

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



**ĐỒ ÁN LIÊN MÔN 4**

**ĐỀ TÀI: THIẾT KẾ VÀ SỬ DỤNG FPGA**  
**Ể ĐIỀU KHIỂN HỆ TRUYỀN ĐỘNG ĐIỆN**

NGƯỜI HƯỚNG DẪN	:	PGS. TS LÊ TIẾN DŨNG
SINH VIÊN THỰC HIỆN	:	LƯU CHÍ CƯỜNG
		TRẦN VĂN QUỐC BẢO
		ĐỖ TRỌNG VINH (Nhóm Trưởng)
NHÓM	:	9
NHÓM HỌC PHẦN	:	20NH36

*Đà Nẵng, tháng 6/2024*

# MỤC LỤC

LỜI CẢM ƠN .....	5
CHƯƠNG 1: TỔNG QUAN, THIẾT KẾ PHẦN CỨNG .....	6
1. Tổng quan.....	6
1.1 Lý do chọn đề tài.....	6
1.2 Mục tiêu của dự án.....	6
1.3 Nội dung đề tài.....	7
1.4 Biểu đồ Gantt của dự án.....	8
1.5 Bảng phân công nhiệm vụ.....	8
2. Nguyên lý chung của hệ thống.....	10
3. Thiết kế phần cứng.....	12
3.1 Phân tích các thành phần lực, momen của hệ .....	12
3.2 Xác định các thành phần momen .....	12
3.3 Tốc độ dự kiến của tải.....	13
3.4 Tốc độ dự kiến của động cơ .....	13
3.5 Quy đổi momen tải, momen quán tính của hệ về trục động cơ .....	14
3.6 Tính công suất của hệ.....	15
3.7 Chọn động cơ, hộp số và kiểm nghiệm.....	15
3.7.1 Chọn động cơ .....	15
3.7.2 Tính chọn hộp số .....	16
4. Lựa chọn phương án truyền động .....	17
4.1 Tính chọn mạch chỉnh lưu.....	17
4.2 Tính chọn nghịch lưu .....	19
4.3 Tính chọn bộ lọc.....	21
4.4 Tính chọn cảm biến.....	21
5.4.1 Tính chọn cảm biến dòng .....	22

5.4.2 Tính chọn cảm biến tốc độ .....	23
5. Trình bày kế hoạch các chương tiếp theo .....	25
5.1 Kế hoạch chương 2 .....	25
5.2 Kế hoạch chương 3 .....	25
5.3 Kế hoạch chương 4 .....	25
CHƯƠNG 2: TỔNG HỢP PHẦN ĐIỀU KHIỂN .....	26
1. Mô hình tổng quát hệ thống .....	26
2. Xây dựng các phương trình toán học .....	26
2.1 Mô hình không gian trạng thái của động cơ IM trong hệ tọa độ dq .....	26
2.2 Mô hình toán học của khâu mô hình từ thông .....	28
2.3 Mô hình toán học của khâu chuyển đổi hệ tọa độ uvw $\rightarrow \alpha\beta$ .....	29
2.4 Mô hình toán học của khâu chuyển đổi $\alpha\beta \rightarrow dq$ .....	30
2.5 Phương pháp SVPWM .....	31
3. Thiết kế bộ điều khiển .....	35
3.1 Thiết kế bộ điều khiển dòng điện .....	35
3.2 Thiết kế bộ điều khiển từ thông. ....	37
3.3 Thiết kế bộ điều khiển tốc độ .....	38
CHƯƠNG 3: THIẾT KẾ, TRIỂN KHAI FPGA .....	40
1. Lập trình bộ điều khiển PI .....	40
1.1 Công thức của bộ PI: .....	40
1.2 Lưu đồ thuật toán .....	40
1.3 Code lập trình bộ điều khiển PI .....	40
2. Lập trình khối biến đổi clark thuận .....	43
2.1 Công thức biến đổi clark thuận: .....	43
2.2 Lưu đồ thuật toán: .....	44
2.3. Code mô phỏng bộ biến đổi clark thuận: .....	44
3. Code biến đổi Park thuận .....	46

3.1. Công thức biến đổi park thuận: .....	46
3.2. Lưu đồ thuật toán: .....	46
3.3 Code mô phỏng bộ biến đổi Park thuận .....	47
4. Code khối Clark ngược sửa đổi – Park ngược .....	49
4.1 Công thức biến đổi Clark ngược sửa đổi và Park ngược: .....	49
4.2 Lưu đồ thuật toán.....	50
4.3 Code mô tả bộ biến đổi clark sửa đổi và park ngược .....	50
5. Lập trình thuật toán SVPWM .....	53
5.1 Lưu đồ thuật toán.....	53
5.2 Code mô phỏng .....	60
CHƯƠNG 4: KIỂM NGHIỆM VÀ ĐÁNH GIÁ.....	67
1. Mô phỏng và kiểm nghiệm mô hình trên Matlab .....	67
1.1 Các thông số cài đặt trong hệ thống .....	67
1.2 Kiểm nghiệm và đánh giá kết quả mô phỏng.....	68
1.3 Tốc độ, momen và dòng điện .....	69
1.4 Tương quan giữa momen và tốc độ .....	70
2. Mô phỏng và kiểm nghiệm code Modelsim trên Matlab/Simulink .....	71
2.1. Mô phỏng và kiểm nghiệm bộ điều khiển PI .....	71
2.2. Mô phỏng và kiểm nghiệm khối biến đổi Clark thuận .....	72
2.3. Mô phỏng và kiểm nghiệm khối biến đổi Clark, Park thuận .....	73
2.4. Mô phỏng và kiểm nghiệm khối Clark ngược sửa đổi – Park ngược .....	74
2.5. Mô phỏng và kiểm nghiệm thuật toán SVPWM.....	75
2.6. Mô phỏng sơ đồ toàn hệ thống.....	76
3. Kết luận chung .....	80
TÀI LIỆU THAM KHẢO.....	82

## **LỜI CẢM ƠN**

PBL4 (Project Based Learning 4) là một môn học yêu cầu kỹ năng kết hợp kiến thức từ nhiều môn học khác nhau và khai thác khả năng tự học, tự nghiên cứu của sinh viên. Đây là đồ án liên môn thứ 4 của sinh viên thuộc chuyên ngành Kỹ thuật Điều khiển và Tự động hóa Khóa 20 nói chung, nên chắc chắn rằng sinh viên vẫn sẽ gặp không ít khó khăn trong quá trình nghiên cứu và thực hiện đồ án. Đây cũng chính là lý do tại sao nhóm em rất cảm ơn Thầy Lê Tiến Dũng đã tạo mọi điều kiện thuận lợi nhất, tận tình giúp đỡ và giải đáp tất cả những thắc mắc của mọi người trong quá trình thực hiện đồ án để nhóm em có thể duy trì được tiến độ đã đặt ra và hoàn thành đồ án lần này.

Có lẽ sản phẩm chúng em vẫn chưa đạt được kết quả tốt nhất, nhưng với sự giúp đỡ và góp ý từ các Thầy Cô khoa Điện, chúng em đã học hỏi được rất nhiều kiến thức bổ ích và vẫn rất tự hào về những gì mà mình đã đạt được trong dự án lần này.

Nhóm em kính chúc các Thầy Cô khoa Điện thật nhiều sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy và học tập của mình.

## CHƯƠNG 1: TỔNG QUAN, THIẾT KẾ PHẦN CỨNG

### 1. Tổng quan

#### 1.1 Lý do chọn đề tài

Sự bùng nổ của tiến bộ khoa học trong lĩnh vực điện - điện tử - tin học hiện nay đã có những thay đổi sâu sắc cả về mặt lý thuyết lẫn thực tế lĩnh vực truyền động điện tự động. Trước hết phải kể đến sự ra đời và ngày càng hoàn thiện các bộ biến đổi điện tử công suất, với kích thước nhỏ gọn, nhẹ, độ tác động nhanh cao, dễ dàng ghép nối với các mạch điều khiển dùng mạch vi điện tử, vi xử lý... Các hệ truyền động điện tự động ngày nay thường sử dụng nguyên tắc điều khiển vector cho các động cơ xoay chiều. Phần lớn các mạch điều khiển này dùng kỹ thuật số với chương trình phần mềm linh hoạt, dễ dàng thay đổi các cấu trúc tham số hoặc các luật điều khiển, vì vậy làm tăng độ chính xác và hiệu suất cho các hệ truyền động.

FPGA có nhiều ưu điểm vượt trội so với các công nghệ khác và kể cả chi phí, có lợi thế trong việc cạnh tranh với thị trường. FPGA là viết tắt của cụm từ tiếng anh Field - Programmable Gate Array là một loại vi mạch tích hợp dùng cấu trúc mảng phần tử logic mà người dùng có thể lập trình được. Xuất phát từ những vấn đề thực tế đã nêu trên, chúng tôi đã chọn đề tài: **“Thiết kế và sử dụng FPGA để điều khiển hệ truyền động điện.”**

#### 1.2 Mục tiêu của dự án

Mục tiêu của đề tài là thiết kế và sử dụng FPGA để điều khiển hệ truyền động điện. Để đạt được mục tiêu này, các bước cần thực hiện bao gồm:

- + Nghiên cứu và phân tích yêu cầu của hệ thống truyền động điện: Xác định các thông số kỹ thuật, chức năng và yêu cầu của hệ thống truyền động điện.

- + Thiết kế mạch điều khiển truyền động điện sử dụng FPGA: Sử dụng công cụ thiết kế FPGA để thiết kế mạch điều khiển truyền động điện. Mạch điều khiển này có nhiệm vụ nhận tín hiệu từ các cảm biến và điều khiển các thiết bị truyền động điện, như động cơ, biến tần, bộ điều khiển PID, vv.

- + Lập trình FPGA: Sử dụng ngôn ngữ lập trình phù hợp (ví dụ: Verilog, VHDL) để lập trình FPGA và triển khai mạch điều khiển truyền động điện trên FPGA.

- + Kiểm tra và xác nhận hệ thống: Tiến hành kiểm tra và xác nhận hệ thống truyền động điện sau khi triển khai trên FPGA. Kiểm tra sẽ bao gồm việc kiểm tra tính

đúng đắn của mạch điều khiển và tính chính xác của các chức năng điều khiển truyền động điện.

+ Tối ưu hóa và cải tiến hệ thống: Dựa trên kết quả kiểm tra và phản hồi từ hệ thống, tiến hành tối ưu hóa và cải tiến hệ thống truyền động điện để đạt được hiệu suất và chất lượng tốt nhất.

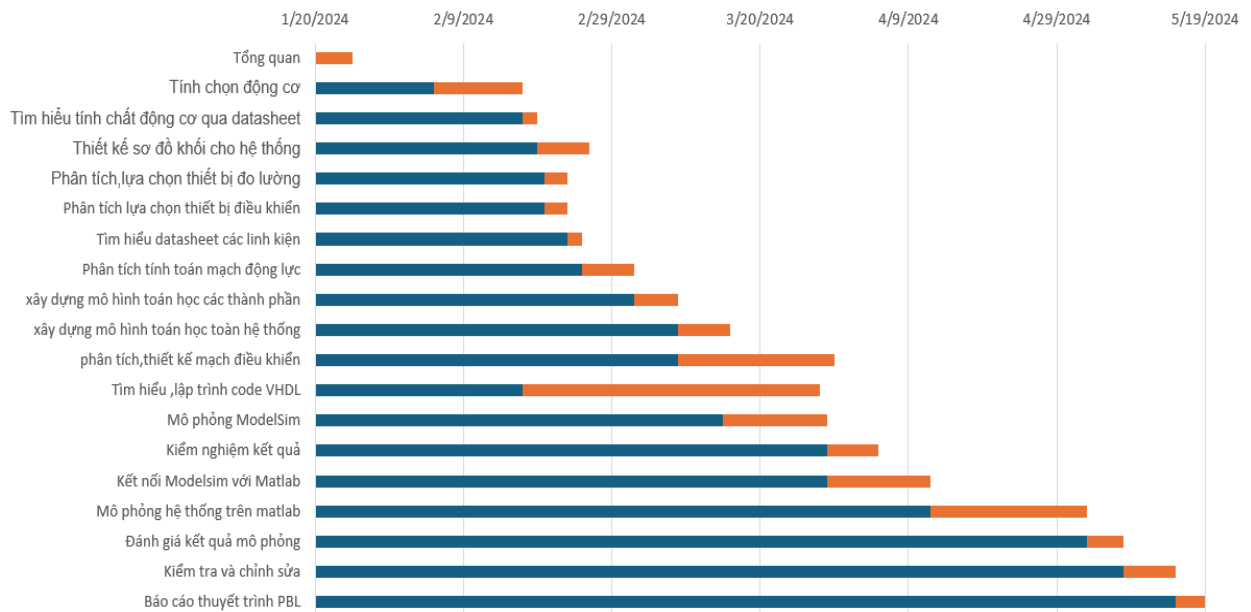
+ Đánh giá hệ thống: Đánh giá hiệu suất và hiệu quả của hệ thống truyền động điện.

