ma trận $\varphi^T(k)$ và thông số ước lượng mô hình rời rạc trong ma trận $\hat{\theta}(k)$. Thông qua sự chênh lệch giữa vận tốc ước lượng và vận tốc thực tế, luật điều khiển sẽ đưa ra output phù hợp trong mục 4.3

$$\hat{y}(k) = \varphi^T(k-d)\hat{\theta}(k)$$

$$\begin{cases} y(k) + a_1y(k-1) + a_2y(k-2) + a_3y(k-3) = b_1u(k-1) + b_2u(k-2) + b_3u(k-3) \\ \theta(k) = [a_1(k) \ a_2(k) \ a_3(k) \ b_1(k) \ b_2(k) \ b_3(k)]^T \\ \varphi(k) = [-y(k-1) - y(k-2) - y(k-3) \ u(k-1) \ u(k-2) \ u(k-3)]^T \end{cases}$$

Bước 1: Khởi tạo ma trận thông số khởi đầu $\theta(0)$ và ma trận covariance $P(0)$

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

$$\theta(0) = \text{random}(6,1)$$

Bước 2: Cập nhật ma trận trạng thái hệ thống $\varphi(k)$

$$\varphi(k) = [-y(k-1) - y(k-2) - y(k-3) \ u(k-1) \ u(k-2) \ u(k-3)]^T$$

Bước 3: Tính toán ma trận gain Kalman $L(k)$ mới theo công thức:

$$L(k) = \frac{P(k-1)\varphi(k-d)}{\lambda + \varphi^T(k-d)P(k-1)\varphi(k-d)}$$

Bước 4: Tính toán ma trận sai số ước lượng $\epsilon(k)$:

$$\epsilon(k) = y(k) - \varphi^T(k-d)\hat{\theta}(k-1)$$

Bước 5: Tính toán ma trận trạng thái hệ thống $\varphi(k)$ mới và lấy các hệ số ước lượng trạng thái động cơ

$$\hat{\theta}(k) = \hat{\theta}(k-1) + L(k)\epsilon(k)$$

$$\hat{\theta}(k) = [\hat{a}_1(k) \ \hat{a}_2(k) \ \hat{a}_3(k) \ \hat{b}_1(k) \ \hat{b}_2(k) \ \hat{b}_3(k)]^T$$

Bước 6: Tính toán ma trận covariance P mới theo công thức

$$\begin{aligned} P(k) &= \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi^T(k-d)\varphi(k-d)P(k-1)}{\lambda + \varphi^T(k-d)P(k-1)\varphi(k-d)} \right] \\ &= \frac{1}{\lambda} [P(k-1) - L(k)\varphi(k-d)P(k-1)] \end{aligned}$$

Bước 7: Quay về Bước 2, bước qua lần lấy mẫu và điều khiển mới

Tóm lại, giải thuật này làm biến mất hiện tượng rung động cơ khi xác lập khi điều khiển vị trí.

5.1.3 GIẢI THUẬT NHÂN CHUỖI MA TRẬN VỚI SỐ PHÉP TOÁN TỐI ƯU

Rất nhiều báo cáo khoa học bỏ qua vấn đề thực nghiệm này, nhân chia và tính toán với số ma trận tương đối nhiều của “giải thuật ước lượng bình phương tối thiểu đệ quy, hồi quy, cải tiến, không nghịch đảo ma trận” là một thách thức lớn với những vi xử lý xung nhịp thấp và kìm hãm hiệu năng của những vi xử lý nhiều khả năng hơn. Hãy cùng tác giả xem qua ví dụ sau đây để thấy được sự khác biệt rõ rệt giữa việc nhân chia chuỗi ma trận có giải thuật và việc nhân chia chuỗi ma trận một cách ngẫu nhiên.

A là ma trận 10×30 , B là 30×5 , C là 5×60

Chúng ta có thể nhân chia các ma trận theo thứ tự bất kì miễn là giữ nguyên dạng chuỗi ma trận:

$$(ABC)D = (AB)(CD) = A(BCD) = A(BC)D$$

Xét 2 cách nhân chuỗi ma trận ABC:

Bảng 5: So sánh hai giải thuật nhân chuỗi ma trận

Phép nhân ma trận	Số phép nhân	Số phép cộng	Số phép toán floating point
Chiều của 2 ma trận lần lượt là: MxN và NxL	MNL	ML(N-1)	2MNL - ML
Thứ tự (AB)C	4500	3850	8350
Thứ tự A(BC)	27000	24600	51600
$\{[(P(k-1)\varphi^T(k))\varphi(k)]P(k-1)\}$	288	210	498
$[P(k-1)\varphi^T(k)][\varphi(k)P(k-1)]$	108	60	168

Như ta đã thấy, cách tính toán ma trận covariance P thứ 2 giúp chúng ta đưa số phép tính floating point chỉ còn **33.73%** so với lúc ban đầu.

5.1.4 GIẢI THUẬT THAY ĐỔI MA TRẬN KHỞI ĐẦU P(0)

Giá trị khởi đầu của ma trận covariance P được chọn là bội số của ma trận đơn vị thay vì là chính ma trận đơn vị. Nếu hệ thống thay đổi liên tục hoặc những thông số khó đoán biết trước, các bạn nên chọn r “lớn”, nếu hệ thống của bạn ổn định và thay đổi chậm, hệ số “r” nhỏ sẽ có ích cho bạn.

$$P(0) = \begin{bmatrix} r & 0 & 0 & 0 & 0 & 0 \\ 0 & r & 0 & 0 & 0 & 0 \\ 0 & 0 & r & 0 & 0 & 0 \\ 0 & 0 & 0 & r & 0 & 0 \\ 0 & 0 & 0 & 0 & r & 0 \\ 0 & 0 & 0 & 0 & 0 & r \end{bmatrix}$$

Khoảng từ 100 đến 1000 của r được cho là “lớn”, lúc đó tín hiệu điều khiển của động cơ sẽ thay đổi nhanh và liên tục, điều này có ưu điểm là tốc độ thích nghi, đổi lại nhược điểm là sự ổn định của hệ thống.

Ngược lại với giá trị từ 1 đến 10, r được cho là “nhỏ”. Lớn và nhỏ ở đây tùy thuộc những hệ thống khác nhau. Lúc này tín hiệu điều khiển của động cơ có xu hướng thay đổi chậm hơn, hệ thống thích nghi chậm hơn nhưng ổn định hơn.

Công thức sau đây thể hiện toàn bộ những gì đã giải thích ở trên một cách thuyết phục hơn theo cơ sở toán học:

$$P(k) = \left[P^{-1}(0) + \sum_1^k \varphi(k) \varphi^T(k) \right]^{-1}$$

5.1.5 GIẢI THUẬT THAY ĐỔI MA TRẬN THÔNG SỐ KHỞI ĐẦU $\theta(0)$

Người lập trình có thể cho những thông số khởi đầu một giá trị bất kì, tuy nhiên, nếu hệ thống được mô tả gần đúng với thực tế ngay từ đầu thì vẫn tốt hơn.

Theo kinh nghiệm thực tế rút ra từ mô hình động cơ, tác giả khuyến nghị chọn hệ số $a_1 = -1$ và $b_1 = 1$, các hệ số còn lại bằng 0. Các bạn có thể chọn những thông số theo kiểu khác và đóng góp thêm cho mã nguồn mở của bộ điều khiển tại www.payitforward.edu.vn.

5.2 CẢI TIẾN KHỐI MÔ HÌNH CHUẨN

5.2.1 GIẢI THUẬT CHỐNG TRÀN DO LỖI CHIA CHO 0 TRONG KHỐI MÔ HÌNH CHUẨN

Cùng nhìn lại giải thuật điều khiển theo mô hình chuẩn đối với

Điều khiển thích nghi vận tốc động cơ:

$$u = \left(\frac{b_{1m}}{b_1} + \frac{b_{2m}}{b_1} z^{-1} \right) u_c - \left[\left(\frac{a_{1m}}{b_1} - \frac{a_1}{b_1} \right) + \left(\frac{a_{2m}}{b_1} - \frac{a_2}{b_1} \right) z^{-1} \right] y - \left(\frac{b_2}{b_1} z^{-1} \right) u$$

Điều khiển thích nghi vị trí động cơ:

$$\begin{aligned} u = & \left(\frac{b_{1m}}{b_1} + \frac{b_{2m}}{b_1} z^{-1} + \frac{b_{3m}}{b_1} z^{-2} \right) u_c \\ & - \left[\left(\frac{a_{1m}}{b_1} - \frac{a_1}{b_1} \right) + \left(\frac{a_{2m}}{b_1} - \frac{a_2}{b_1} \right) z^{-1} - \left(\frac{a_{3m}}{b_1} - \frac{a_3}{b_1} \right) z^{-2} \right] y - \left(\frac{b_2}{b_1} z^{-1} \right. \\ & \left. + \frac{b_3}{b_1} z^{-2} \right) u \end{aligned}$$

Ta nhận thấy rằng hệ số b_1 phải được giới hạn khác 0 để tránh lỗi divide by zero, nếu $b_1 = 0$ thì $b_1 = z^{-1} b_1$.

Điều này rất quan trọng với sự ổn định của hệ thống, tuy nhiên lại không có tài liệu nào đề cập đến vấn đề này, vì thế đôi lúc xuất hiện lỗi khó hiểu trong lúc bộ điều khiển hoạt động. Giải thuật càng nhiều thì chúng ta càng phải cẩn thận với những lỗi nhỏ nhất như thế này.

5.3 TỐI ƯU HÓA BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN

5.3.1 THÊM MỘT KHÂU TÍCH PHÂN VÀO KHỐI R

Để hệ thống bền vững hơn với nhiễu tần số thấp và tránh những sai số ở miền này, một khâu tích phân được kiến nghị thêm vào khối R.

$$R = (z - 1)^v R_1$$

Luật điều khiển được viết lại như sau:

5.3.1A TRƯỜNG HỢP ĐIỀU KHIỂN VẬN TỐC

Đối tượng:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2}$$

Mô hình mẫu:

$$y_m(z) = \frac{B_m}{A_m} u_c(z) = \frac{b_{1m} z + b_{2m}}{z^2 + a_{1m} z + a_{2m}}$$

Bước 1: Phân tích B dưới dạng $B = B^+ B^-$

$$B = B^+ B^-$$

$$\begin{cases} B^+ = b_1 z + b_2 \\ B^- = 1 \end{cases}$$

Bước 2: Kiểm tra mô hình mẫu có thỏa mãn điều kiện cần để tồn tại lời giải hay không

$$B_m = B^- B'_m$$

$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

$$B_m = B'_m B^-$$

$$\Rightarrow B'_m = B_m = b_{1m} z + b_{2m}$$

$$[A_m] - [B_m] \geq [A] - [B] \quad (2 - 1 \geq 2 - 1)$$

Bước 3: Chọn bậc của A_0 thỏa mãn điều kiện tồn tại lời giải

$$\text{Bậc } A_0 \geq 2 \text{ Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1 + v$$

$$[A_0] = 2[A] - [A_m] - [B^+] - 1 = 2 * 2 - 2 - 1 - 1 + 1 = 1$$

$$\Rightarrow A_0 = z$$

Bước 4: Chọn bậc của S và R_1

$$\text{Bậc } R_1 = \text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } A - v$$

$$\text{Bậc } S < \text{Bậc } A + v$$

$$[R_1] = [A_0] + [A_m] - [A] - v = 1 + 2 - 2 - 1 = 0$$

$$[S] < 2 + 1 = 3$$

$$\text{Vậy } [S] = 2$$

$$\begin{cases} R_1 = r_0 \\ S = s_0 z^2 + s_1 z + s_2 \end{cases}$$

Bước 5: Tính S và R_1 bằng cách giải phương trình:

$$AR_1(z-1)^v + BS = A_0 A_m$$

$$AR_1(z-1)^1 + B^- S = A_0 A_m$$

$$\Rightarrow (z-1)(z^2 + a_1 z + a_2)r_0 + (s_0 z^2 + s_1 z + s_2) = z^3 + a_{1m} z^2 + a_{2m} z$$

$$r_0 = 1$$

$$\begin{cases} s_0 = a_{1m} - a_1 + 1 \\ s_1 = a_{2m} - a_2 + a_1 \\ s_2 = a_2 \end{cases}$$

$$R_1 = z - 1$$

Bước 6: Tính R và T

$$R = R_1 B^+$$

$$T = A_0 B'_m$$

$$S = (a_{1m} - a_1 + 1)z^2 + (a_{2m} - a_2 + a_1)z + (a_2)$$

$$R = B^+ R_1 = (z - 1)(b_1 z + b_2)$$

$$T = A_0 B'_m = z(b_{1m} z + b_{2m})$$

Luật điều khiển

$$Ru = Tu_c - Sy$$

$$\begin{aligned} [b_1 z^2 + (b_2 - b_1)z - b_2]u \\ = (b_{1m} z^2 + b_{2m} z)u_c - [(a_{1m} - a_1 + 1)z^2 + (a_{2m} - a_2 + a_1)z + (a_2)]y \end{aligned}$$

$$\begin{aligned} [b_1 + (b_2 - b_1)z^{-1} - b_2 z^{-2}]u \\ = (b_{1m} + b_{2m} z^{-1})u_c - [(a_{1m} - a_1 + 1) + (a_{2m} - a_2 + a_1)z^{-1} + a_2 z^{-2}]y \end{aligned}$$

$$u = \left(\frac{b_{1m} + b_{2m}}{b_1} z^{-1} \right) u_c - \left[\frac{(a_{1m} - a_1 + 1)}{b_1} + \frac{(a_{2m} - a_2 + a_1)z^{-1}}{b_1} + \frac{a_2 z^{-2}}{b_1} \right] y - \left[\frac{(b_2 - b_1)}{b_1} z^{-1} - \frac{b_2 z^{-2}}{b_1} \right] u$$

5.3.1B TRƯỜNG HỢP ĐIỀU KHIỂN VỊ TRÍ

Đối tượng:

$$y(z) = \frac{B}{A} u(z) = \frac{b_1 z^2 + b_2 z + b_3}{z^3 + a_1 z^2 + a_2 z + a_3}$$

Mô hình mẫu:

$$y_m(z) = \frac{Bm}{Am} u_c(z) = \frac{b_1 m z^2 + b_2 m z + b_3 m}{z^3 + a_1 m z^2 + a_2 m z + a_3 m}$$

Bước 1: Phân tích B dưới dạng $B = B^+ B^-$

$$B = B^+ B^-$$

$$\begin{cases} B^+ = b_1 z^2 + b_2 z + b_3 \\ B^- = 1 \end{cases}$$

Bước 2: Kiểm tra mô hình mẫu có thỏa mãn điều kiện cần để tồn tại lời giải hay không

$$B_m = B^- B'_m$$

$$\text{Bậc } A_m - \text{Bậc } B_m \geq \text{Bậc } A - \text{Bậc } B$$

$$B_m = B'_m B^-$$

$$\Rightarrow B'_m = B_m = b_1 m z^2 + b_2 m z + b_3 m$$

$$[A_m] - [B_m] \geq [A] - [B] \quad (3 - 2 \geq 3 - 2)$$

Bước 3: Chọn bậc của A_0 thỏa mãn điều kiện tồn tại lời giải

$$\text{Bậc } A_0 \geq 2 \text{ Bậc } A - \text{Bậc } A_m - \text{Bậc } B^+ - 1 + v$$

$$[A_0] = 2[A] - [A_m] - [B^+] - 1 + 1$$

$$= 2 * 3 - 3 - 2 - 1 + 1 = 1$$

$$\Rightarrow A_0 = z$$

Bước 4: Chọn bậc của S và R_1

$$\text{Bậc } R_1 = \text{Bậc } A_0 + \text{Bậc } A_m - \text{Bậc } A - v$$

$$\text{Bậc } S < \text{Bậc } A + v$$

$$[R_1] = [A_0] + [A_m] - [A] - v = 1 + 3 - 3 - 1 = 0$$

$$[S] < [A] + v = 3 + 1 = 4$$

$$\begin{cases} R_1 = r_0 \\ S = s_0 z^3 + s_1 z^2 + s_2 z^1 + s_3 \end{cases}$$

Bước 5: Tính S và R_1 bằng cách giải phương trình:

$$AR_1(z-1)^v + BS = A_0A_m$$

$$AR_1(z-1)^1 + B^-S = A_0A_m$$

$$\Rightarrow (z-1)(z^3 + a_1z^2 + a_2z + a_3)r_0 + (s_0z^3 + s_1z^2 + s_2z^1 + s_3) \\ = z(z^3 + a_1mz^2 + a_2mz + a_3m)$$

$$r_0 = 1$$

$$\begin{cases} s_0 = a_{1m} - a_1 + 1 \\ s_1 = a_{2m} - a_2 + a_1 \\ s_2 = a_{3m} - a_3 + a_2 \\ s_3 = a_3 \end{cases}$$

$$R_1 = z - 1$$

Bước 6: Tính R và T

$$R = R_1B^+$$

$$T = A_0 B'_m$$

$$S = (a_{1m} - a_1 + 1)z^3 + (a_{2m} - a_2 + a_1)z^2 + (a_{3m} - a_3 + a_2)z + a_3$$

$$R = B^+R_1 = (z-1)(b_1z^2 + b_2z + b_3)$$

$$T = A_0B'_m = z(b_1mz^2 + b_2mz + b_3m)$$

Luật điều khiển

$$Ru = Tu_c - Sy$$

$$[b_1z^3 + (b_2 - b_1)z^2 + (b_3 - b_2)z - b_3]u \\ = z(b_1mz^2 + b_2mz + b_3m)u_c - [(a_{1m} - a_1 + 1)z^3 + (a_{2m} - a_2 + a_1)z^2 \\ + (a_{3m} - a_3 + a_2)z + a_3]y$$

$$[b_1 + (b_2 - b_1)z^{-1} + (b_3 - b_2)z^{-2} - b_3z^{-3}]u \\ = (b_1m + b_2mz^{-1} + b_3mz^{-2})u_c - [(a_{1m} - a_1 + 1) + (a_{2m} - a_2 + a_1)z^{-1} \\ + (a_{3m} - a_3 + a_2)z^{-2} + a_3z^{-3}]y$$

$$\begin{aligned}
& u \\
&= \left(\frac{b_1 m + b_2 m}{b_1} z^{-1} + \frac{b_3 m}{b_1} z^{-2} \right) u_c \\
&- \left[\frac{(a_{1m} - a_1 + 1) + (a_{2m} - a_2 + a_1) z^{-1} + (a_{3m} - a_3 + a_2) z^{-2} + a_3 z^{-3}}{b_1} \right] y \\
&- \left[\frac{(b_2 - b_1) z^{-1} + (b_3 - b_2) z^{-2} - b_3 z^{-3}}{b_1} \right] u
\end{aligned}$$

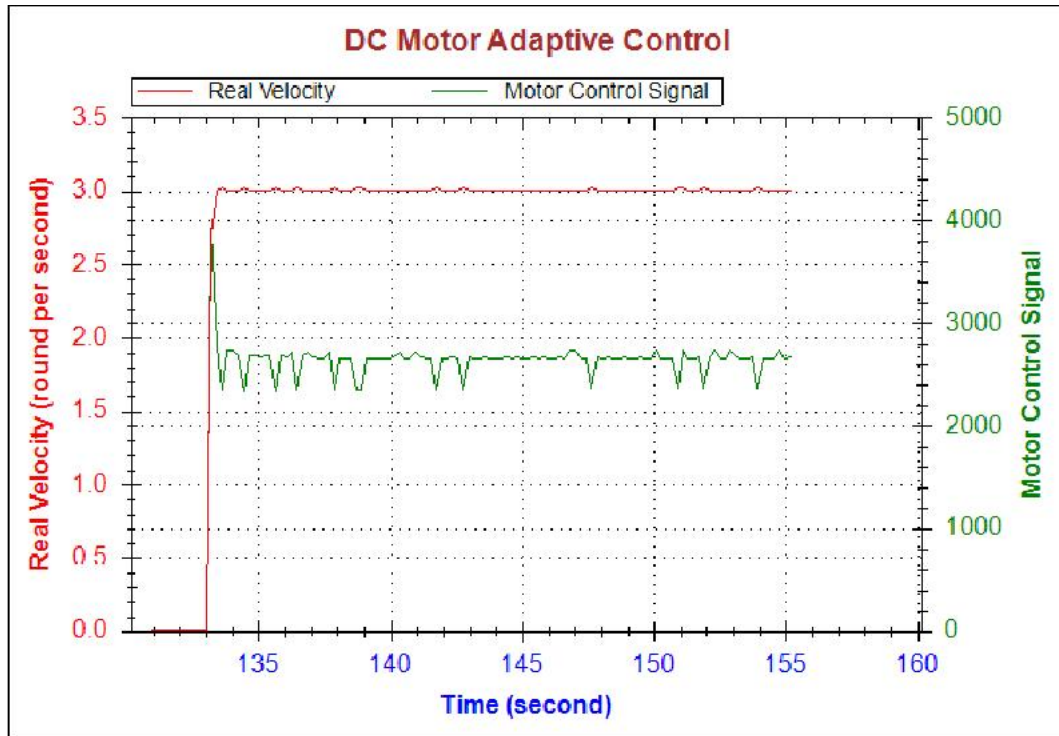
5.3.2 VÀI PHÉP THỬ VỚI NHIỀU DẠNG MÔ HÌNH CHUẨN KHÁC NHAU

Trong quá trình nghiên cứu và nghiên ngẫm những vấn đề của luận văn, tác giả nhận thấy rằng việc chọn mô hình chuẩn cũng ảnh hưởng không nhỏ đến chất lượng bộ điều khiển.

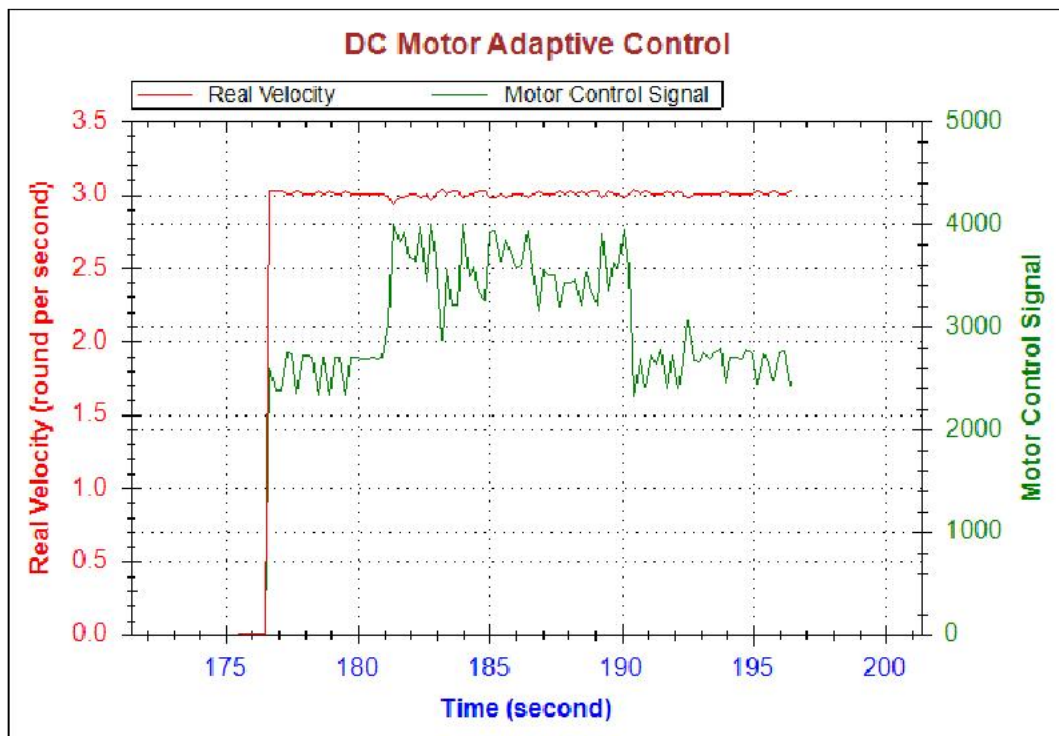
Khi chúng ta chọn mô hình mẫu quá cực đoan với thời gian xác lập rất bé, output của bộ điều khiển sẽ bị bão hòa trầm trọng hơn nhưng bù lại sẽ thích nghi nhanh hơn với các thay đổi.

Ngược lại, khi chọn một mô hình mẫu với thời gian xác lập dài, đặc biệt với trường hợp điều khiển vị trí, chúng ta cần một khoản thời gian không nhỏ để động cơ quay được một số vòng nhất định, mô hình xác lập dài khiến động cơ đạt được xác lập một cách mượt mà nhưng bù lại, khi đạt gần xác lập và trong quá trình xác lập lại thể hiện yếu kém khi giữ trạng thái xác lập một cách khó khăn.