+ Viết báo cáo và trình bày kết quả: Viết báo cáo kết quả nghiên cứu và trình bày kết quả cho cộng đồng nghiên cứu và các nhà quản lý liên quan.

### ***1.3 Nội dung đề tài***

- Tìm hiểu nguyên lý chung của hệ thống.
- Tính chọn phần cứng của hệ truyền động.
- Xây dựng mô hình toán học động cơ không đồng bộ xoay chiều ba pha rotor lồng sóc.
- Tìm hiểu phương pháp điều khiển vector, điều khiển động cơ dựa trên cơ sở điều chế vector không gian.
- Xây dựng cấu trúc hệ truyền động điều khiển vị trí động cơ không đồng bộ xoay chiều ba pha.
- Thiết kế các bộ điều khiển có trong hệ thống.
- Tổng hợp toàn bộ hệ thống.
- Lập trình hệ thống trên Modelsim bằng ngôn ngữ VHDL.
- Mô phỏng, kiểm nghiệm hệ thống trên Matlab Simulink.

#### 1.4 Giải đồ Gantt của dự án



Hình 1.1: Giải đồ Gantt của dự án

#### 1.5 Bảng phân công nhiệm vụ

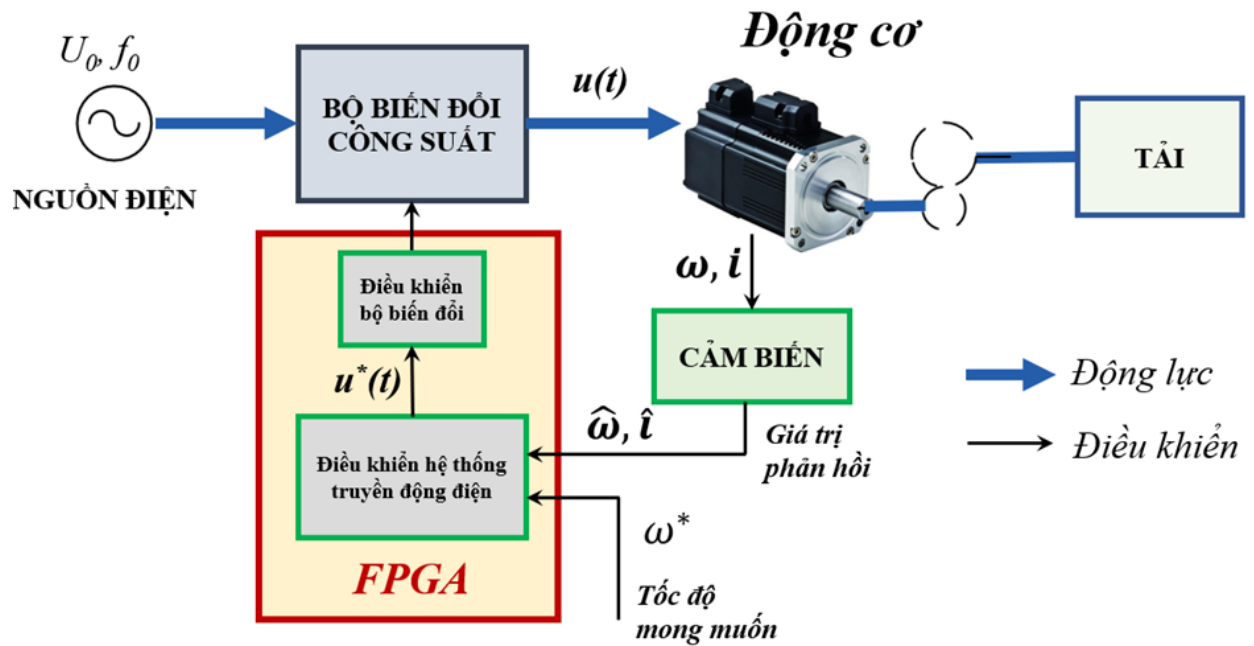
Tên thành viên	Nhiệm vụ
Lưu Chí Cường (NT)	<ul style="list-style-type: none"> <li>- Nghiên cứu và viết phần: Điều khiển vị trí động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc</li> <li>- Tìm hiểu về nguyên tắc và phương pháp điều khiển vị trí động cơ không đồng bộ xoay chiều 3 pha</li> <li>- Mô tả cách thiết kế và lập trình thuật toán điều khiển vị trí</li> <li>- Đánh giá hiệu suất và độ chính xác của hệ thống điều khiển vị trí</li> <li>- Nhận xét giữa các phương pháp điều khiển</li> <li>- Tìm hiểu, viết code VHDL, mô phỏng kết nối Matlab</li> </ul>
Trần Văn Quốc Bảo	<ul style="list-style-type: none"> <li>- Nghiên cứu và viết phần: Tổng quan động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc</li> <li>- Tính chọn thiết bị</li> <li>- Mô phỏng động cơ, các bộ biến đổi, hệ thống điều khiển</li> <li>- Mô tả cách hoạt động của động cơ trong các trường hợp khác nhau</li> <li>- Đóng góp ý kiến và hỗ trợ trong viết phần: Điều khiển vị trí động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc</li> </ul>



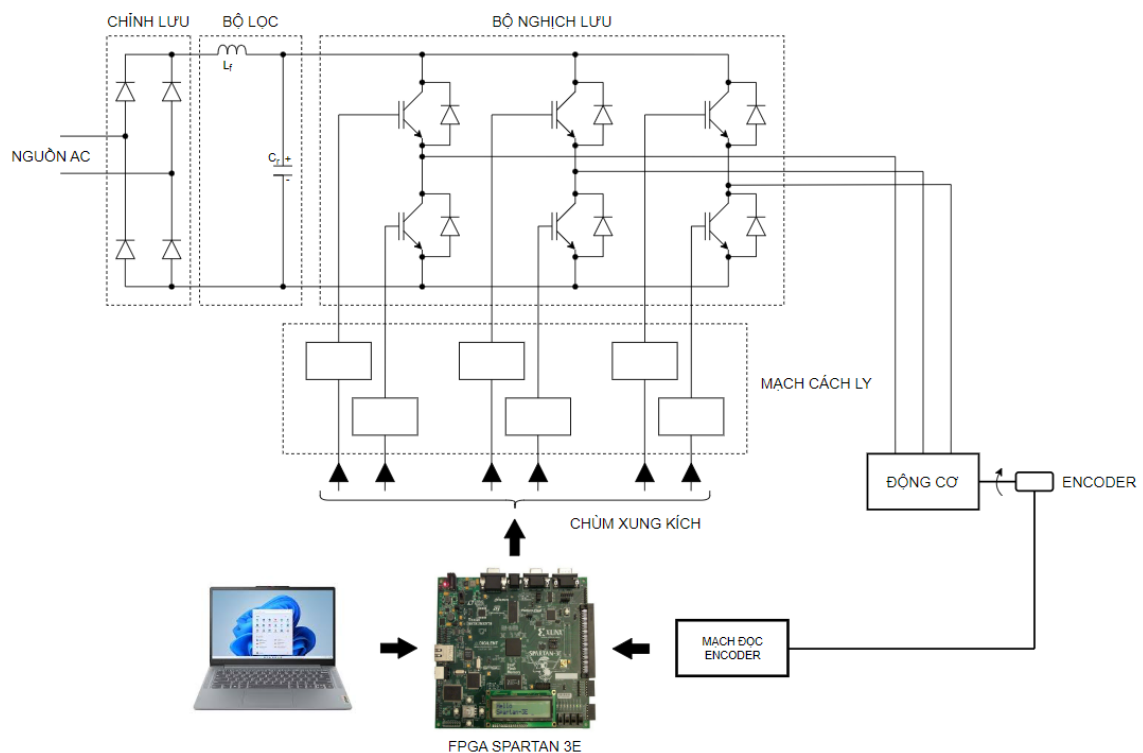
	<ul style="list-style-type: none"> <li>- Nhận xét mô phỏng, kiểm nghiệm</li> <li>- Tìm hiểu, viết code VHDL, mô phỏng kết nối Matlab</li> </ul>
Nguyễn Bá Phong Đình	<ul style="list-style-type: none"> <li>- Đóng góp ý kiến và hỗ trợ trong viết phần: Sử dụng FPGA thiết kế bộ điều khiển vị trí động cơ không đồng bộ xoay chiều 3 pha.</li> <li>- Đóng góp ý kiến và hỗ trợ trong viết phần: Tổng quan động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc.</li> <li>- Kiểm tra độ chuẩn xác và sắp xếp lại cấu trúc và logic của các chương.</li> <li>- Chỉnh sửa và biên tập các phần viết để đảm bảo tính logic, ngữ pháp và chính xác của nội dung.</li> <li>- Đảm bảo sự nhất quán và chất lượng của bài viết cuối cùng.</li> </ul>
Đỗ Trọng Vinh	<ul style="list-style-type: none"> <li>- Nghiên cứu và viết phần: Biến tần và hệ thống biến tần - động cơ không đồng bộ xoay chiều 3 pha lồng sóc.</li> <li>- Tìm hiểu về công nghệ biến tần và vai trò của nó trong việc điều khiển động cơ không đồng bộ xoay chiều 3 pha.</li> <li>- Mô tả cách hoạt động của hệ thống biến tần và các thành phần chính trong nó.</li> <li>- Đánh giá ưu điểm và hạn chế của việc sử dụng biến tần trong ứng dụng này.</li> <li>- Đóng góp ý kiến và hỗ trợ trong viết phần: Sử dụng FPGA thiết kế bộ điều khiển vị trí động cơ không đồng bộ xoay chiều 3 pha.</li> </ul>
Nguyễn Trọng Thanh	<ul style="list-style-type: none"> <li>- Nghiên cứu và viết phần: Sử dụng FPGA thiết kế bộ điều khiển vị trí động cơ không đồng bộ xoay chiều 3 pha.</li> <li>- Tìm hiểu về công nghệ FPGA và khả năng điều khiển động cơ không đồng bộ xoay chiều 3 pha bằng FPGA.</li> <li>- Mô tả cách thiết kế và triển khai bộ điều khiển vị trí động cơ sử dụng FPGA.</li> <li>- Đánh giá hiệu suất và ưu điểm của việc sử dụng FPGA trong ứng dụng này.</li> <li>- Đóng góp ý kiến và hỗ trợ trong viết phần: Tổng quan động cơ không đồng bộ xoay chiều 3 pha rotor lồng sóc.</li> </ul>

Bảng 1.1: Phân chia nhiệm vụ

## 2. Nguyên lý chung của hệ thống



Hình 1.2: Sơ đồ nguyên lý



Hình 1.3: Sơ đồ chi tiết

Hệ truyền động điện bao gồm 2 phần: phần điện và phần cơ. Trung tâm của hệ là động cơ xoay chiều 3 pha không đồng bộ.