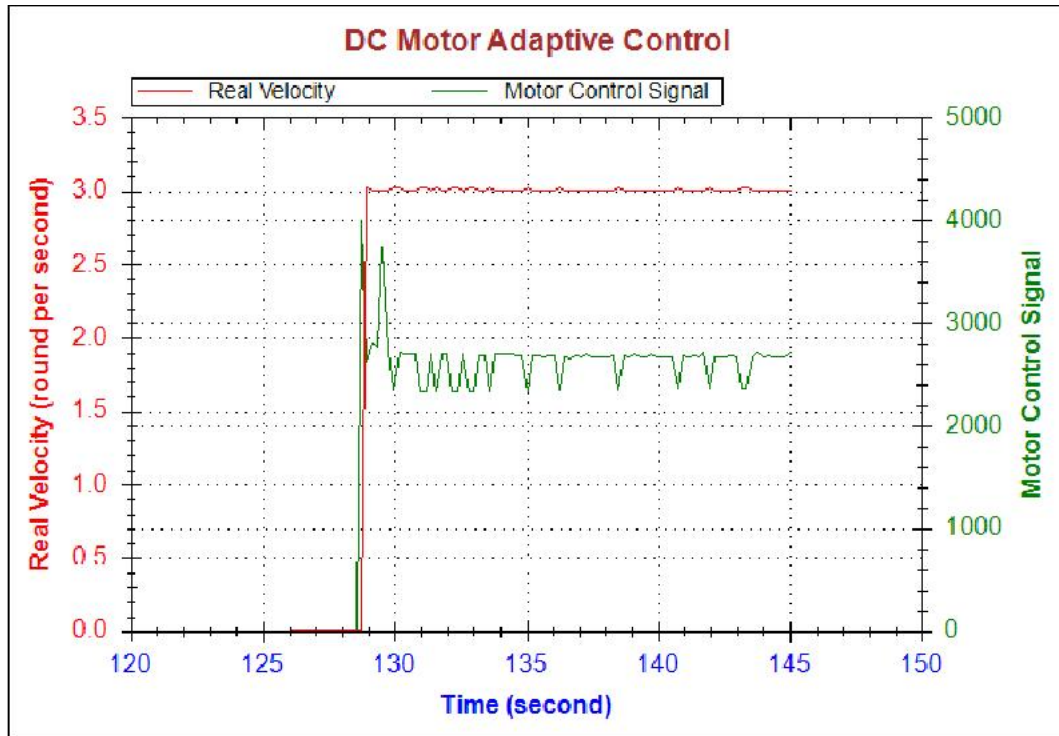
Sau đây, tác giả sẽ đưa ra vài kết quả thực nghiệm thể hiện sự khác nhau của bộ điều khiển khi chọn những mô hình mẫu khác nhau:



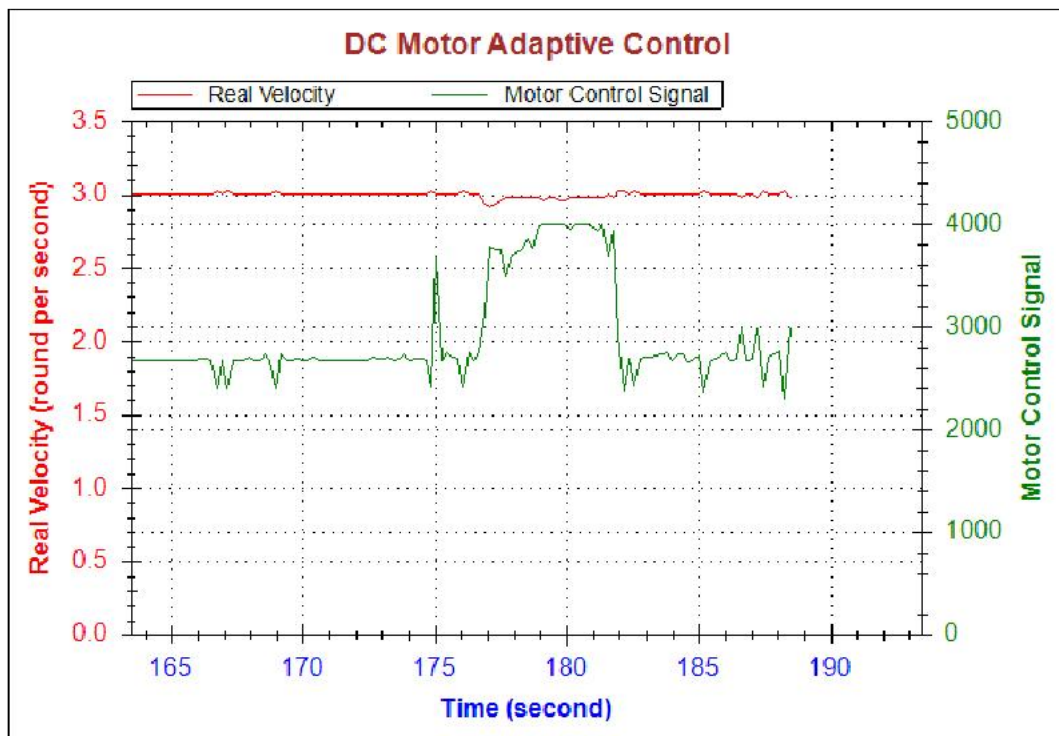
Hình 26: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 0.2 thời gian lấy mẫu



Hình 27: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 0.2 thời gian lấy mẫu khi có tải thay đổi



Hình 28: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 2 thời gian lấy mẫu



Hình 29: Đáp ứng và tín hiệu điều khiển của mô hình với thời gian xác lập bằng 2 thời gian lấy mẫu khi có tải thay đổi

Nhân xét:

Đối với những mô hình mẫu có thời gian xác lập bé, tín hiệu điều khiển động cơ thay đổi nhanh hơn để thích nghi nhiều hơn với điều kiện của mô hình mẫu, điều này làm chất lượng thích nghi cao hơn với một bất lợi là sự ổn định của hệ thống, vì thế cần chọn mô hình mẫu có thời gian xác lập phù hợp.

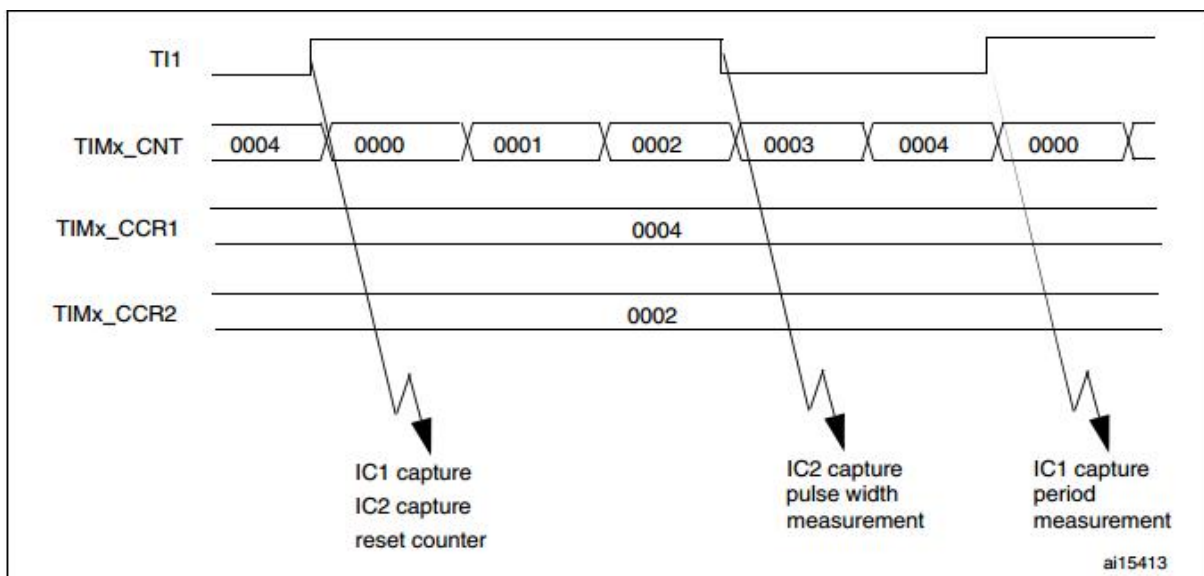
5.4 CẢI TIẾN KIẾN TRÚC LẬP TRÌNH

5.4.1 SỬ DỤNG NGẮT TRONG TRUYỀN VÀ NHẬN USART

Code lập trình của tác giả được lập trình tối ưu cho việc truyền nhận hoàn toàn bằng ngắt. Điều này không làm vướng bận bộ điều khiển khi đang tính toán.

Phương pháp truyền nhận polling hoàn toàn không phù hợp cho ứng dụng này do bộ vi xử lý phải làm việc rất nhiều với các ma trận ước lượng.

5.4.2 SỬ DỤNG PHƯƠNG PHÁP ĐO VẬN TỐC TỐI ƯU



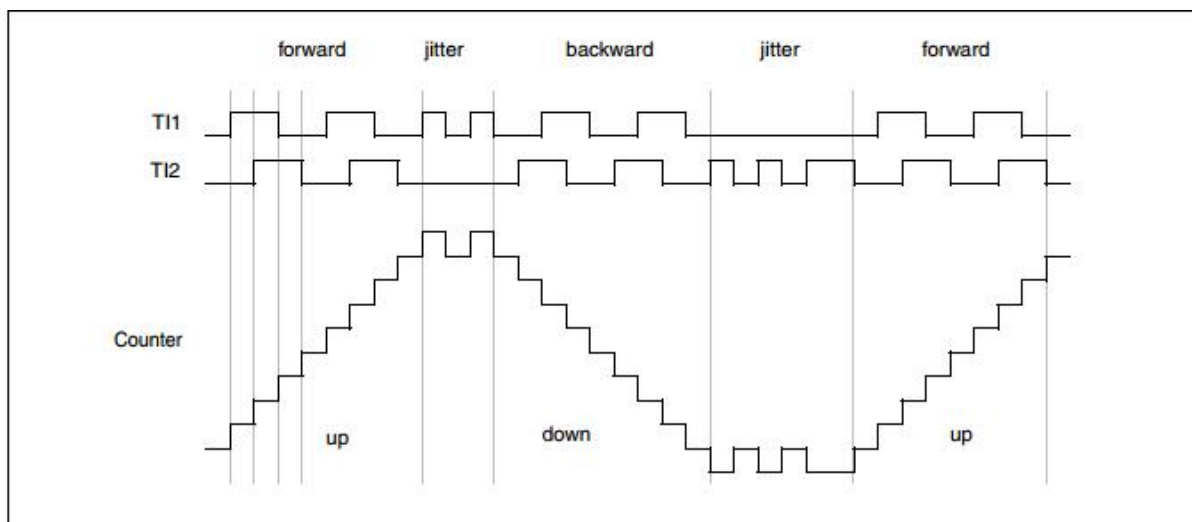
Hình 30: Phương pháp đo vận tốc bằng Input Capture

Có 2 cách đọc vận tốc mà chúng ta thường dùng là đếm số xung encoder trong 1 khoảng thời gian nhất định và đo thời gian trôi qua giữa hai xung. Cách thứ hai được tác giả đánh giá là mang nhiều ưu điểm hơn, vì độ chính xác cao, giải thuật không cần ngắt, không phải lãng phí một khoảng thời gian trong chu kỳ lấy mẫu để đọc xung và đọc được cả chiều của động cơ trong thời điểm đo.

Tuy nhiên cả 2 cách trên đều có một nhược điểm đó chính là vận tốc tối thiểu, luôn tồn tại một vận tốc tối thiểu để có thể đọc encoder một cách chính xác mà không làm ảnh hưởng đến

kết quả của bộ điều khiển, chúng ta chỉ có thể đưa ra hai hướng giải quyết: Một là tăng thời gian lấy mẫu và điều khiển lên, hai là sử dụng bộ encoder có số xung lớn hoặc hộp giảm tốc.

5.4.3 SỬ DỤNG BỘ ENCODER ĐẾM BẰNG PHẦN CỨNG x4 SỐ XUNG



Hình 31: Phương pháp đo vị trí bằng Quadrature Encoder Interface

Bộ đọc encoder của chip ARM được tối ưu hóa bằng phần cứng, nghĩa là thanh ghi mang nhiệm vụ đếm xung encoder tự động tăng hoặc giảm ứng với chiều lên xuống của từng kênh encoder mà không cần phải vào ngắt từ đó tối ưu hiệu năng của hệ thống.

Đặc biệt trong trường hợp động cơ đảo chiều rất nhanh và liên tục, hoặc trong trường hợp có nhiều ở 1 kênh, bộ đọc encoder x4 xung sẽ không làm xuất hiện trường hợp đếm thừa/thiếu xung làm lệch đáp ứng trong điều khiển vị trí.

5.4.4 SỬ DỤNG HIỆU QUẢ NHỮNG LỆNH TIỀN XỬ LÝ KHI LẬP TRÌNH

Do có rất nhiều phương pháp tối ưu hóa khác nhau khi thực hiện giải thuật, tác giả khuyến nghị nên sử dụng những lệnh tiền xử lý để thay đổi và so sánh kết quả của những giải thuật khác nhau một cách dễ dàng và nhanh chóng hơn.

CHƯƠNG 6: CÁC KẾT QUẢ THỰC NGHIỆM

Chúng ta lần lượt làm một số thí nghiệm để so sánh hiệu năng của 4 bộ điều khiển sau đây:

1. Bộ điều khiển PID chọn đúng hệ số
2. Bộ điều khiển PID với hệ số chưa tốt
3. Bộ điều khiển tự chỉnh cơ bản
4. Bộ điều khiển tự chỉnh tối ưu hóa

Trong hai trường hợp là:

1. Điều khiển động cơ với tải không đổi
2. Điều khiển động cơ với tải thay đổi

Trước hết, tác giả xin được giới thiệu bộ PID với cách chỉnh hệ số chỉnh bằng tay:

- Phương trình liên tục và rời rạc của giải thuật PID:

$$G(s) = Kp + \frac{Ki}{s} + Kd * s$$

$$G(z) = \frac{U(z)}{E(z)} = Kp + \frac{Ki * T}{2} * \frac{z + 1}{z - 1} + \frac{Kd}{T} * \frac{z - 1}{z}$$

$$U(k) = Kp * [E(k) - E(k - 1)] + 0.5Ki * T * [E(k) + E(k - 1)] + \frac{Kd * [E(k) - 2E(k - 1) + E(k - 2)]}{T} + U(k - 1)$$

- Phương pháp chọn hệ số PID bằng tay:

Cài đặt hệ số Ki, Kd bằng zero, tăng từ từ Kp cho đến khi hệ dao động. Tăng từ từ Ki cho đến khi hệ thống tốt hơn, tương tự sau đó đến hệ số Kd. Việc điều chỉnh các thông số phụ thuộc nhiều vào kinh nghiệm và có thể được tóm tắt qua bảng sau đây.

Bảng 6: Xác định các thông số PID bằng tay

Thông số	Thời gian lên	Vọt lố	Thời gian xác lập	Sai số xác lập	Tính ổn định
K_p	Giảm	Tăng	Thay đổi nhỏ	Giảm	Giảm
K_i	Giảm	Tăng	Tăng	Loại bỏ	Giảm
K_d	Thay đổi nhỏ	Giảm	Giảm	Không ảnh hưởng	Tăng nếu Kd nhỏ

Sau đó, tác giả xin được giới thiệu bộ PID với cách chỉnh hệ số chỉnh bằng Ziegler–Nichols:

Cài đặt hệ số K_i , K_d bằng zero, tăng từ từ K_p cho đến khi hệ dao động. Chọn hệ số K_p bé nhất gây ra dao động là K_u , tần số dao động là P_u . Các hệ số PID được tính như sau:

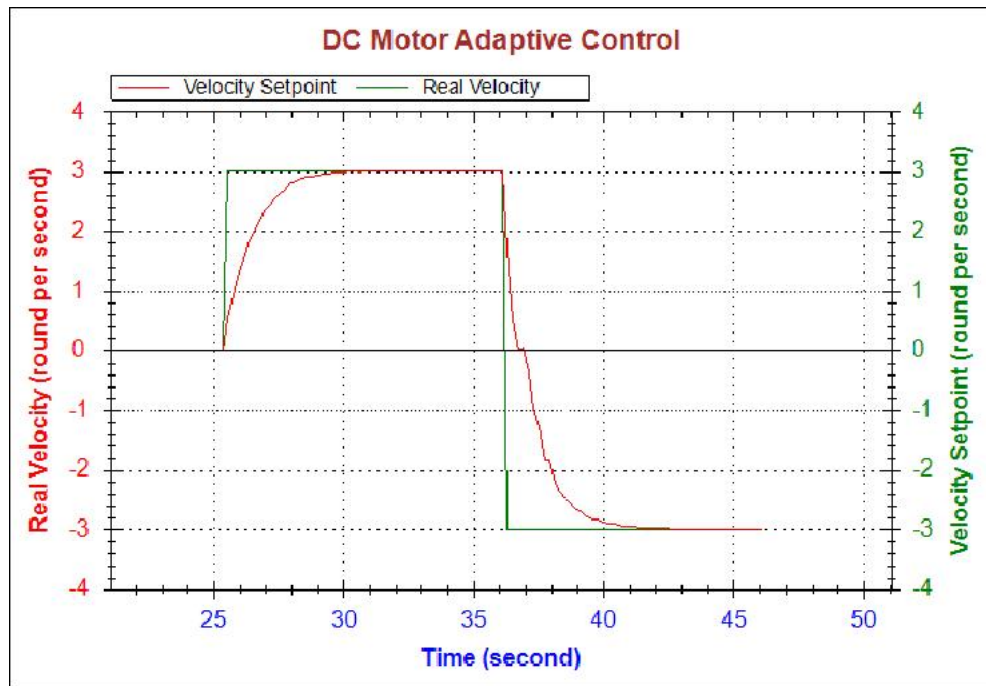
Bảng 7: Xác định hệ số PID bằng Ziegler–Nichols

Ziegler–Nichols method			
Control Type	K_p	K_i	K_d
P	$0.50K_u$	-	-
PI	$0.45K_u$	$1.2K_p/P_u$	-
PID	$0.60K_u$	$2K_p/P_u$	$K_pP_u/8$

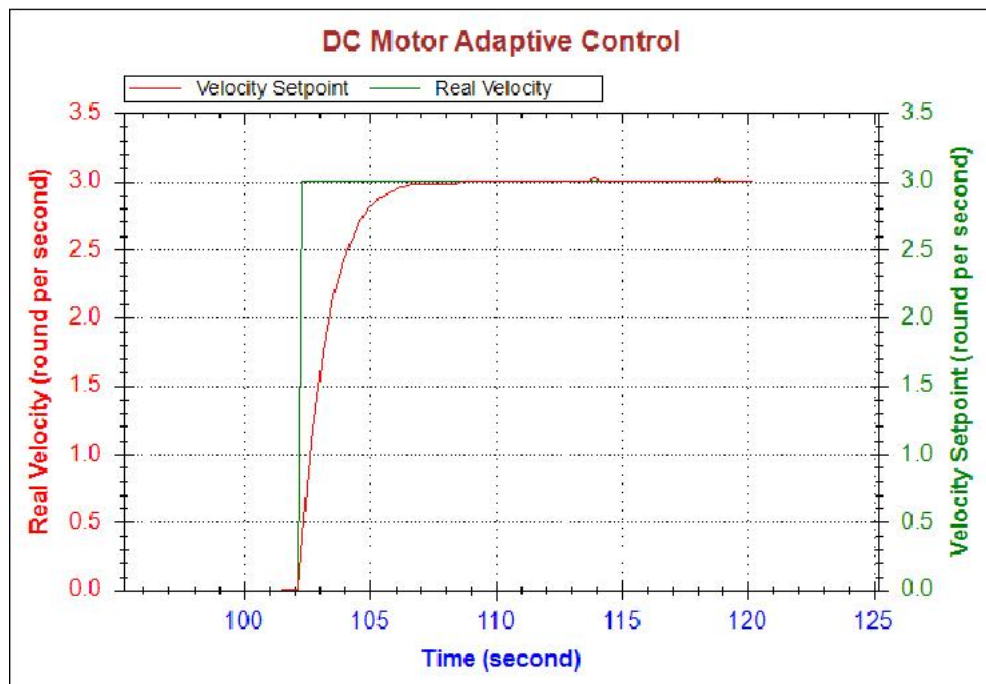
6.1 ĐIỀU KHIỂN VẬN TỐC ĐỘNG CƠ

6.1.1 ĐÁNH GIÁ CHẤT LƯỢNG ĐÁP ỨNG VÀ TÍNH THÍCH NGHI CỦA CÁC BỘ ĐIỀU KHIỂN TRÊN CÙNG MỘT MÔ HÌNH

BỘ ĐIỀU KHIỂN PID VỚI HỆ SỐ ZIEGLER–NICHOLS



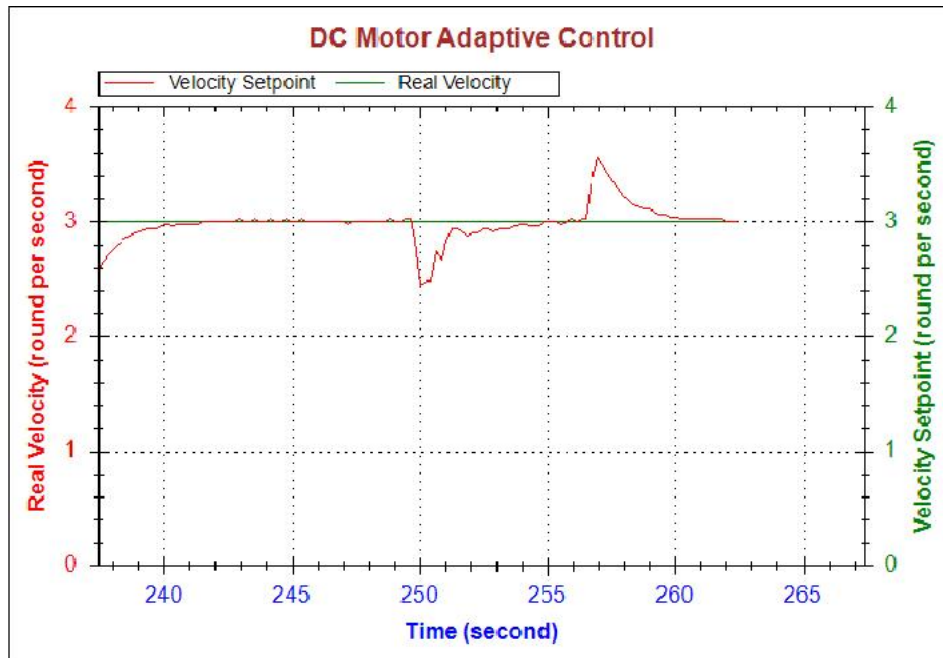
Hình 32: Đáp ứng của bộ PID với hệ số chưa tốt khi chưa có sự thay đổi của tải (toàn cục)



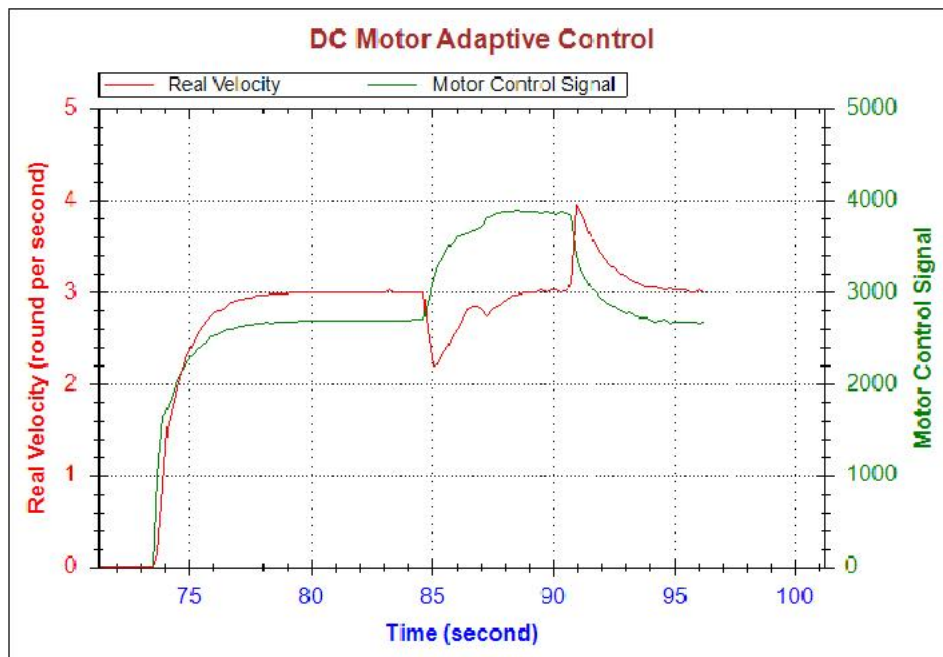
Hình 33: Đáp ứng của bộ PID với hệ số chưa tốt khi chưa có sự thay đổi của tải (phóng to)

Nhận xét:

Bộ điều khiển PID hoạt động với thời gian xác lập dài nếu cần sai số xác lập bé $\sim 1\%$ và ngược lại nếu chúng ta điều chỉnh hệ số PID để xác lập nhanh thì phải đánh đổi bằng vọt lố hoặc sai số xác lập.