Chức năng của từng khối:

- Trung tâm của hệ là động cơ điện xoay chiều 3 pha không đồng bộ, rotor của động cơ sẽ nối trực truyền động cho tải
- Bộ biến đổi công suất: biến đổi điện áp lưới xoay chiều 3 pha cố định sang điện áp xoay chiều 3 pha phù hợp (biên độ, tần số), thay đổi được cấp cho động cơ
- Bộ điều khiển: Ở dự án lần này chúng ta sử dụng vi điều khiển FPGA để tính toán điện áp mong muốn cần phải đặt vào động cơ mong muốn phải đặt vào động cơ là bao nhiêu dựa trên thông tin phản hồi của động cơ và giá trị mong muốn, giá trị điện áp đó sẽ gửi tín hiệu đến bộ biến đổi công suất để thực hiện biến đổi điện áp thực bằng với điện áp tính toán
- Cảm biến: đo các giá trị thực của động cơ (dòng, điện áp...)

Nguyên lý hoạt động:

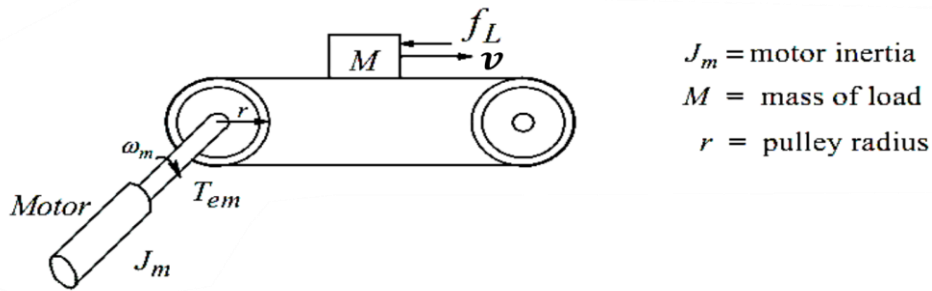
Trong từng sampling time thì nó sẽ cập nhật thông tin ở đầu vào bao gồm giá trị mong muốn và giá trị đo được từ cảm biến nó tính toán (chỉ vào bộ điều khiển) theo chương trình được lập trình sẵn cho ra kết quả điện áp cần đặt vào động cơ là bao nhiêu, sau đó sẽ tính toán và gửi tín hiệu phát xung đến các van điều khiển để điều chế độ rộng xung tạo ra điện áp điều khiển động cơ sao cho điện áp thực bằng điện áp mong muốn và như vậy giá trị điện áp tính toán ra được là giá trị tính toán và nó sẽ được gửi đến, ra lệnh cho bộ biến đổi để thực hiện các giá trị điện áp thực bằng giá trị tính toán. Như vậy giá trị thực sẽ cấp nguồn cho động cơ và động cơ sẽ hoạt động để thay đổi và giá trị thực về tốc độ, moment, vị trí sẽ bám theo giá trị mong muốn và điều khiển tải theo yêu cầu tải cần hoạt động như thế nào.

Quá trình này cứ tính toán và lặp lại theo từng chu kỳ lấy mẫu và thường giá trị bên trong bộ điều khiển là giá trị số và tính toán, hoạt động theo những phương trình, chương trình còn những cái bên ngoài là giá trị thực, giá trị vật lý và trong hệ thống này những mạch như bộ biến đổi đến động cơ đến tải và ngược lại, dòng năng lượng đi từ nguồn qua bộ biến đổi đến động cơ đến tải là dòng, có công suất lớn, phần này gọi là phần lực.

Tùy theo chế độ hoạt động là từ chế độ động cơ từ điện năng chuyển thành cơ năng hoặc cũng có thể là ngược lại là động cơ hoạt động ở chế độ máy phát thì chiều năng lượng là đi ngược lại từ cơ năng thành điện năng có thể là để triệt tiêu cơ năng trong các chế độ hãm và phần điều khiển thì được biểu diễn bởi những đường nét đứt và phần điều khiển thì nó điều khiển các quá trình biến đổi năng lượng này.

### 3. Thiết kế phần cứng

#### 3.1 Phân tích các thành phần lực, momen của hệ



Hình 1.4: Phân tích các thành phần lực, momen của hệ

Trong đó:

Khối lượng tải: 100kg

Độ dài băng tải: 3m

Bán kính trục:  $r = 0,1\text{m}$ .

$J_M$ : Momen quán tính (Nm).

$\omega_M$ : Tốc độ quay trên trục động cơ (rad/s)

$u$ : Tốc độ chuyển động của tải (m/s)

Giả sử  $f_{cản} = 12.5\text{N}$

#### 3.2 Xác định các thành phần momen

Áp dụng định luật II Newtons cho hệ:

$$F - f_{cản} = ma \quad (1)$$

Và công thức:  $u = r\omega_M \quad (2)$

Từ (1) và (2) Ta có:

$$F = mr \frac{d\omega}{dt} + f_{cản}$$

Momen tải quy về trục động cơ:

$$M_C = (mr \frac{d\omega}{dt} + f_{cản}) \times r \quad (3)$$

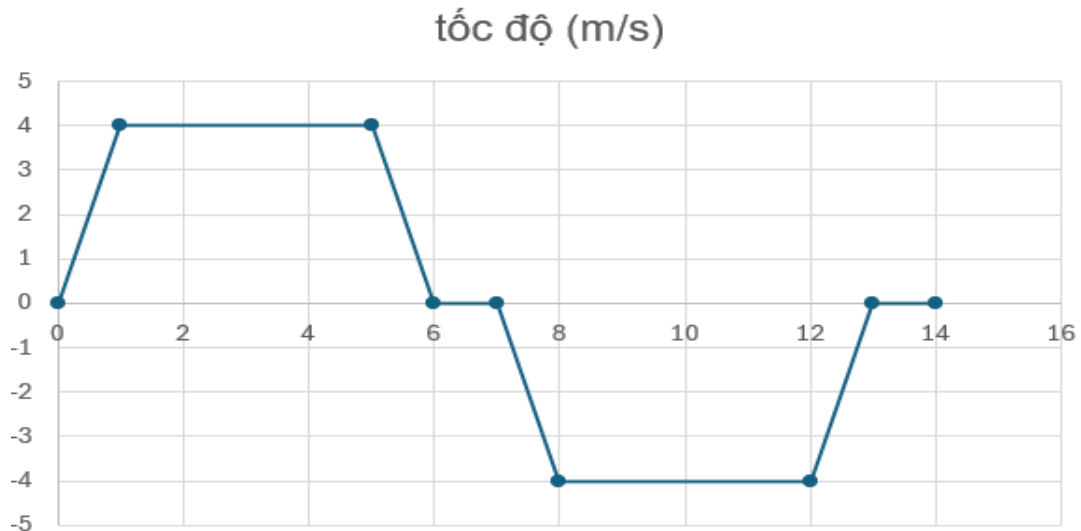
Momen động lượng của hệ là:

$$L = J_m \frac{d\omega}{dt}$$

Với  $J_m$  là momen quán tính của hệ.

### 3.3 Tốc độ dự kiến của tải

Giai đoạn 1: ( $t=0s \rightarrow 1s$ )	Tải từ $v = 0m/s$ tăng $\rightarrow v = 4m/s$
Giai đoạn 2: ( $t=1s \rightarrow 5s$ )	Tải giữ nguyên tốc độ $v = 4m/s$
Giai đoạn 3: ( $t=5s \rightarrow 6s$ )	Tải từ $v = 4 m/s$ giảm $\rightarrow v = 0m/s$
Giai đoạn 4: ( $t=6s \rightarrow 7s$ )	Động cơ đứng yên
Giai đoạn 5: ( $t=7s \rightarrow 8s$ )	Đảo chiều, tải từ $v=0 \rightarrow v = -4m/s$
Giai đoạn 6: ( $t=8s \rightarrow 12s$ )	Tải giữ nguyên tốc độ $v = -4m/s$
Giai đoạn 7: ( $t=12s \rightarrow 13s$ )	Tải từ $v= 4 m/s \rightarrow v = 0m/s$
Giai đoạn 8: ( $t=13s \rightarrow 14s$ )	Động cơ đứng yên

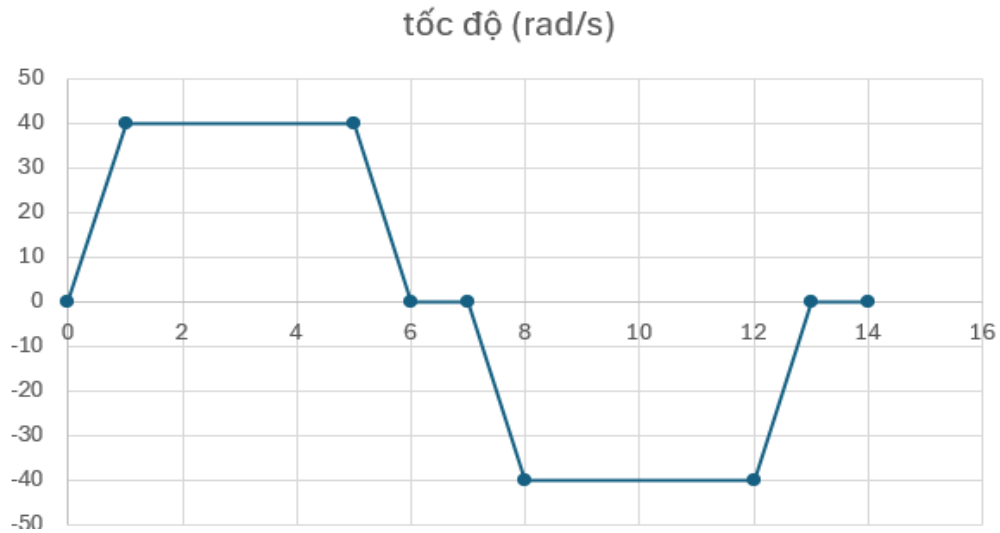


Hình 1.5: Đồ thị tốc độ mong muốn của tải

### 3.4 Tốc độ dự kiến của động cơ

$$\omega = \frac{u}{r} = \frac{0,4}{1} = 40 \frac{\text{rad}}{\text{s}} \approx 382 \text{ vòng /phút}$$

Từ công thức trên ta có đồ thị tốc độ dự kiến của động cơ như sau:



Hình 1.6: Tốc độ dự kiến của động cơ

### 3.5 Quy đổi momen tải, momen quán tính của hệ về trục động cơ

Công thức tính momen điện từ của động cơ:

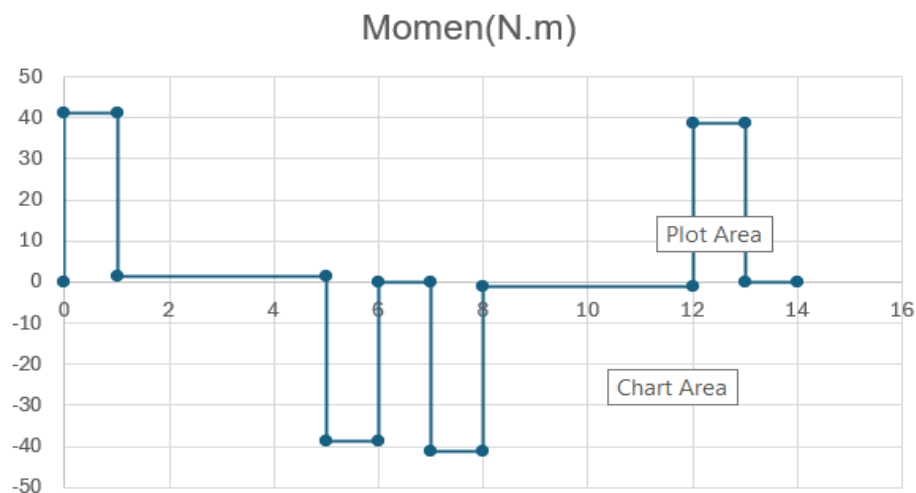
$$M_e = M_c + J_m \frac{d\omega}{dt} + r \times f_{cản}$$

Vì  $J_m$  là momen quán tính của hệ, Với  $J_m$  là momen quán tính của hệ và  $J_m \frac{d\omega}{dt}$  là momen yêu cầu để động cơ quay, vì giá trị này rất nhỏ nên ta tạm thời bỏ qua.