Hình 34: Đáp ứng của bộ PID với hệ số chưa tốt với tải thay đổi

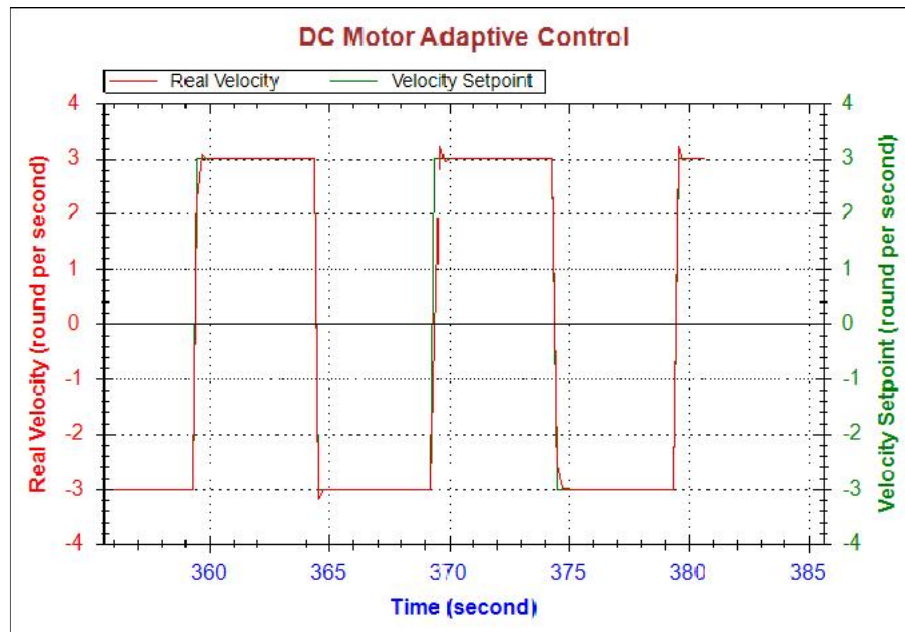


Hình 35: Tín hiệu điều khiển của bộ PID với hệ số chưa tốt với tải thay đổi

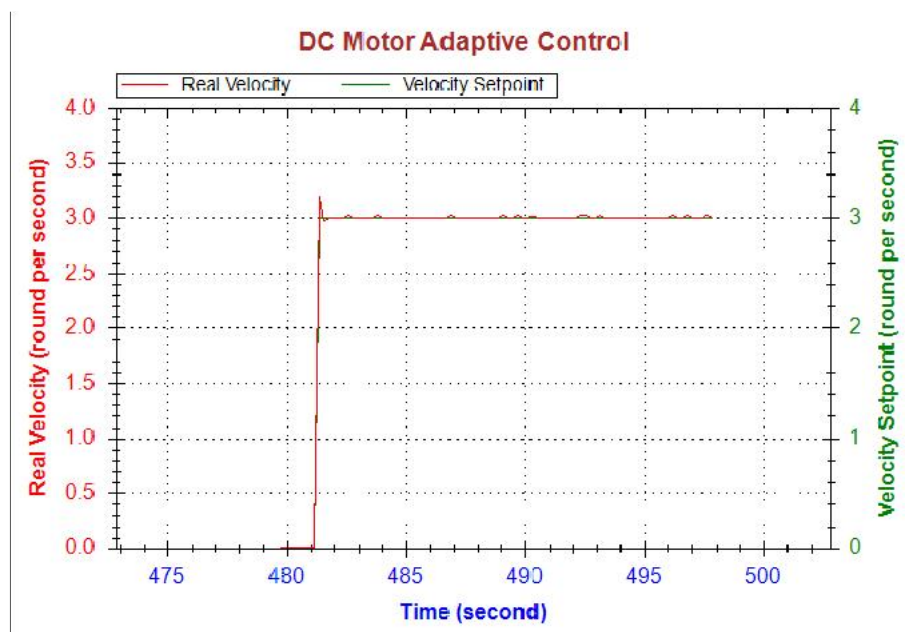
Nhận xét:

Bộ điều khiển PID thể hiện sự lẹp vế khi mang tính thích nghi thấp, khi thay đổi thông số của tải hai lần thì ngay lập tức xuất hiện những vọt lố, tín hiệu điều khiển đáp ứng rất chậm vì luật điều khiển đơn giản.

BỘ ĐIỀU KHIỂN PID VỚI HỆ SỐ CHỌN PHÙ HỢP



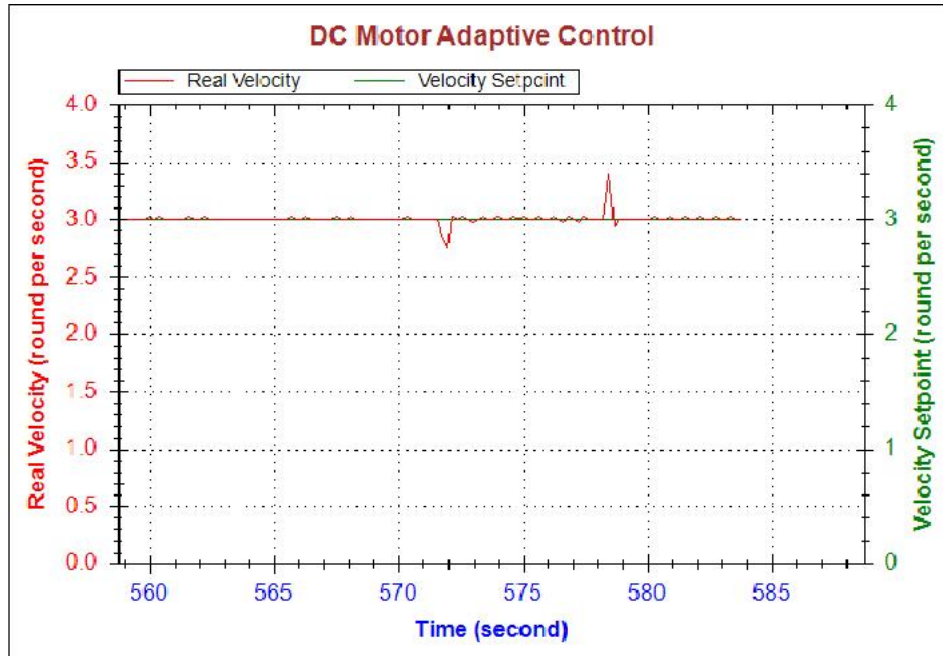
Hình 36: Đáp ứng của bộ PID chọn đúng hệ số khi chưa có sự thay đổi của tải (toàn cục)



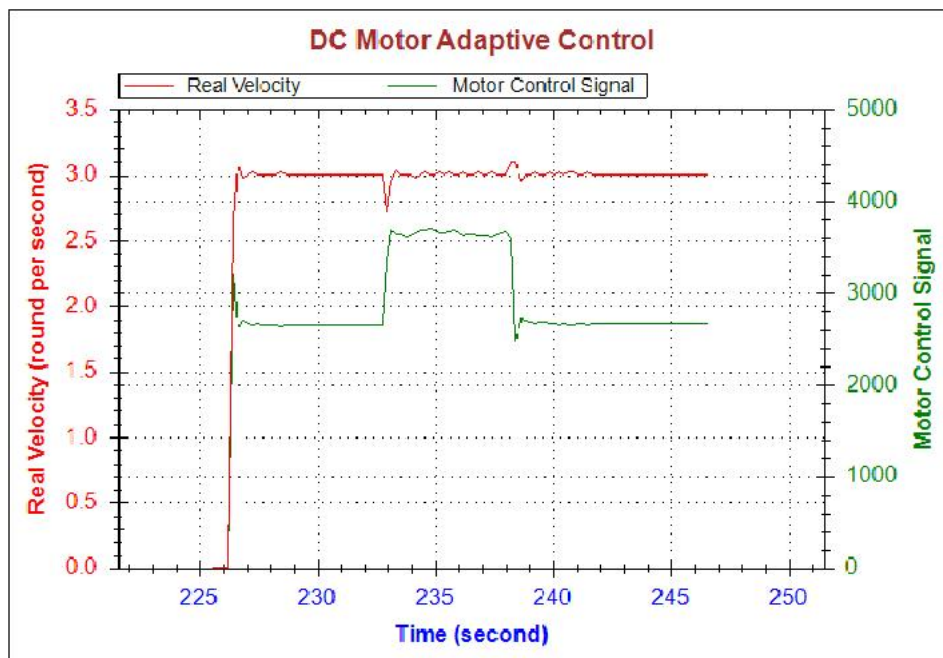
Hình 37: Đáp ứng của bộ PID chọn đúng hệ số khi chưa có sự thay đổi của tải (phóng to)

Nhận xét:

Bộ điều khiển PID chọn đúng hệ số do tác giả có kinh nghiệm và may mắn nên chọn được hệ số PID tốt hơn với chất lượng điều khiển tương đương bộ STR với sai số xác lập $\sim 1\%$ và thời gian xác lập nhanh



Hình 38: Đáp ứng của bộ PID chọn đúng hệ số với tải thay đổi

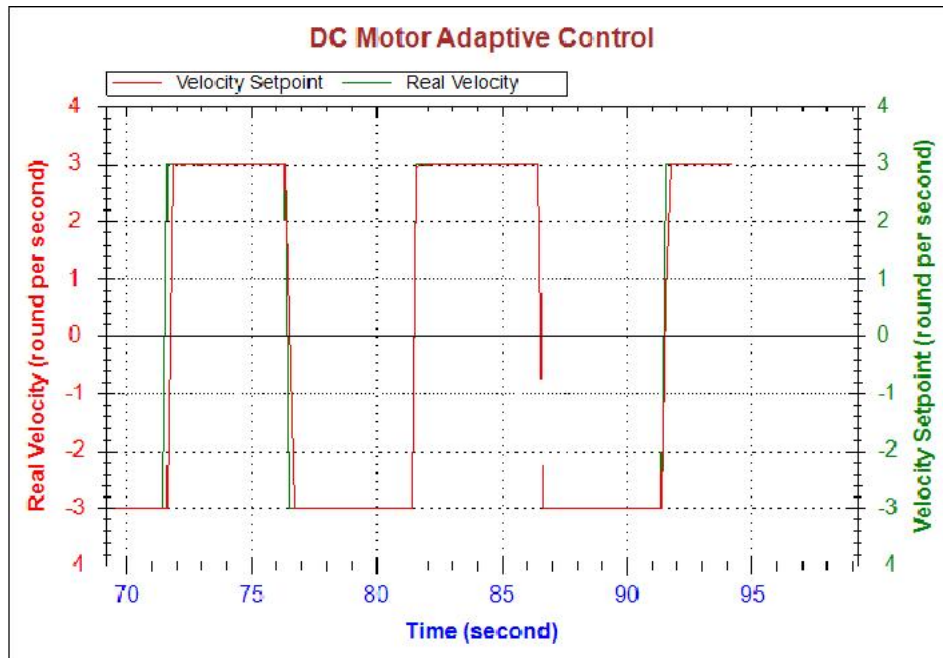


Hình 39: Tín hiệu điều khiển của bộ PID chọn đúng hệ số với tải thay đổi

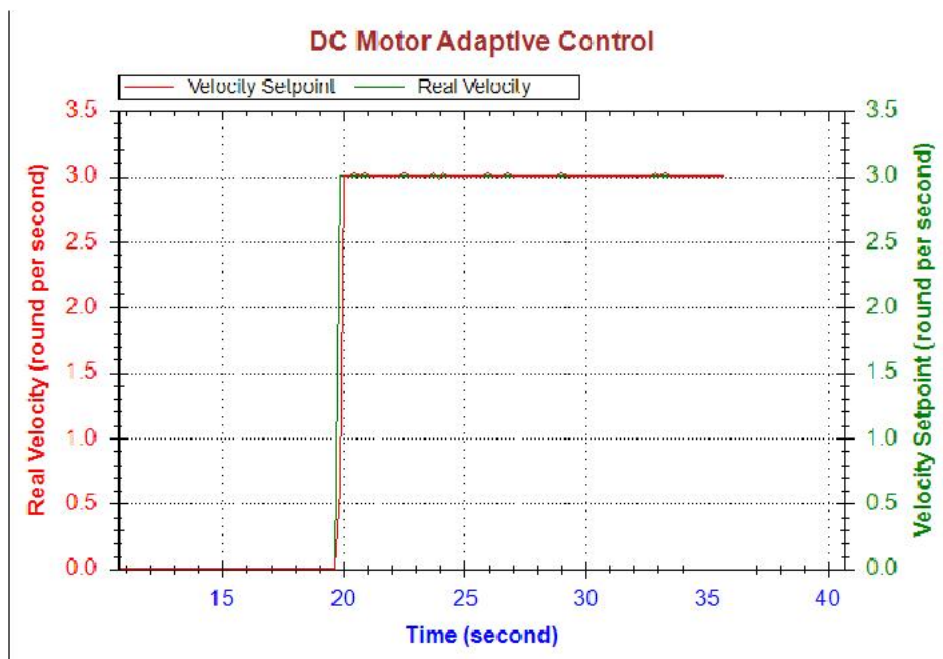
Nhận xét:

Bộ điều khiển PID chọn đúng hệ số do tác giả có kinh nghiệm và may mắn nên chọn được hệ số PID tốt hơn tuy nhiên tính thích nghi với hệ số PID chọn đúng vẫn chưa thể bằng bộ STR.