Vậy ta tính Momen cần để động cơ cung cấp cho tải theo từng giai đoạn với công thức:

$$M_e = r^2 m \frac{d\omega}{dt} + r \times f_{cản}$$

Từ đây ta có đồ thị momen động cơ:



Hình 1.7: Đồ thị momen điện từ

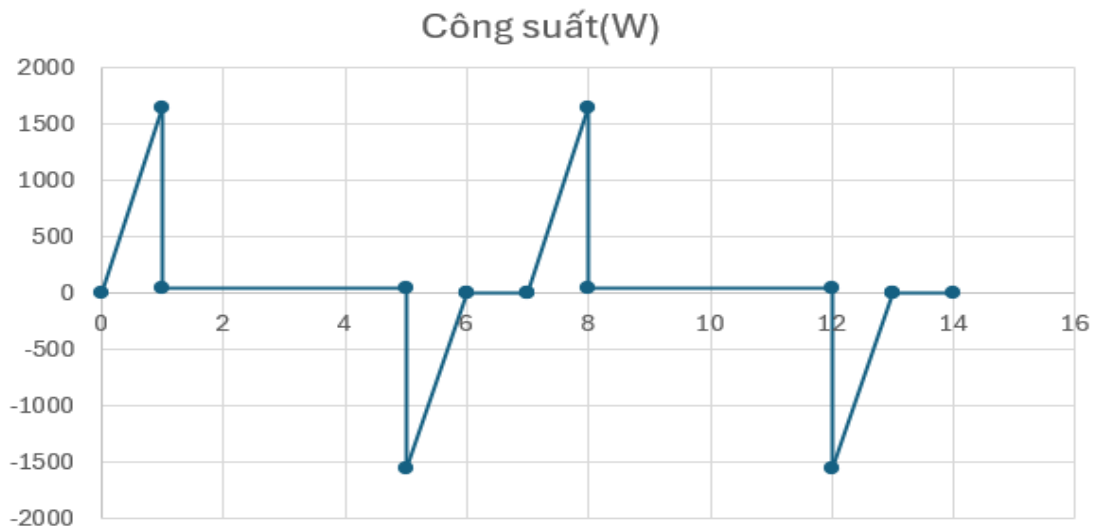
Do momen điện từ có thành phần âm nên không dùng để tính toán, ta tính momen đẳng trị

$$M_{dt} = \sqrt{\frac{\sum_j^n T_{emj}^2 t_j}{\sum_j^n t_j}} = 22 \text{ (Nm)}$$

### 3.6 Tính công suất của hệ

Công suất động cơ tại các giai đoạn:  $P_i = T_i \times \omega_m$

Từ đây ta có đồ thị công suất dự kiến



Hình 1.8: Đồ thị công suất của hệ

Ta có :

$$P_{dt} = \sqrt{\frac{\sum P_i^2 \times \Delta t_i}{\sum \Delta t_i}} \sum P_i^2 \times \Delta t_i$$

$$= (P_1)2\Delta t_1 + (P_2)2\Delta t_2 + (P_3)2\Delta t_3 + (P_4)2\Delta t_4 + (P_5)2\Delta t_5$$

$$+ (P_6)2\Delta t_6 + (P_7)2\Delta t_7 + (P_8)2\Delta t_8$$

Từ đây ta có :

$$P_{dt} = \sqrt{\frac{(1650)^2 \times 2 + 1550^2 \times 2 + 50^2 \times 8}{14}} = 856 \text{ W}$$

Chọn hệ số dự trữ  $k=1.2$  động cơ cần thỏa mãn

$$T_{eq} = 1.2 \times M_{dt} = 26.4 \text{ (N.m)}$$

$$P_{eq} = 1.2 \times P_{dt} = 1028 \text{ (W)}$$

### 3.7 Chọn động cơ, hộp số và kiểm nghiệm

#### 3.7.1 Chọn động cơ

Dựa vào  $P_{eq}$ ,  $T_{eq}$  tính ở trên ta lựa chọn động cơ KE90S2-2 với các thông số:

Công suất định mức ( $P_{dm}$ )	1.5KW
Điện áp định mức ( $U_{dm}$ )	380 V
Tần số định mức (f)	50 Hz
Tốc độ định mức ( $n_{dm}$ )	2900 vòng/phút = 305 rad/s
Dòng điện định mức ( $I_{dm}$ )	3.19A
Momen định mức ( $T_{dm}$ )	4.94 Nm
Momen tối đa	11.362 Nm
Hiệu suất	81.3%

Bảng 1.2: Thông số động cơ KE90S2-2



Hình 1.9: Động cơ 3 pha không đồng bộ KE90S2-2

### 3.7.2 Tính chọn hộp số

Momen đẳng trị của động cơ mà bài toán yêu cầu là  $T_{eq}=26.4$  (N.m)

Momen định mức của động cơ lựa chọn là 4.94 (N.m)

Vì vậy ta cần sử dụng thêm 1 hộp số giảm tốc

$$\text{Hệ số ratio} = \frac{T_{eq}}{T_{dm}} = 5.3$$

→ Chọn ratio =6

Sau khi có hộp số  $T_{dm}=4.94 > T_{eq}=4.4$ (N.m)

Như vậy ta cần gắn thêm hộp số có tỉ số 6:1 để đáp ứng tốc độ, momen tương ứng vẫn đảm bảo yêu cầu tốc độ sau khi lắp hộp giảm tốc là 483 v/p > tốc độ yêu cầu: 381 v/p

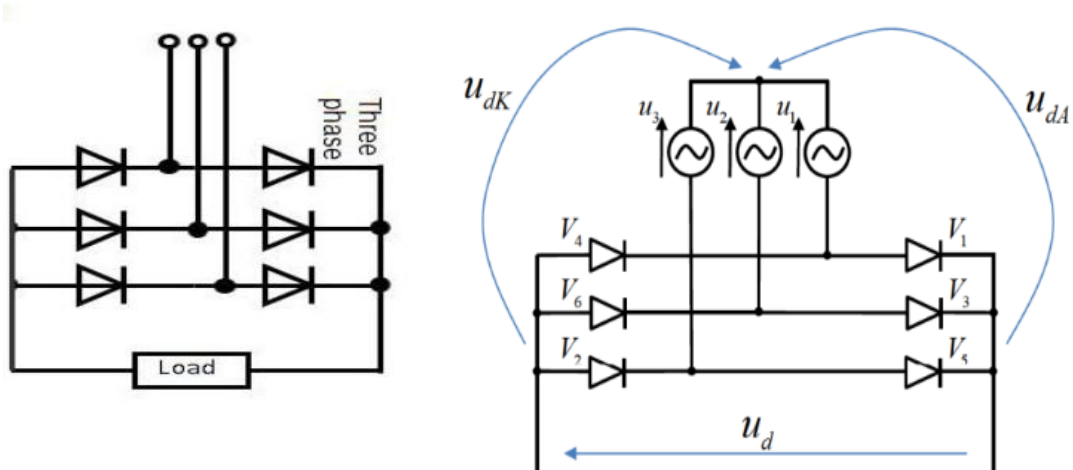
Từ tỉ số ratio ta chọn hộp số giảm tốc DKM:UBK-6BH



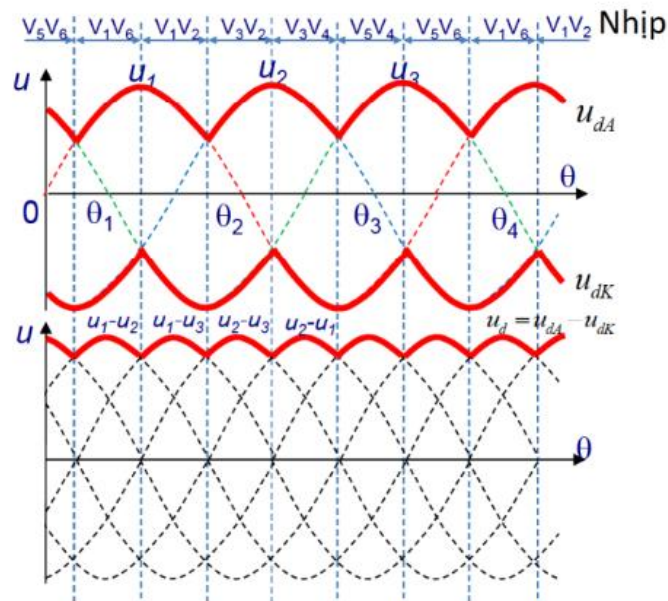
#### 4. Lựa chọn phương án truyền động

##### 4.1 Tính chọn mạch chỉnh lưu

Nguyên lý mạch chỉnh lưu:



Hình 1.10: Cấu tạo mạch chỉnh lưu cầu 3 pha diode



Hình 1.11: Đồ thị điện áp dây sau chỉnh lưu

- Từ  $0^\circ$  đến  $30^\circ$

Ở chu kỳ điện áp  $U_3$  lớn nhất nên  $V_5$  dẫn và  $U_2$  nhỏ nhất nên  $V_6$  dẫn. Dòng điện đi từ nguồn  $U_3$  qua  $V_5$ , qua tải, đến  $V_6$  về nguồn  $U_2$ . Nên điện áp chỉnh lưu  $U_d = U_3 - U_2$ .

- Từ  $30^\circ$  đến  $60^\circ$

Ở chu kỳ điện áp  $U_1$  lớn nhất nên  $V_1$  dẫn và  $U_2$  nhỏ nhất nên  $V_6$  dẫn. Dòng điện đi từ nguồn  $U_1$  qua  $V_1$ , qua tải đến  $V_6$  về nguồn  $U_2$ . Nên điện áp chỉnh lưu  $U_d = U_1 - U_2$ .

- Từ  $60^\circ$  đến  $90^\circ$

Ở chu kì điện áp  $U_1$  lớn nhất nên  $V_1$  dẫn và  $U_3$  nhỏ nhất nên  $V_2$  dẫn. Dòng điện đi từ nguồn  $U_1$  qua  $V_1$ , qua tải đến  $V_2$  về nguồn  $U_3$ . Nên điện áp chỉnh lưu  $U_d = U_1 - U_3$ .

- Từ  $90^\circ - 120^\circ$

Ở chu kì điện áp  $U_2$  lớn nhất nên  $V_3$  dẫn và  $U_3$  nhỏ nhất nên  $V_2$  dẫn. Dòng điện đi từ nguồn  $U_2$  qua  $V_3$ , qua tải đến  $V_2$  về nguồn  $U_3$ . Nên điện áp chỉnh lưu  $U_d = U_2 - U_3$ .

- Từ  $120^\circ$  đến  $150^\circ$

Ở chu kì điện áp  $U_2$  lớn nhất nên  $V_3$  dẫn,  $U_1$  nhỏ nhất nên  $V_4$  dẫn. Dòng điện đi từ nguồn  $U_2$  đến  $V_3$ , qua tải đến  $V_4$  về nguồn  $U_1$ . Nên điện áp chỉnh lưu  $U_d = U_2 - U_1$ .

- Từ  $150^\circ$  đến  $180^\circ$

Ở chu kì điện áp  $U_3$  lớn nhất nên  $V_5$  dẫn,  $U_1$  nhỏ nhất nên  $V_4$  dẫn. Dòng điện đi từ nguồn  $U_3$  qua  $V_5$ , qua tải đến  $V_4$  về nguồn  $U_1$ . Nên điện áp chỉnh lưu  $U_d = U_3 - U_1$ .

Cấu trúc bộ chỉnh lưu cầu 3 pha:

Ta sử dụng bộ chỉnh lưu cầu 3 pha không điều khiển với các van bán dẫn là các diode.