BỘ ĐIỀU KHIỂN TỰ CHỈNH



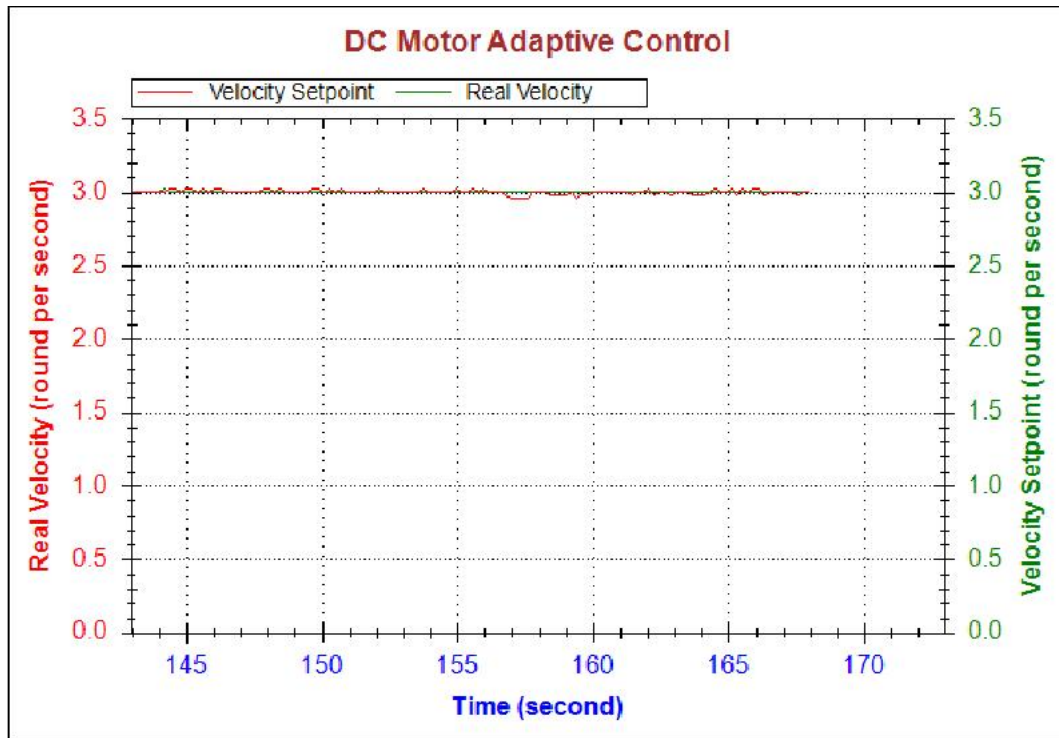
Hình 40: Đáp ứng của bộ STR khi chưa có sự thay đổi của tải



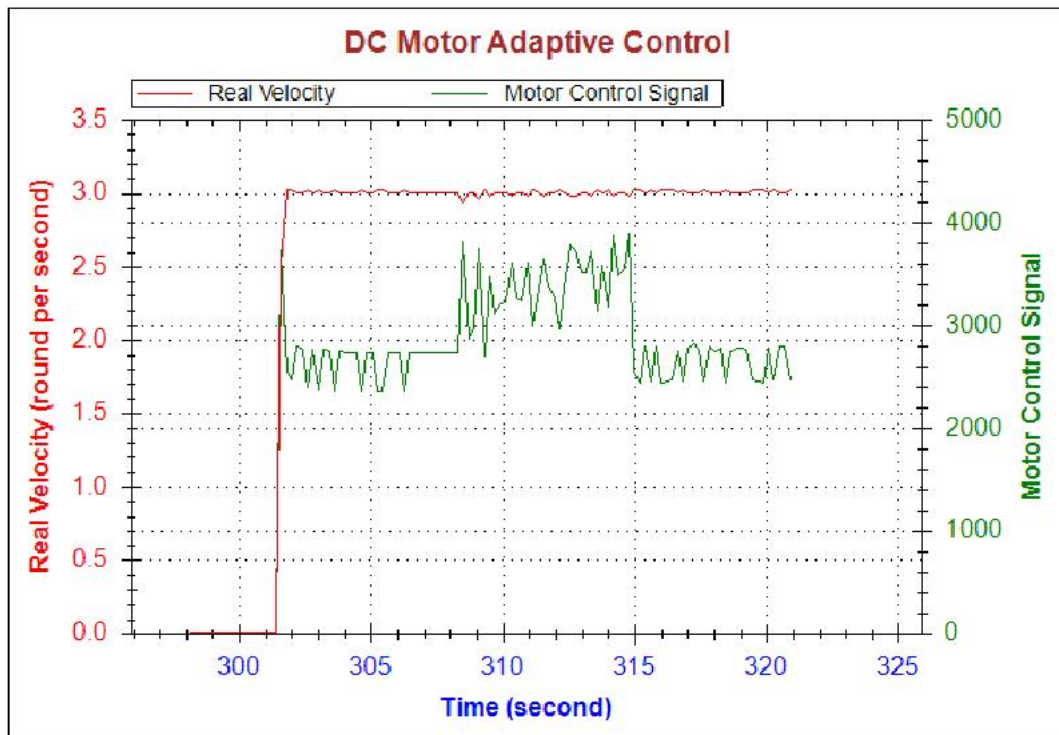
Hình 41: Đáp ứng của bộ STR khi chưa có sự thay đổi của tải (phóng to)

Nhận xét:

Bộ điều khiển STR hoạt động tốt khi cho thời gian xác lập ngắn nhất và sai số xác lập $\sim 1\%$



Hình 42: Đáp ứng của bộ STR với tải thay đổi



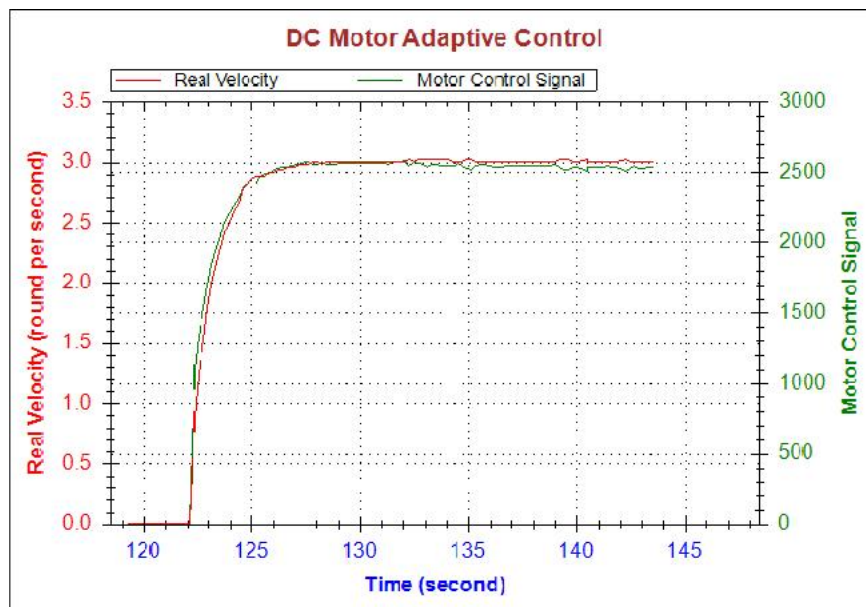
Hình 43: Tín hiệu điều khiển của bộ STR với tải thay đổi

Nhận xét:

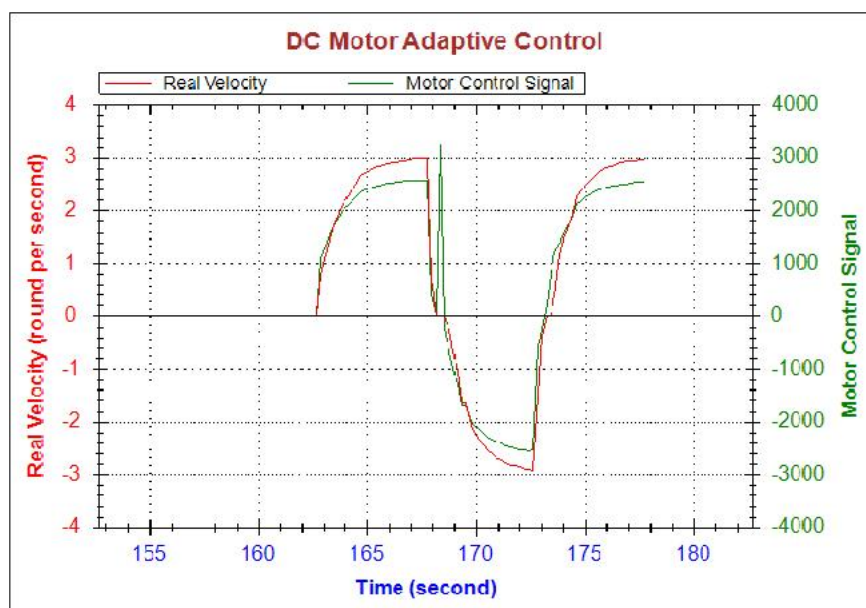
Bộ điều khiển mang tính thích nghi rất tốt với khi ngay lập tức điều chỉnh thông số phù hợp với sự thay đổi của tải

6.1.2 ĐÁNH GIÁ CHẤT LƯỢNG CỦA TÍN HIỆU ĐIỀU KHIỂN

BỘ ĐIỀU KHIỂN PID VỚI HỆ SỐ CHƯA TỐT

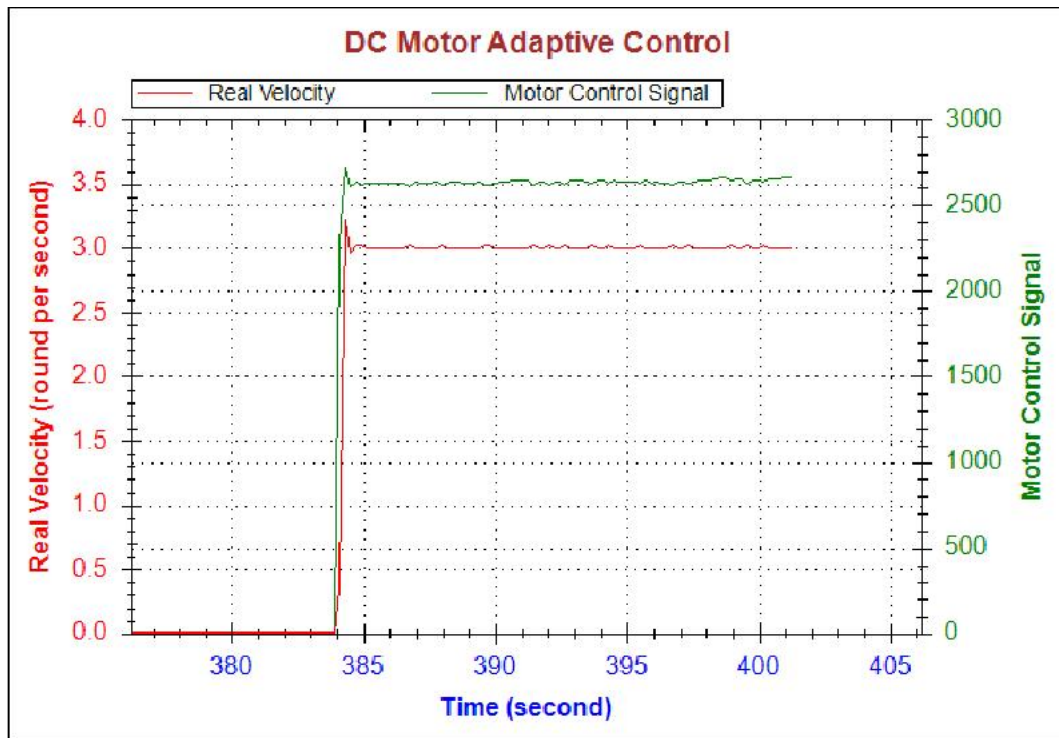


Hình 44: Tín hiệu điều khiển của bộ PID với hệ số chưa tốt (phóng to)

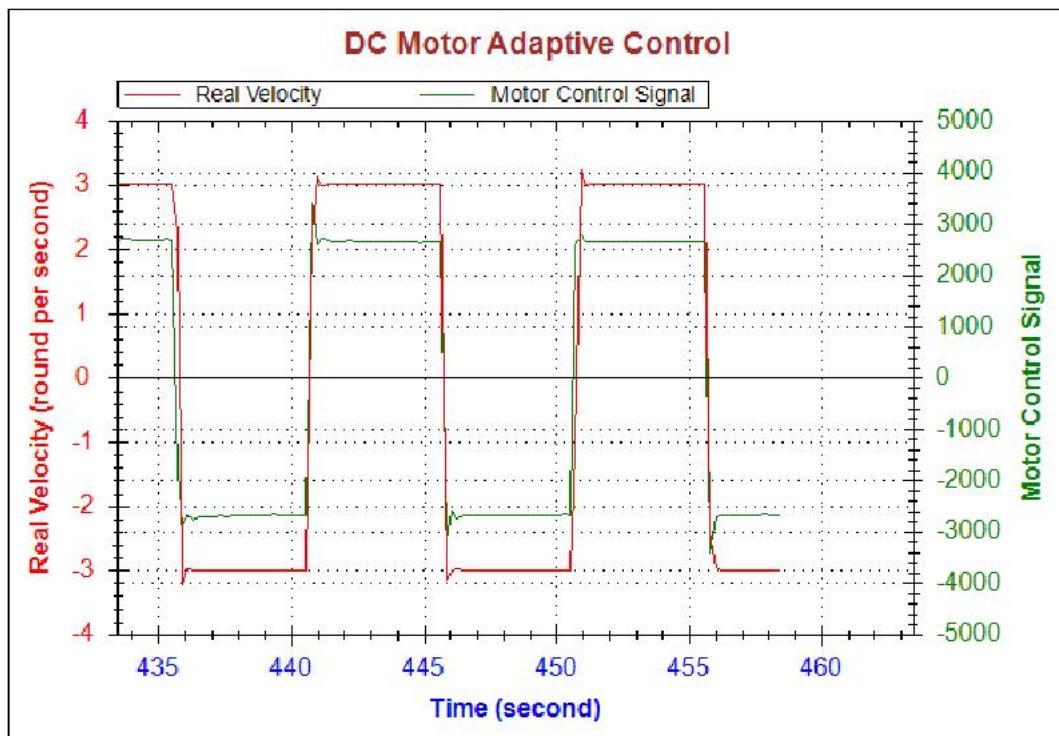


Hình 45: Tín hiệu điều khiển của bộ PID với hệ số chưa tốt (toàn cục)

BỘ ĐIỀU KHIỂN PID VỚI HỆ SỐ CHỈNH BẰNG TAY



Hình 46: Tín hiệu điều khiển của bộ PID chọn đúng hệ số (phóng to)

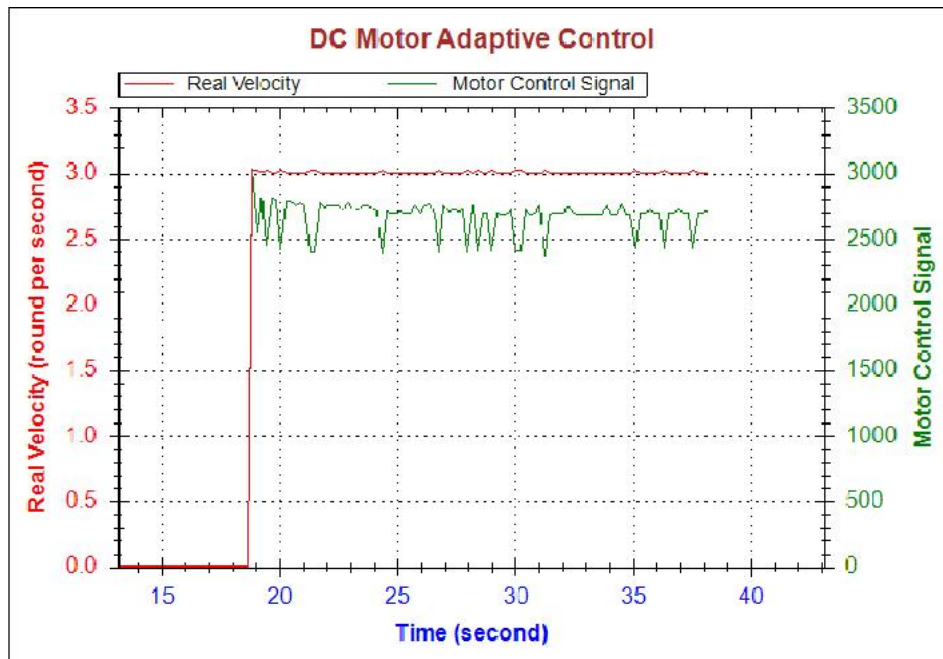


Hình 47: Tín hiệu điều khiển của bộ PID chọn đúng hệ số (toàn cục)

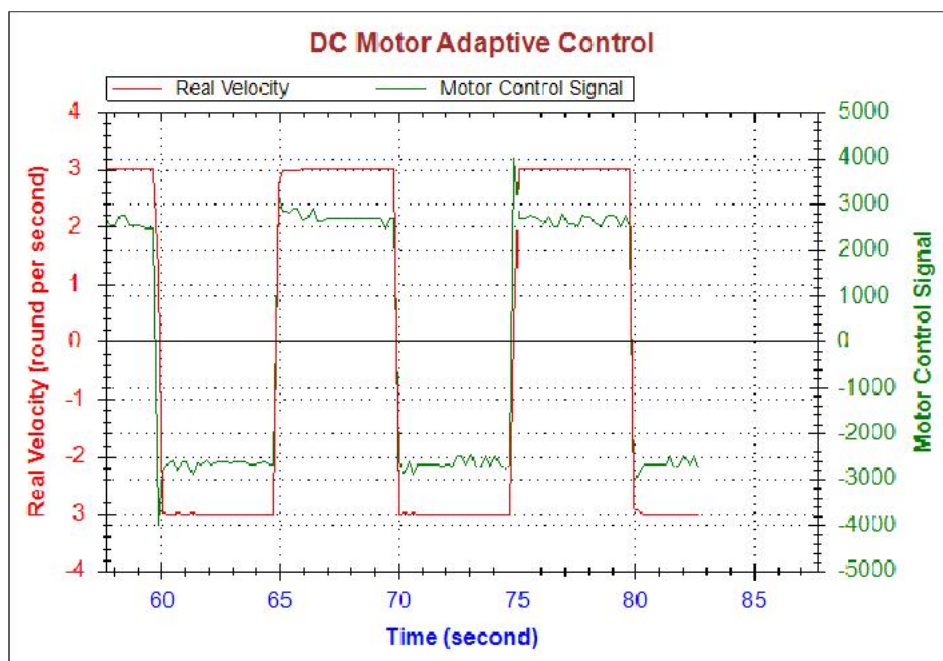
Nhận xét:

Tín hiệu điều khiển của bộ PID chọn đúng hệ số có độ dao động ít hơn do đó tính thích nghi thấp với sự thay đổi của môi trường

BỘ ĐIỀU KHIỂN TỰ CHỈNH



Hình 48: Tín hiệu điều khiển của bộ STR (phóng to)



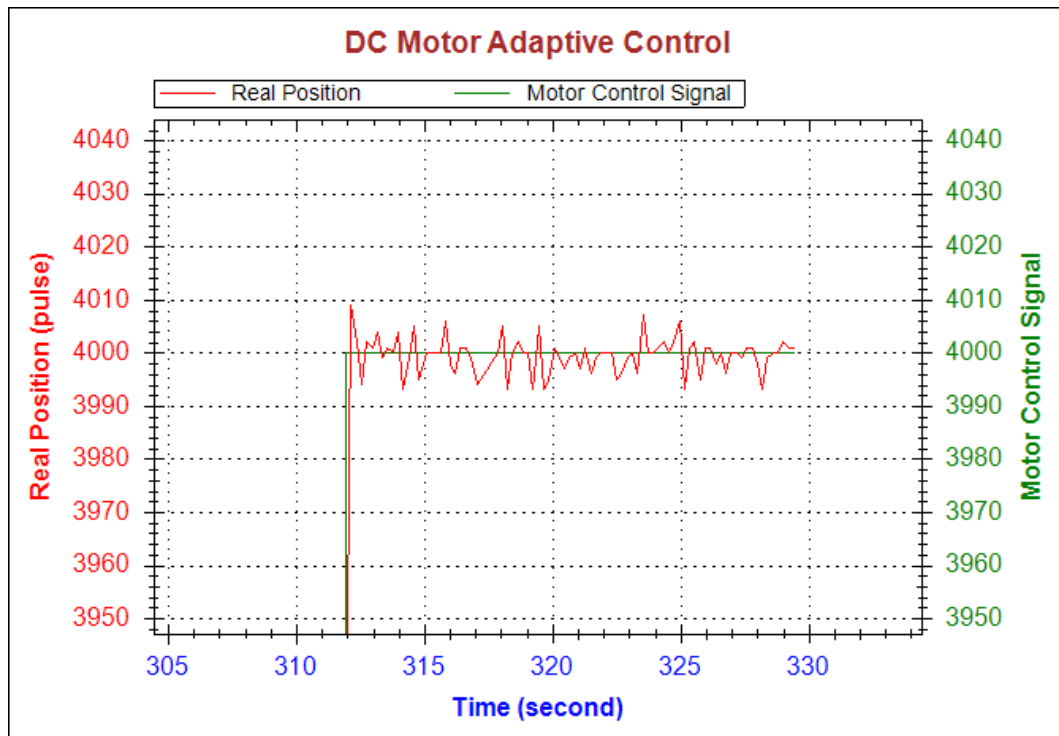
Hình 49: Tín hiệu điều khiển của bộ STR (toàn cục)

Nhận xét:

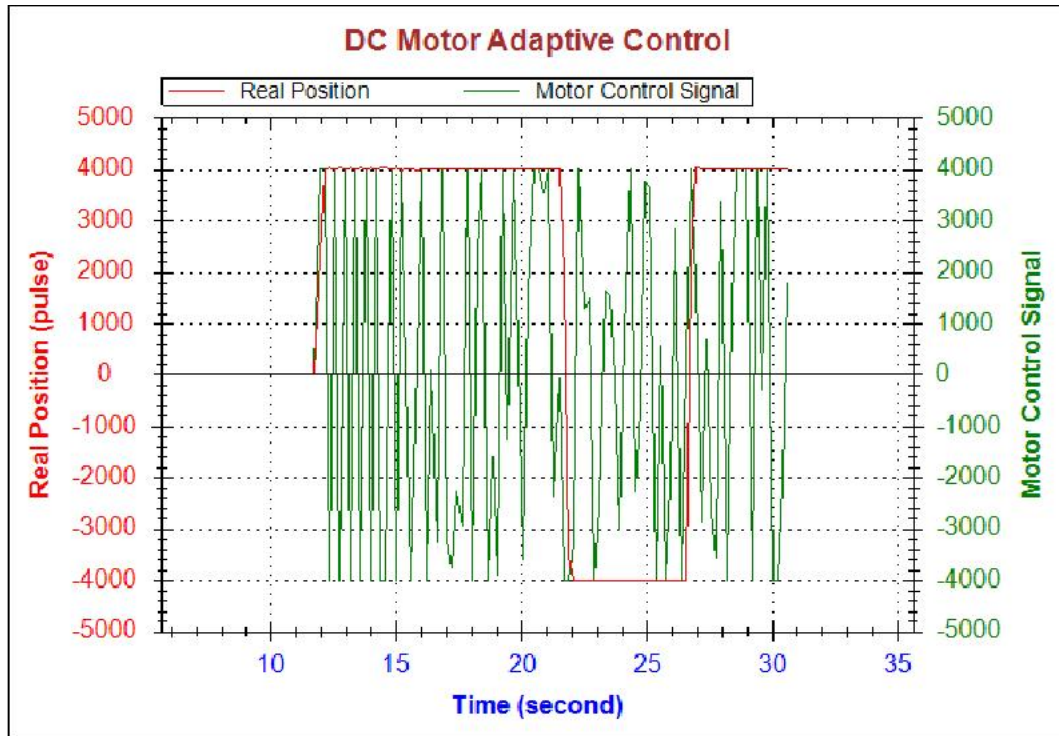
Tín hiệu điều khiển của bộ STR có độ nhạy cao, nhờ đó thích ứng nhanh với những thay đổi từ môi trường.