Điện áp xoay chiều 3 pha: 220/380V

Điện áp sau bộ chỉnh lưu:

$$U_d = \frac{3\sqrt{6}}{\pi} U = \frac{3\sqrt{6}}{\pi} 220 = 515 \text{ V}$$

Điện áp ngược đặt lớn nhất đặt lên diode:

$$U_{ng} = U\sqrt{6} = 220\sqrt{6} = 539 \text{ V}$$

Chọn hệ số dự trữ  $K = 1,3$

Nên điện áp ngược  $U'_{ng} = KU_{ng} = 1,3.539 = 700 \text{ V}$

Dòng điện định mức của động cơ:  $I_{dm} = 3.19 \text{ A}$

Suy ra dòng điện trung bình qua các van:

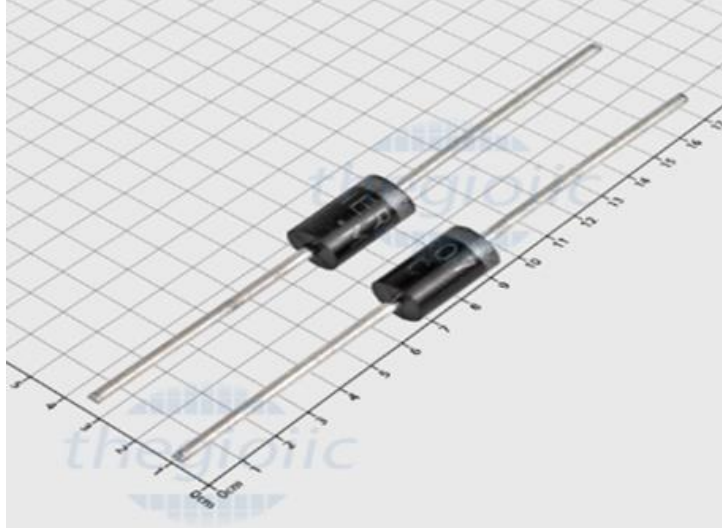
$$I_{van} = \frac{I_{dm}}{3} = \frac{3.19}{3} = 1.1 \text{ A}$$

$$I'_{van} = KI_{van} = 1,3.1,1 = 1,43 \text{ A}$$

Vậy chọn diode HER507 có các thông số:

Điện áp ngược:  $U_{ng} = 800 \text{ V}$

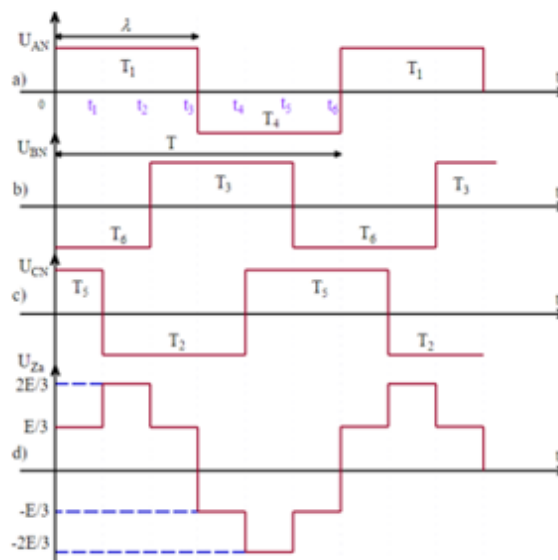
Dòng định mức:  $I_{dm} = 5 \text{ A}$



Hình 1.12: Diode HER507

#### 4.2 Tính chọn nghịch lưu

Nguyên lý mạch chỉnh lưu:



Hình 1.13: Sơ đồ điện áp ra sau nghịch lưu

Để tạo ra điện áp 3 pha đối xứng, luật dẫn điện của các van phải tuân theo:

T1 và T4 dẫn lệch nhau  $\pi$  và tạo ra pha A

T2 và T5 dẫn lệch nhau  $\pi$  và tạo ra pha C

T3 và T6 dẫn lệch nhau  $\pi$  và tạo ra pha B

Các pha lệch nhau góc  $\frac{2\pi}{3}$ .

Dạng điện áp ra trên tải được xác định:

Trong khoảng  $0 \rightarrow t_1$ : T1, T6, T5 dẫn, ta thấy  $U_{ZA} = E/3$ .

Trong khoảng  $t_1 \rightarrow t_2$ : T1, T2, T6 dẫn, ta thấy  $U_{ZA} = 2E/3$ .

Trong khoảng  $t_2 \rightarrow t_3$ : T1, T2, T3 dẫn, ta thấy  $U_{ZA} = E/3$ .

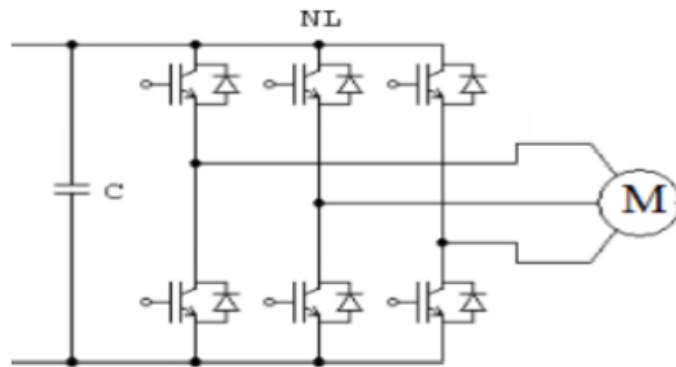
Trong khoảng  $t_3 \rightarrow t_4$ : T2, T3, T4 dẫn, ta thấy  $U_{ZA} = -E/3$ .

Trong khoảng  $t_4 \rightarrow t_5$ : T3, T4, T5 dẫn, ta thấy  $U_{ZA} = -2E/3$ .

Trong khoảng  $t_5 \rightarrow t_6$ : T4, T5, T6 dẫn, ta thấy  $U_{ZA} = -E/3$ .

Qui trình tiếp tục như vậy trong các chu kì khác.

Cấu trúc bộ nghịch lưu 3 pha :



Hình 1.14: Cấu trúc bộ nghịch lưu

→ Ta chọn hệ số dự trữ điện áp  $K_u = 1,2$

Điện áp ngược đặt lên IGBT là  $U_{ng} = K_u U_d = 1,2 \cdot 515 = 618 \text{ V}$

Dòng điện qua mỗi van  $I_{dm} = \frac{P_{dm}}{\sqrt{3} \cdot U} = \frac{1500}{\sqrt{3} \cdot 220} = 4 \text{ A}$

Chọn hệ số dự trữ dòng điện  $K_i = 1,3$

$$\Rightarrow I_v = K_i I_{dm} = 1,3 \cdot 4 = 5,2 \text{ A}$$

Chọn IGBT 75N65FL2 có  $U_{max} = 650 \text{ V}$ ,  $I_{max} = 75 \text{ A}$



Hình 1.15: IGBT 75N65FL2

#### 4.3 Tính chọn bộ lọc

Để giảm sự nhấp nhô của dòng điện sau chỉnh lưu ta có thể dùng cuộn cảm L và để giảm sự nhấp nhô của điện áp sau chỉnh lưu ta dùng tụ điện C. Vì thế chọn bộ lọc LC tuân thủ theo công thức:

$$LC = \frac{K_{sb} + 1}{(p \cdot 2 \cdot \pi \cdot f)^2}$$

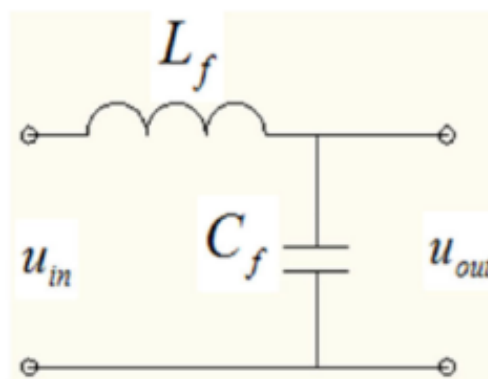
Trong đó:  $p$  là số xung đập mạch trong 1 chu kỳ.

$K_{sb}$  là hệ số san bằng (tối thiểu)

Tại đây ta dùng bộ chỉnh lưu cầu 3 pha nên có  $p = 6$ ;  $f = 50 \text{ Hz}$

Ta chọn  $C = 10^{-2} \text{ F}$  và  $K_{sb} = 60$

Suy ra: 
$$L = \frac{K_{sb} + 1}{(p \cdot 2 \cdot \pi \cdot f)^2 \cdot C} = \frac{60 + 1}{(6 \cdot 2 \cdot \pi \cdot 50)^2 \cdot 10^{-2}} = 0.018 \text{ H}$$



Hình 1.16: Mạch lọc 1 chiều LC

#### 4.4 Tính chọn cảm biến

Trong quá trình động cơ hoạt động cần phải sử dụng cảm biến đo đặc giá trị đầu ra để xem có đáp ứng được yêu cầu của người điều khiển không, ngoài ra cần phải đo

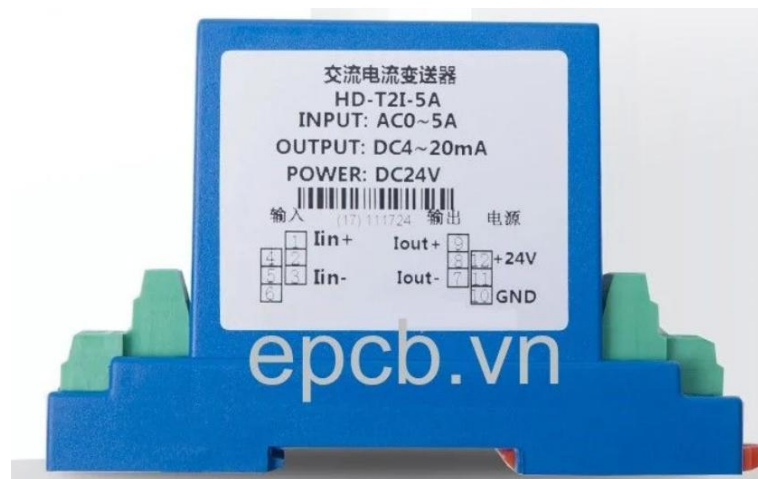
đặc giá trị đầu ra của các giá trị để phản hồi về bộ điều khiển, qua đó tính toán để hệ hoạt động đúng với mong muốn.

Trong dự án lần này, nhóm sẽ sử dụng 2 loại cảm biến là cảm biến dòng và cảm biến tốc độ.

#### 5.4.1 Tính chọn cảm biến dòng

Ta có dòng điện định mức của động cơ là 3.19A

Từ đây ta chọn cảm biến CR4150-5:



Hình 1.17 : Cảm biến dòng

Thông số kỹ thuật :

Ngõ vào		Ngõ ra	
Dòng điện ngõ vào	AC 0-5A	Tín hiệu ngõ ra	4 – 20mA ; 0 – 20mA ; 0 – 5V ; 0 – 10V
Tần số	40Hz – 60Hz	Điện trở tải ngõ ra	RL < 500 $\Omega$ (Đầu ra là tín hiệu dòng điện) RL > 10 k $\Omega$ (Đầu ra là tín hiệu điện áp)

Nguồn cấp	AC85-265V hoặc DC24V, $\pm 10\%$
Công suất tiêu thụ định mức	< 1W (1 ngõ vào, 1 ngõ ra. Nguồn 24VDC, ngõ ra 20mA) < 1.5W (1 ngõ vào, 1 ngõ ra. Nguồn 220V, ngõ ra 20mA)
Thời gian đáp ứng	< 400ms (0 – 90%)
Độ chính xác	< 0,5%
Cách điện	2000VAC/1min (Ngõ vào, ngõ ra, nguồn điện)
Vật liệu chống điện	> 100M $\Omega$ (Giữa ngõ vào, ngõ ra, nguồn điện)
Nhiệt độ hoạt động	– 20°C tới 55°C
Tương thích điện từ	GB/T 18268 (IEC61326-1)

Bảng 1.3 : Thông số kỹ thuật của cảm biến CR4150-5

#### 5.4.2 Tính chọn cảm biến tốc độ

Để đo giá trị tốc độ của động cơ và chuyển giá trị tốc độ tức thời của động cơ thành dạng tín hiệu cung cấp cho vi điều khiển ta sử dụng bộ cảm biến encoder với nguyên tắc chia điểm ở 1 vị trí cố định và đo góc quay. Sau đó encoder sẽ chuyển các giá trị đo được thành xung vuông cấp cho vi điều khiển.

Tốc độ định mức của động cơ là 389 vòng/phút từ đây ta chọn encoder Omron E6A2-CW5C-500P/R-2M.