6.2 ĐIỀU KHIỂN VỊ TRÍ ĐỘNG CƠ

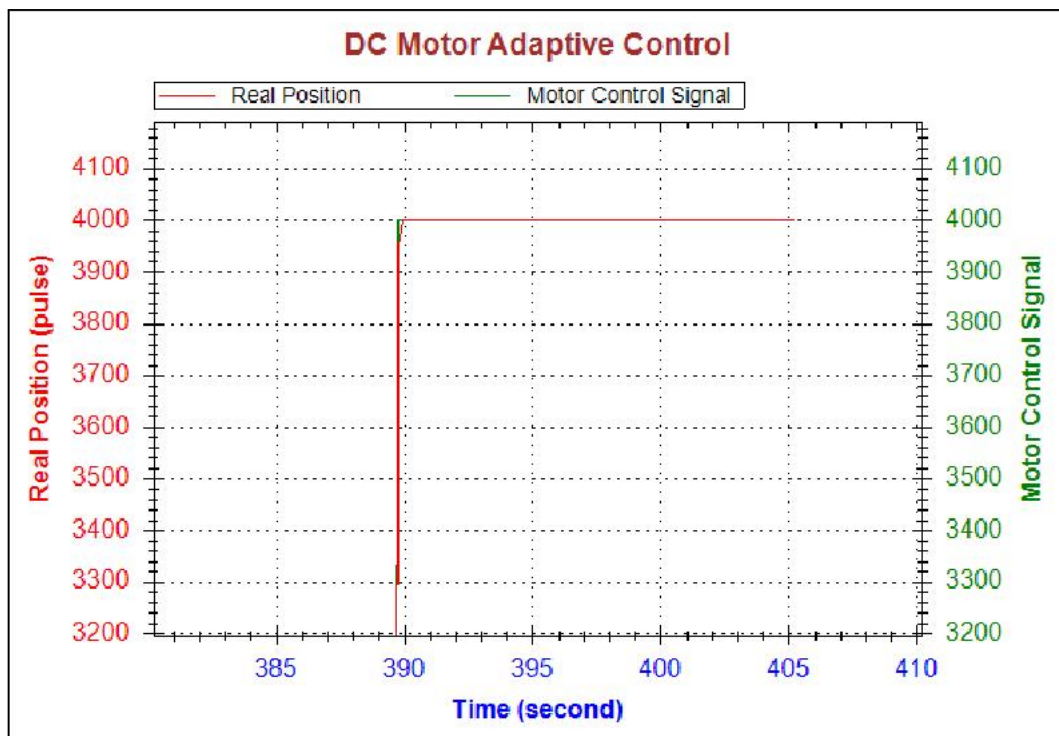
6.2.1 ĐÁNH GIÁ CHẤT LƯỢNG CỦA NHỮNG TỐI ƯU HÓA TRONG GIẢI THUẬT STR



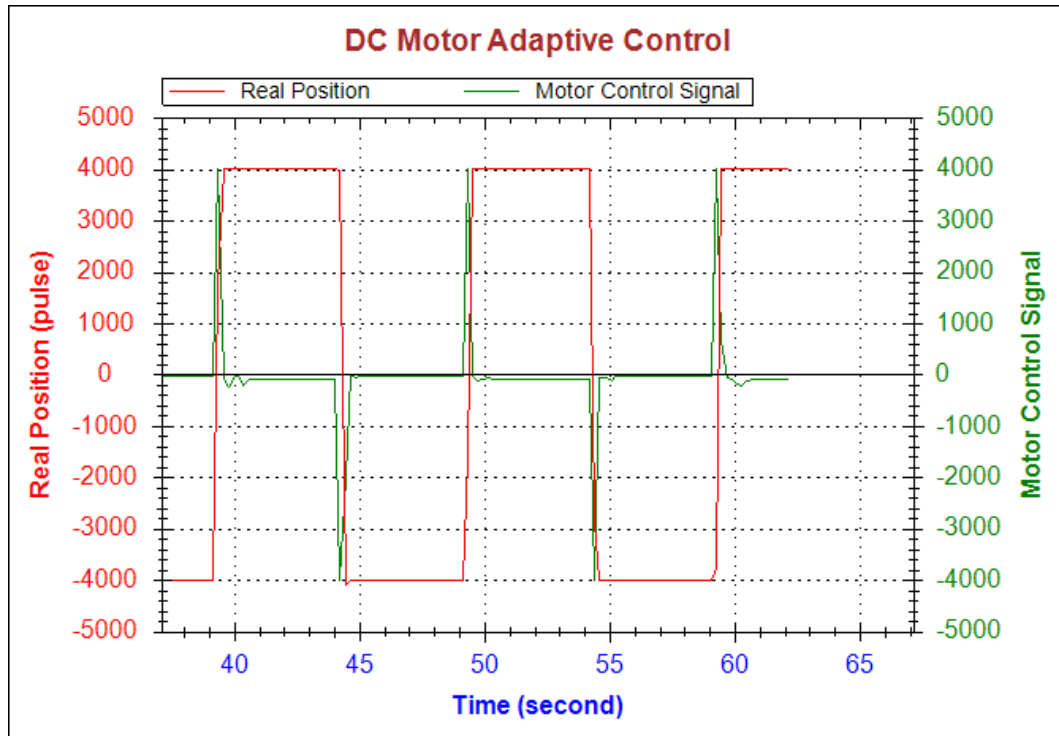
Hình 50: Điều khiển vị trí động cơ với giải thuật STR cơ bản, xảy ra hiện tượng rung lắc động cơ khi đạt xác lập



Hình 51: Tín hiệu điều khiển vị trí động cơ của bộ PID bị bão hòa và dao động mạnh



Hình 52: Điều khiển vị trí động cơ với giải thuật STR cải tiến, hệ số trễ bằng 1, loại bỏ hoàn toàn hiện tượng rung lắc



Hình 53: Tín hiệu điều khiển vị trí động cơ của bộ STR

CHƯƠNG 7: ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

7.1 VỀ PHẦN GIẢI THUẬT ƯỚC LƯỢNG

1. Có một giải thuật thích nghi mới hơn mà tác giả vô tình tìm ra trong lúc tìm lại liệu cho luận văn hiện thời, giải thuật này áp dụng cho trường hợp cho hệ thống có nhiều đầu vào và nhiều đầu ra. Nếu chúng ta áp dụng vừa điều khiển vị trí, vừa điều khiển vận tốc động cơ, chúng ta có thể tạo ra một bộ điều khiển vận năng tương tự như các nhà khoa học trên thế giới.
2. Như ta đã biết, hệ số quên giúp cho bộ điều khiển thích nghi với sự thay đổi của 6 thông số động cơ trong phương trình của hàm truyền rời rạc, tuy nhiên 6 thông số này có độ thay đổi dao động khác nhau với mức độ khác nhau, các tác giả khác hiện nay đang phát triển một giải thuật áp dụng một ma trận quên ứng với từng thông số khác nhau của động cơ thay vì chỉ duy nhất một hệ số. Điều này chắc chắn mang lại một bước tiến không nhỏ cho bộ điều khiển động cơ hiện thời.
3. Ma trận covariance P là “trái tim” của bộ ước lượng, những bất ổn định dù nhỏ nhất trong P cũng gây ra những thay đổi lớn trong bộ điều khiển. Một vài phương pháp tối ưu hóa đặc tính toán học của ma trận P gồm có reset ma trận P, tối ưu hóa tổng đường chéo ma trận, tối ưu hóa độ hội tụ và độ vọt lố.

7.2 VỀ PHẦN BỘ ĐIỀU KHIỂN THEO MÔ HÌNH CHUẨN

1. Đa thức A_0 trong bộ điều khiển theo mô hình chuẩn được đưa ra theo một công thức khác theo một số tài liệu.
2. Tác giả đề nghị thay thế thử bộ điều khiển theo mô hình chuẩn bằng bộ điều khiển tối thiểu năng lượng, cách điều khiển này tối thiểu sự dao động của tín hiệu điều khiển, vừa tránh vọt lố, vừa tiết kiệm năng lượng.

7.3 VỀ KHỐI MÔ HÌNH CHUẨN

1. Đối với trường hợp điều khiển vị trí, khi tải thay đổi quá lớn vẫn tồn tại vọt lố khi mô hình mẫu có thời gian xác lập nhỏ, khi sử dụng mô hình mẫu có thời gian xác lập lớn, động cơ hoàn toàn không có vọt lố như không xác lập được.

Tác giả khuyến nghị chúng ta thay đổi mô hình chuẩn một cách phù hợp hoặc tính toán mô hình chuẩn khác nhau ngay trong quá trình điều khiển động cơ để tận dụng toàn bộ ưu điểm của các mô hình chuẩn khác nhau. (Xem thêm mục vài phép thử với nhiều dạng mô hình chuẩn khác nhau).

7.4 VỀ PHẦN CỨNG HOẶC CODE LẬP TRÌNH

1. Cần một giải thuật lọc nhiễu ở đầu vào và đầu ra của động cơ như bộ lọc Kalman để hiệu quả hơn.
2. Cần thử nghiệm với nhiều loại động cơ hơn nữa, trong quá trình làm luận văn, tác giả chỉ đã thử trên 4 loại động cơ khác nhau.
3. Truyền nhận UART trong luận văn của tác giả sử dụng ngắt để truyền, điều này mang lại hiệu năng cao hơn so với truyền nhận bằng hình thức polling, tác giả đã thử dùng phương pháp truyền nhận bằng DMA nhưng gặp khó khăn do độ trễ cao. Tuy nhiên, qua kinh nghiệm từ các thế hệ đi trước, việc truyền nhận UART DMA bằng kit STM32F4 là khả thi. Việc truyền nhận bằng DMA giúp hiệu năng của bộ vi xử lý tăng thêm 30% so với phương pháp hiện thời theo ý kiến chủ quan của tác giả.
4. Ngoài ra, tác giả khuyến nghị truyền nhận bằng chuẩn USB 2.0 có sẵn trên kit STM32F4-Discovery. Chuẩn này hoạt động giống như cổng Com nhưng giao thức có tốc độ nhanh và đáng tin cậy hơn hẳn.

7.5 KẾ HOẠCH PHÁT TRIỂN

Tác giả cung cấp những tài liệu tham khảo đi kèm luận văn, bản tóm tắt luận văn, bản in đầy đủ, mã nguồn và giải thích các công thức để người đọc tự tìm hiểu, nghiên cứu, khắc phục

những khuyết điểm mà luận văn này còn mắc phải. Để đặt câu hỏi và trao đổi kinh nghiệm, người đọc có thể truy cập forum www.payitforward.edu.vn

PHỤ LỤC TÀI LIỆU THAM KHẢO

- [1] Thai Hoang – Huynh, *Advance control engineering theory – Chapter 4: Adaptive controllers*, <http://www4.hcmut.edu.vn/~hthoang/ltdknc/index.htm> 2008.
- [2] Prof. Zoran Vukic, *A tutorial on adaptive control Self-tuning approach*, University of Zagreb, Croatia, Faculty of Electrical engineering and Computing, Department of Control and Computer Engineering in Automation, 2000.
- [3] Captain Pasetha Saralak, Lecturer at Chulachomlao Royal Military Academy, Mechanical Engineering Faculty, *A Self-tuning controller*
- [4] Nguyen Duc, Thanh - Nguyen Thi Phuong, Ha - Nguyen Xuan, Bac - Nguyen Duc, Hoang, *Controlling Speed of DC Motor Using Pole Placement Self Tuning Regulator Algorithm experimented by using DSpace-DS1104 Control Card*, Publishing House of Vietnam National University, 2009
- [5] Ghandakly, A.A, *Design of an adaptive controller for a DC motor within an existing PLC framework*, Industry Applications Conference, 1996. Thirty-First IAS Annual Meeting, IAS '96, Conference Record of the 1996 IEEE