Hình 1.18 : Encoder Omron E6A2-CW5C-500P/R-2M

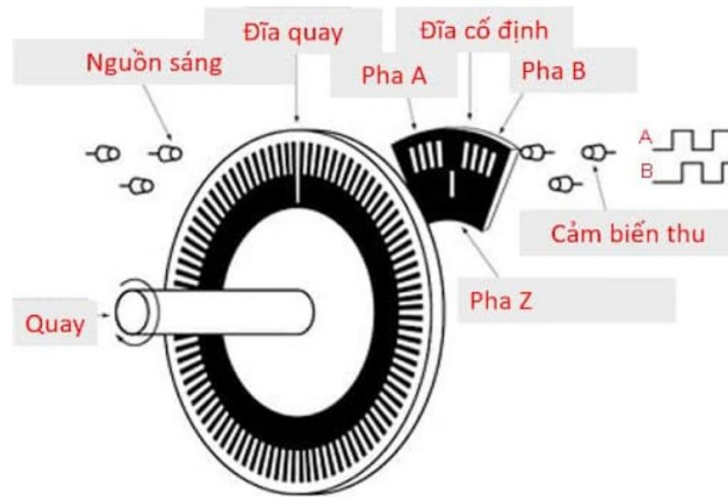
Độ phân giải	500P/R
Dòng điện xâm nhập	Khoảng 9A
Các pha đầu ra	A, B
Đầu ra	Tối đa 30mA
Chênh lệch pha đầu ra	$90 \pm 45$ độ giữa A và B
Tối đa vòng quay cho phép	5000/vòng/phút
Độ bền điện môi giữa các bộ phận được sạc và vỏ	500VAC, 50/60Hz 1

Bảng 1.4: Thông số cảm biến tốc độ

Công thức tính tốc độ cho động cơ từ encoder :

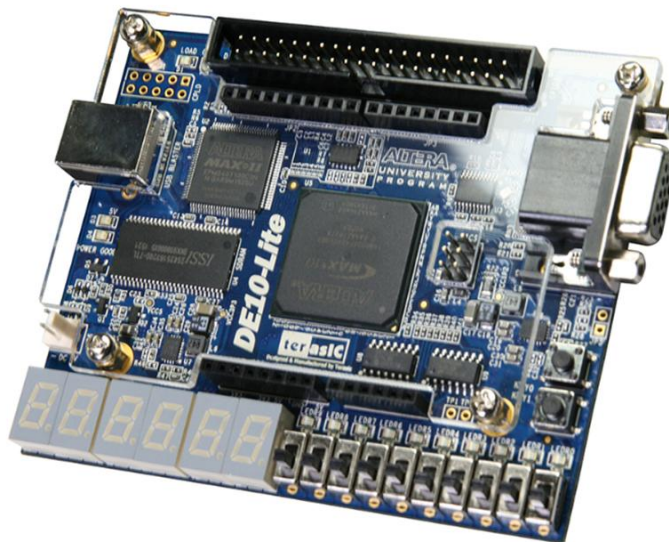
$$\text{Tốc độ (vòng/phút)} = \frac{\text{số xung đếm được trong thời gian lấy mẫu} \times 60}{\text{thời gian lấy mẫu (s)} \times \text{độ phân giải encoder} \left(\frac{P}{R}\right)}$$





Hình 1.19: Nguyên lý hoạt động của Encode

Trình bày tổng quan phương án phân điều khiển bằng FPGA.  
Sử dụng bộ điều khiển FPGA DE10 - Lite Board



Hình 1.20: Bộ điều khiển FPGA DE10 – Lite Board

DE10-Lite Board là một kit phát triển FPGA (Field - Programmable Gate Array) được sản xuất bởi Terasic, một công ty chuyên về thiết kế và phát triển các sản phẩm liên quan đến FPGA. Kit này cung cấp một môi trường phát triển linh hoạt cho việc thử nghiệm và phát triển các ứng dụng liên quan đến vi điều khiển, xử lý tín hiệu, truyền thông và nhiều ứng dụng khác.

DE10 - Lite Board được xây dựng trên cơ sở chip FPGA Cyclone IV EP4CE22E22C8N của Altera (hiện đã thuộc Intel). Nó cung cấp một loạt các tính năng và cổng giao tiếp như GPIO, VGA, USB, và cổng âm thanh, cho phép người dùng tạo ra các hệ thống phức tạp và đa dạng.

Một số tính năng chính của DE10 - Lite Board bao gồm:



- **FPGA Cyclone IV EP4CE22E22C8N:** Chip FPGA mạnh mẽ có khả năng lập trình lại, cho phép bạn tạo ra các logic và chức năng tùy chỉnh.
- **Cổng GPIO (General Purpose Input/Output):** Cho phép kết nối với các thiết bị ngoại vi khác.
- **VGA:** Cho phép hiển thị hình ảnh trên màn hình.
- **USB UART:** Cổng USB UART cho phép giao tiếp với máy tính hoặc các thiết bị khác thông qua giao thức UART.
- **Âm thanh:** Cho phép phát và thu âm thanh qua cổng âm thanh.

Ngoài ra, DE10-Lite Board cũng hỗ trợ một loạt các phần mềm và công cụ phát triển, bao gồm Quartus II của Altera (Intel) để lập trình FPGA, và các công cụ hỗ trợ như hướng dẫn sử dụng và ví dụ ứng dụng để giúp người dùng bắt đầu nhanh chóng với việc phát triển ứng dụng trên nền tảng FPGA này.

## **5. Trình bày kế hoạch các chương tiếp theo**

### ***5.1 Kế hoạch chương 2***

- Trình bày việc thiết kế phần điều khiển của hệ truyền động bao gồm: Cấu trúc điều khiển (lưu đồ thuật toán của chương trình điều khiển), phương trình và các tham số chi tiết của từng khối trong cấu trúc điều khiển, các tham số của các thuật toán điều khiển.
- Sử dụng các thuật toán điều khiển PI cho các bộ điều khiển tốc độ, bộ điều khiển dòng điện: Nêu các phương pháp thiết kế chỉnh định để dò tìm các thông số của bộ điều khiển này.

### ***5.2 Kế hoạch chương 3***

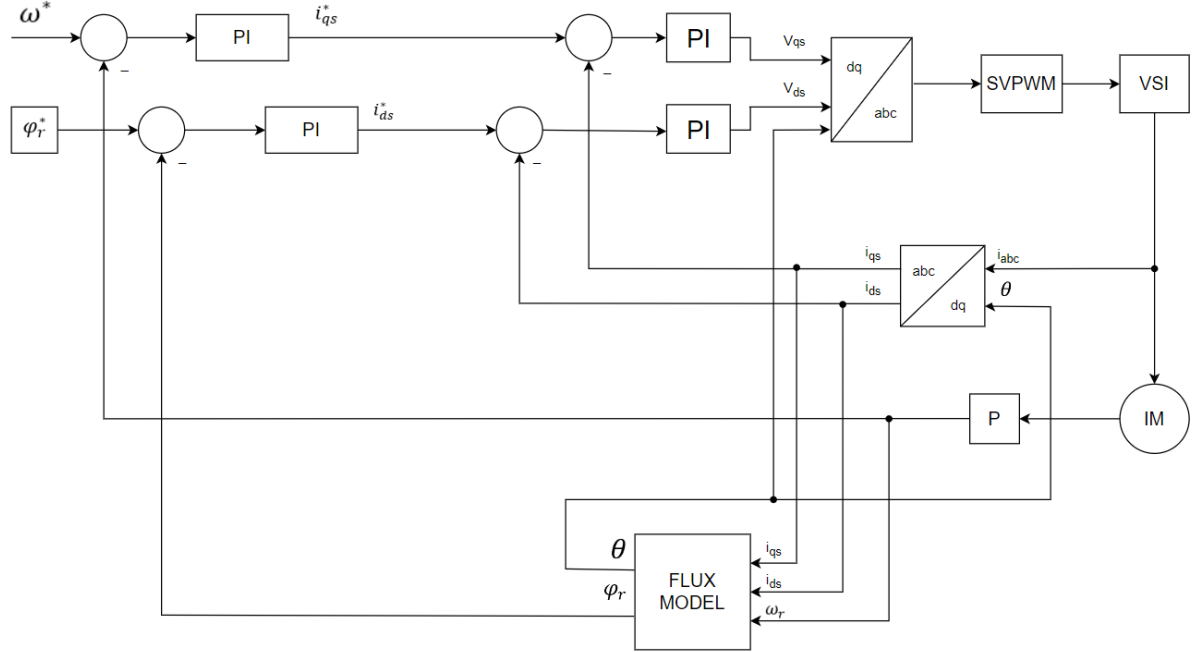
- Trình bày việc thiết kế, lựa chọn loại FPGA để thực hiện phần điều khiển của hệ truyền động điện.
- Trình bày và thuyết minh giải thích việc triển khai phần điều khiển đã tổng hợp của chương 2 sang trên FPGA.

### ***5.3 Kế hoạch chương 4***

- Trình bày việc mô phỏng kiểm nghiệm hệ thống truyền động điện trên phần mềm Matlab Simulink.
- Trình bày việc mô phỏng kiểm nghiệm hệ thống truyền động điện khi thực hiện phần điều khiển bằng FPGA.
- So sánh các kết quả mô phỏng.
- Kết luận

## CHƯƠNG 2: TỔNG HỢP PHẦN ĐIỀU KHIỂN

### 1. Mô hình tổng quát hệ thống



Hình 2.1: Mô hình tổng quát hệ thống

### 2. Xây dựng các phương trình toán học

#### 2.1 Mô hình không gian trạng thái của động cơ IM trong hệ tọa độ dq

Tập hợp các phương trình mô tả điện áp và từ thông của động cơ trên hệ tọa độ dq:

$$\begin{cases} u_s^f = R_s i_s^f + \frac{d\psi_s^f}{dt} + j \cdot \omega_s \psi_s^f \\ 0 = R_r i_r^f + \frac{d\psi_r^f}{dt} + j \omega_r \psi_r^f \\ \psi_s^f = L_s i_s^f + L_m i_r^f \\ \psi_r^f = L_m i_s^f + L_r i_r^f \end{cases}$$

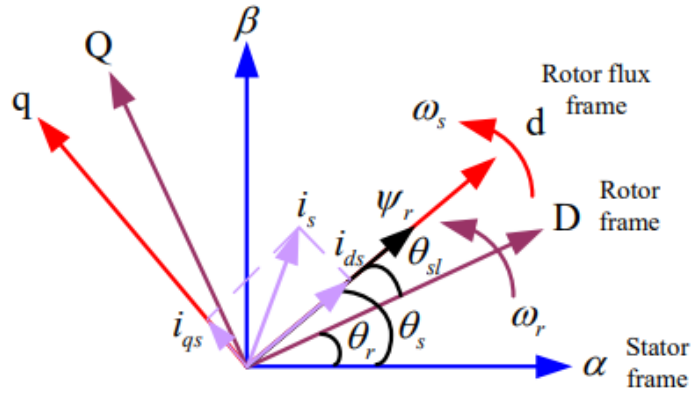
Ta tìm cách khử các đại lượng không quan trọng trong hệ đó là dòng (không đo được) của mạch rotor  $i_r^f$  và cả từ thông stator  $\psi_s^f$ . Từ 2 phương trình từ thông ta có:

$$i_r^f = \frac{1}{L_r} (\psi_r^f - L_m i_s^f)$$

$$\psi_s^f = L_s i_s^f + \frac{L_m}{L_r} (\psi_r^f - L_m i_s^f)$$

Thay kết quả  $i_r^f$  và  $\Psi_s^f$  vào 2 phương trình điện áp, sau đó chiếu xuống 2 trục d và q, chuyển các thành phần vi phân sang về trái, đồng thời chuyển sang viết dưới dạng các thành phần vector ta có hệ phương trình sau:

$$\begin{cases} \frac{di_{sd}}{dt} = -\left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)i_{sd} + \omega_s i_{sq} + \frac{1-\sigma}{\sigma T_r}\Psi'_{rd} + \frac{1-\sigma}{\sigma}\omega\Psi'_{rq} + \frac{1}{\sigma L_s}u_{sd} & (1) \\ \frac{di_{sq}}{dt} = -\omega_s i_{sd} - \left(\frac{1}{\sigma T_s} + \frac{1-\sigma}{\sigma T_r}\right)i_{sq} - \frac{1-\sigma}{\sigma}\omega\Psi'_{rd} + \frac{1-\sigma}{\sigma T_r}\Psi'_{rq} + \frac{1}{\sigma L_s}u_{sq} & (2) \\ \frac{d\Psi'_{rd}}{dt} = \frac{1}{T_r}i_{sd} - \frac{1}{T_r}\Psi'_{rd} + (\omega_s - \omega)\Psi'_{rq} & (3) \\ \frac{d\Psi'_{rq}}{dt} = \frac{1}{T_r}i_{sq} - (\omega_s - \omega)\Psi'_{rd} - \frac{1}{T_r}\Psi'_{rq} & (4) \end{cases}$$



Do trục d trùng với trục vector từ thông rotor  $\psi_r \rightarrow \psi_{rq} = 0$

$$\psi_{dr} = L_m \psi'_{rd} = \frac{L_m}{1 + sT_r} i_{ds}$$

Phương trình moment:

$$T_{em} = \frac{3}{2}p \frac{L_m}{L_r} (i_{qs}\psi_{dr} - i_{ds}\psi_{qr}) = \frac{3}{2}p \frac{L_m^2}{L_r} \psi'_{dr} i_{qs}$$

Phương trình chuyển động:

$$T_{em} - T_L = \frac{J}{z_p} \frac{d\omega}{dt} \Rightarrow \omega(s) = (T_{em} - T_L) \frac{z_p}{j.s}$$

Kết hợp cả 2 phương trình trên ta có:

$$\omega(s) = \left( \frac{3}{2} z_p \frac{L_m^2}{L_r} \Psi'_{rd} i_{sq} - T_L \right) \frac{z_p}{j.s}$$

Từ phương trình (2) và (4), ta có:

$$\psi_r (1 + T_r (s + j\omega_{sl})) = L_m i_s \quad (5)$$

Tách phương trình (5) thành hai phần trên hệ tọa độ dq với giả thiết tựa theo trục từ thông rotor:

$$\begin{aligned}\psi_{dr}(1 + sT_r) &= L_m i_{ds} \\ \omega_{sl} T_r \psi_{dr} &= L_m i_{qs}\end{aligned}$$

Tốc độ trượt:

$$\omega_{sl} = \frac{L_m}{\psi_{dr}} i_{qs}$$

Các tham số của động cơ:

- Điện trở stato  $R_s = 0.08233 \Omega$
- Điện trở rotor  $R_r = 0.0503 \Omega$
- Điện cảm tương hỗ giữa rotor và stator  $L_m = 0.02711 \text{ H}$
- Điện cảm dò phía stator  $L_{\sigma s} = 0.000724 \text{ H}$
- Điện cảm dò phía rotor  $L_{\sigma r} = 0.000724 \text{ H}$

Tính toán các đại lượng cần thiết cho mô phỏng:

- Điện cảm stator:  $L_s = L_m + L_{\sigma s} = 0.027834 \text{ H}$
- Điện cảm rotor:  $L_r = L_m + L_{\sigma r} = 0.027834 \text{ H}$
- Hằng số thời gian stator:  $T_s = \frac{L_s}{R_s} = 0.33808$
- Hằng số thời gian rotor:  $T_r = \frac{L_r}{R_r} = 0.55336$
- Hệ số từ tản toàn phần:  $\sigma = 1 - \frac{L_m^2}{L_r L_s} = 0.05135$

## 2.2 Mô hình toán học của khâu mô hình từ thông

Việc đo từ thông phản hồi trực tiếp từ động cơ rất phức tạp và khó thực hiện, vì vậy ở đây ta cần có khâu mô hình từ thông có nhiệm vụ tính từ thông rotor

$\Psi'_{rd}$  ( $\Psi'_{rd} = \frac{\psi_{rd}}{L_m}$ ) và góc quay  $\vartheta_s$  trên cơ sở các đại lượng đo được như  $i_{sd}, i_{sq}, \omega$ .

Để tính được góc quay  $\vartheta_s$  quy tụ lại ta cần phải tính tốc độ góc  $\omega_r$  của mạch điện rotor:

$$\vartheta_s = \int \omega_s dt$$

Với:  $\omega_s = \omega + \omega_r$

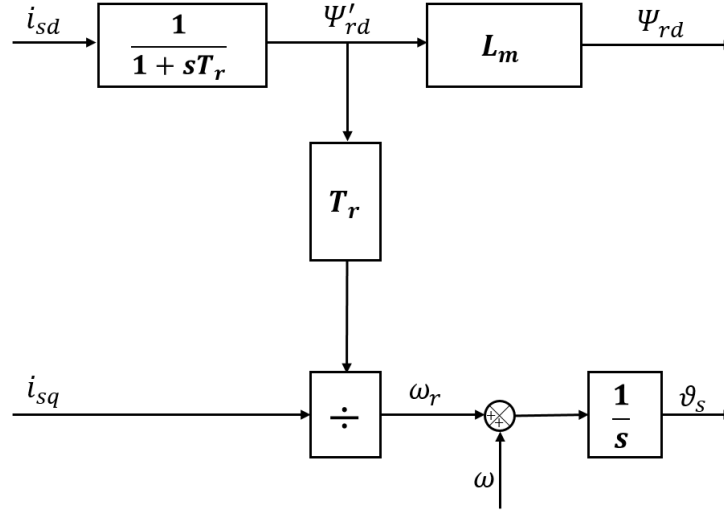
$$\omega_r = \frac{i_{sq}}{T_r \Psi'_{rd}}$$

$$\Rightarrow \vartheta_s(s) = \left( \omega + \frac{i_{sq}}{T_r \Psi'_{rd}} \right) \times \frac{1}{s}$$

Ta có công thức :  $\Psi'_{rd} = \frac{i_{sd}}{1+sT_r}$

Từ đó suy ra:

$$\vartheta_s(s) = \left[ \omega + \left( \frac{i_{sq}}{T_r} \div \frac{i_{sd}}{1+sT_r} \right) \right] \times \frac{1}{s}$$



Hình 2.2: Mô hình toán học của khâu mô hình từ thông

### 2.3 Mô hình toán học của khâu chuyển đổi hệ tọa độ $uvw \rightarrow \alpha\beta$

Giả sử coi ba dòng điện hình *sin* phía stator của động cơ xoay chiều ba pha không đồng bộ là đối xứng ta có công thức:

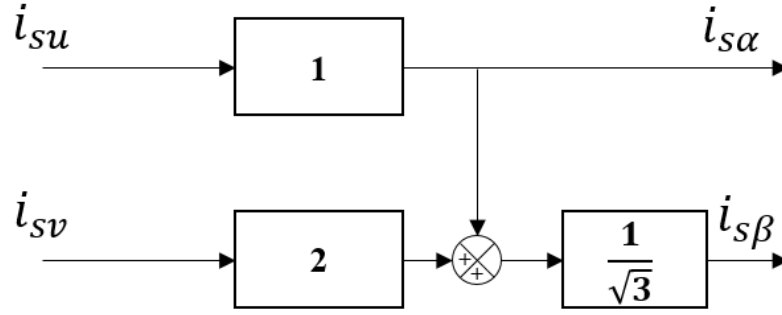
$$i_{su}(t) + i_{sv}(t) + i_{sw}(t) = 0$$

$\Rightarrow$  Lúc này ta chỉ cần đo 2 trong ba pha của dòng điện thì có thể suy ra dòng điện pha còn lại.

Ta có công thức chuyển đổi Clarke:

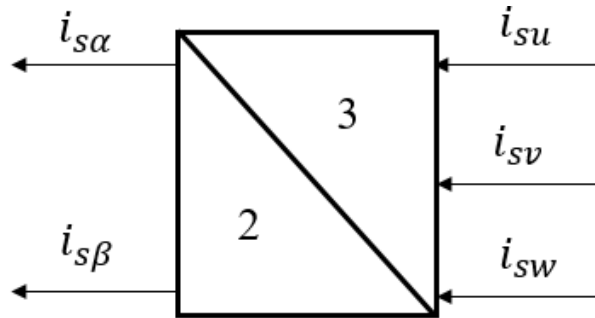
$$\begin{cases} i_{s\alpha} = i_{su} \\ i_{s\beta} = \frac{1}{\sqrt{3}}(i_{su} + 2i_{sv}) \end{cases}$$

Từ đó ta có được:



Hình 2.3: Mô hình toán học của bộ biến đổi  $uvw \rightarrow \alpha\beta$

Ta cũng có thể biểu diễn đơn giản bộ biến đổi  $uvw \rightarrow \alpha\beta$  dưới dạng:



Hình 2.4: Mô hình đơn giản hóa của bộ biến đổi  $uvw \rightarrow \alpha\beta$

#### 2.4 Mô hình toán học của khâu chuyển đổi $\alpha\beta \rightarrow dq$

Vector dòng điện  $i_s$  có thể được viết ở trong hai hệ tọa độ  $\alpha\beta$  và  $dq$  như sau:

$$i_s^s = i_{s\alpha} + j \cdot i_{s\beta}$$

$$i_s^f = i_{sd} + j \cdot i_{sq}$$

Với chỉ số viết bên phải trên cao:

s là chỉ hệ tọa độ stator cố định ( $\alpha\beta$ )

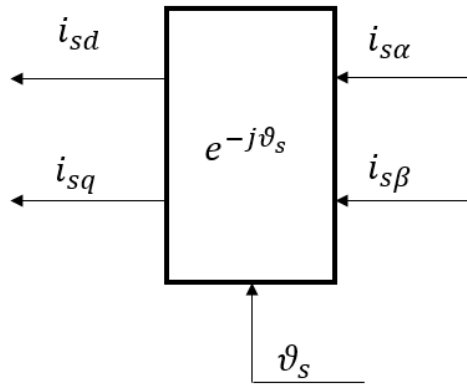
f là chỉ hệ tọa độ từ thông (dq)

Ta có ma trận chuyển đổi Park:

$$\begin{bmatrix} i_{sd} \\ i_{sq} \end{bmatrix} = \begin{bmatrix} \cos \vartheta_s & \sin \vartheta_s \\ -\sin \vartheta_s & \cos \vartheta_s \end{bmatrix} \begin{bmatrix} i_{s\alpha} \\ i_{s\beta} \end{bmatrix}$$

Từ đó ta suy ra được:

$$\begin{aligned} i_s^f &= [i_{s\alpha} \cos \vartheta_s + i_{s\beta} \sin \vartheta_s] + j[i_{s\beta} \cos \vartheta_s - i_{s\alpha} \sin \vartheta_s] \\ &= [i_{s\alpha} + j \cdot i_{s\beta}][\cos \vartheta_s - j \sin \vartheta_s] = i_s^s \cdot e^{-j\vartheta_s} \end{aligned}$$

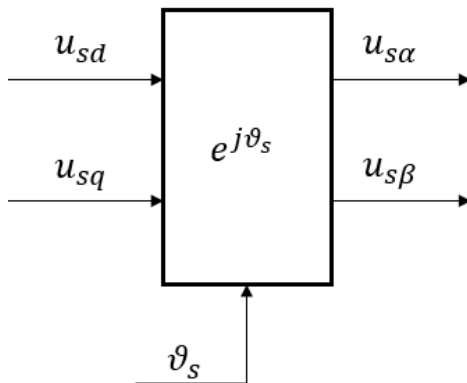


Hình 2.5: Mô hình toán học của bộ biến đổi  $\alpha\beta \rightarrow dq$ :

Ta có ma trận chuyển đổi Park ngược đối với điện áp:

$$\begin{bmatrix} u_{s\alpha} \\ u_{s\beta} \end{bmatrix} = \begin{bmatrix} \cos \vartheta_s & -\sin \vartheta_s \\ \sin \vartheta_s & \cos \vartheta_s \end{bmatrix} \begin{bmatrix} u_{sd} \\ u_{sq} \end{bmatrix}$$

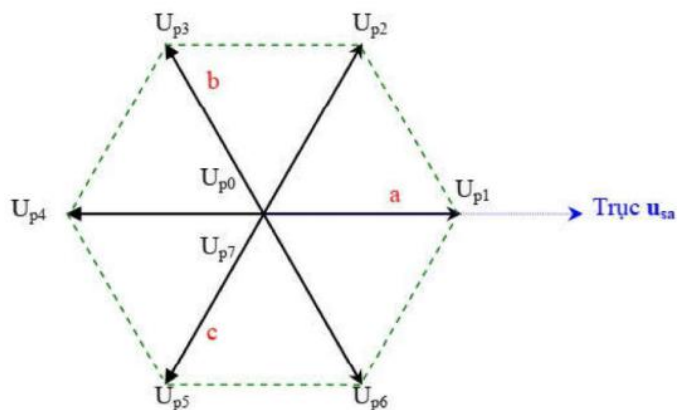
Ta cần chuyển đổi điện áp từ hệ tọa độ  $\alpha\beta$  để có thể áp dụng thuật toán điều chế vector không gian từ đó tính ra điện áp  $u_s(t)$  đặt vào động cơ.



Hình 2.6: Mô hình toán học của bộ biến đổi  $dq \rightarrow \alpha\beta$

## 2.5 Phương pháp SVPWM

Phương pháp sử dụng: Điều chế vector không gian SVPWM



Hình 2.7: Phân bố trạng thái valse và các vector điện áp

Ta có bộ nghịch lưu gồm 6 van  $S_1, S_2, S_3, S_4, S_5, S_6$  đóng mở theo phương pháp đối nghịch, lấy trạng thái  $S_1, S_3, S_5$  làm chuẩn đối với 3 pha  $a, b, c$  và  $S_4, S_6, S_2$  có trạng thái ngược lại.

Mỗi valve có 2 trạng thái mà ta có 3 valve  $\rightarrow$  có  $2^3=8$  trạng thái

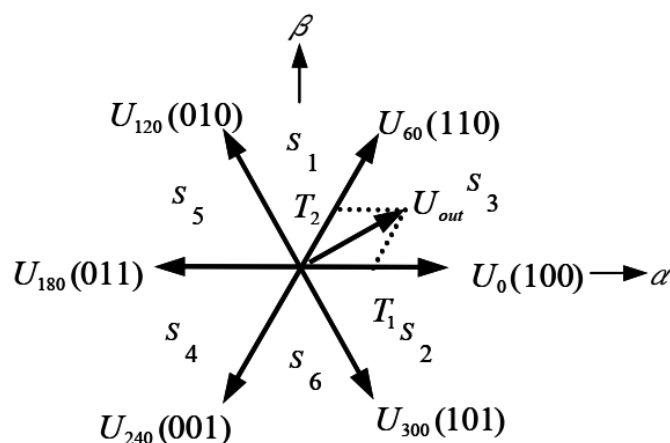
Sáu trong số các cấu hình này tương ứng với các điện áp khác nhau được áp dụng cho động cơ được kết nối và được gọi là *vector cơ bản* kể từ bây giờ. Hai cái cuối cùng được gọi là *vector không* vì chúng đại diện cho 0 Volt trên các cực.

Ta có bảng trạng thái và điện áp tương ứng :

a	b	c	$V_{AN}$	$V_{BN}$	$V_{CN}$	$V_{AB}$	$V_{BC}$	$V_{CA}$
0	0	0	0	0	0	0	0	0
1	0	0	$\frac{2V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$V_{DC}$	0	$-V_{DC}$
1	1	0	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$-\frac{2V_{DC}}{3}$	0	$V_{DC}$	$-V_{DC}$
0	1	0	$-\frac{V_{DC}}{3}$	$\frac{2V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$-V_{DC}$	$V_{DC}$	0
0	1	1	$-\frac{2V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$-V_{DC}$	0	$V_{DC}$
0	0	1	$-\frac{V_{DC}}{3}$	$-\frac{V_{DC}}{3}$	$\frac{2V_{DC}}{3}$	0	$-V_{DC}$	$V_{DC}$
1	0	1	$\frac{V_{DC}}{3}$	$-\frac{2V_{DC}}{3}$	$\frac{V_{DC}}{3}$	$V_{DC}$	$-V_{DC}$	0
1	1	1	0	0	0	0	0	0

Bảng 2.1: Phân bố trạng thái valve và các vector điện áp

Và phân bố trạng thái valve trên các điện áp tương ứng:



Hình 2.8: Phân bố trạng thái valve trên các vector điện áp

Vector điện áp đầu ra được xây dựng trên 2 vector điện áp liền kề. Thời gian đóng mở của 2 vector sẽ tạo ra vector điện áp đầu ra.



$$U_{out} = \frac{T_1}{T} U_X + \frac{T_2}{T} U_{X+60} + \frac{T_0}{T} (U_{000} \text{ or } U_{111})$$

$$T_0 = T - T_1 - T_2$$

Từ đây bài toán quy về tìm T1, T2 để có thể kích xung tạo ra điện áp U<sub>out</sub>

Các bước điều chế ra xung để kích ra các van công suất :

- Từ  $u_\alpha, u_\beta$  ta chuyển đổi chuyển đổi clark ngược

$$\begin{bmatrix} V_{ref1} \\ V_{ref2} \\ V_{ref3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ -\frac{1}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} v_\beta \\ v_\alpha \end{bmatrix} \quad (11)$$

- Sau đó so sánh:

Nếu  $V_{ref1} > 0$  thì a=1 nếu không a=0.

Nếu  $V_{ref2} > 0$  thì b=1 nếu không b=0.

Nếu  $V_{ref3} > 0$  thì c=1 nếu không c=0.

Từ đó tính ra được Sector mà điện áp U<sub>s</sub> đang ở theo công thức:

$$\text{Section no} = a + 2b + 4c \quad (12)$$

Từ đó ta tính T<sub>x</sub>, T<sub>y</sub>, T<sub>z</sub> theo công thức:

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \frac{\sqrt{3}T}{V_{DC}} \begin{bmatrix} V_{ref1} \\ -V_{ref2} \\ -V_{ref3} \end{bmatrix} \quad (13)$$

- Sau đó ta sẽ tính được giá trị T1, T2 theo bảng

	S1	S2	S3	S4	S5	S6
$T_1$	$T_z$	$T_y$	$-T_z$	$-T_x$	$T_x$	$-T_y$
$T_2$	$T_y$	$-T_x$	$T_x$	$T_z$	$-T_y$	$-T_z$

Bảng 2.2: Giá trị T<sub>1</sub>, T<sub>2</sub>

Tuy nhiên nếu T<sub>1</sub> + T<sub>2</sub> > T thì ta phải giới hạn T<sub>1</sub>, T<sub>2</sub>

$$T_{1(sat)} = \frac{T_1 T}{T_1 + T_2}$$

$$T_{2(sat)} = \frac{T_2 T}{T_1 + T_2}$$

Từ T<sub>1</sub>, T<sub>2</sub> tính ra ta tính được thời gian mở trên mỗi valse S<sub>1</sub>, S<sub>3</sub>, S<sub>5</sub>, S<sub>4</sub>, S<sub>6</sub>, S<sub>2</sub>:

$$T_{aon} = \frac{T - T_1 - T_2}{2} = \frac{T_0}{2} \quad (14)$$

$$T_{bon} = T_{aon} + T_1 \quad (15)$$

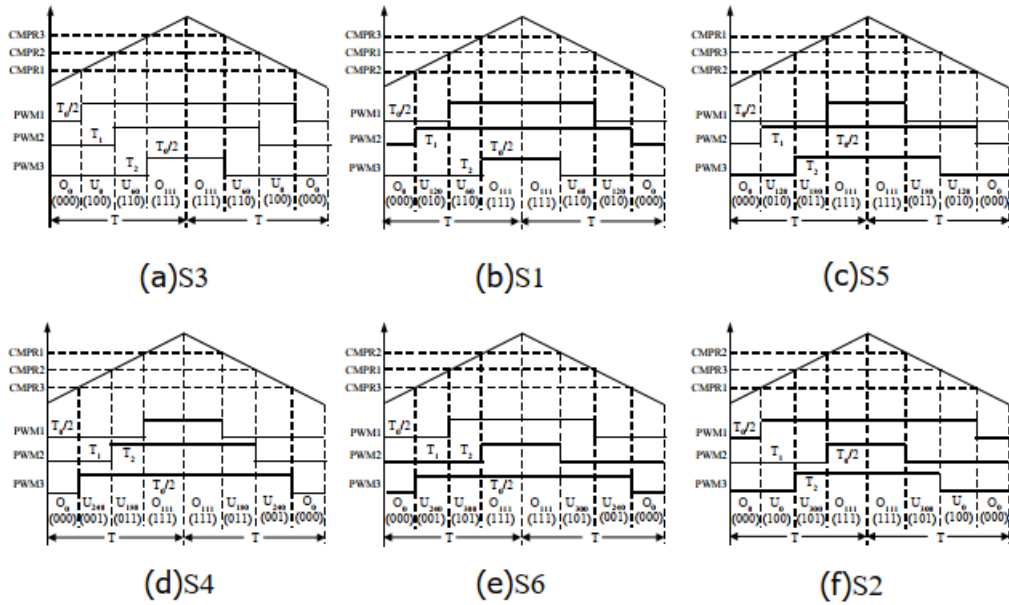
$$T_{con} = T_{bon} + T_2 \quad (16)$$

Từ các giá trị  $T_{aon}$ ,  $T_{bon}$ ,  $T_{con}$  ta tính ra được các giá trị  $CMPR1$ ,  $CMPR2$ ,  $CMPR3$  theo bảng sau :

Section	0°~60° S3	60°~120° S1	120°~180° S5	180°~240° S4	240°~300° S6	300°~360° S2
CMPR1	$T_{aon}$	$T_{bon}$	$T_{con}$	$T_{con}$	$T_{bon}$	$T_{aon}$
CMPR2	$T_{bon}$	$T_{aon}$	$T_{aon}$	$T_{bon}$	$T_{con}$	$T_{con}$
CMPR3	$T_{con}$	$T_{con}$	$T_{bon}$	$T_{aon}$	$T_{aon}$	$T_{bon}$

Bảng 2.3: Giá trị  $CMPR1$ ,  $CMPR2$ ,  $CMPR3$

Từ đây ta so sánh với 1 xung răng cưa và xuất ra tín hiệu xung để điều khiển các valve



Hình 2.9

Xây dựng thuật toán tổng quát:

Bước 1. Từ công thức (11) tính  $V_{ref1}$ ,  $V_{ref2}$ ,  $V_{ref3}$  và thứ tự sector.

Bước 2. Tính thời gian  $T_x$ ,  $T_y$ ,  $T_z$ .

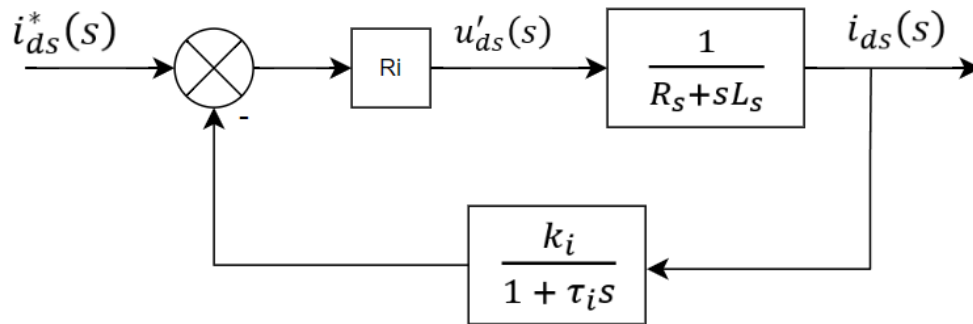
Bước 3. Tính thời gian  $T_1$ ,  $T_2$  cho mỗi sector theo bảng 2.

Bước 4. Tính thời gian  $T_{aon}$ ,  $T_{bon}$ ,  $T_{con}$  theo công thức (14)~(16).

Bước 5. Tính  $CMPR1$ ,  $CMPR2$ ,  $CMPR3$  theo (Bảng 2.3) và so sánh với xung tam giác mẫu giá trị  $[0 \dots T]$  với chu kỳ  $T$ .

### 3. Thiết kế bộ điều khiển

#### 3.1 Thiết kế bộ điều khiển dòng điện



Hình 2.10: Bộ điều khiển dòng điện

Hàm truyền của khâu đo dòng điện được thể hiện dưới dạng hàm truyền là  $\frac{k_i}{1 + \tau_i s}$

Trong đó:

$K_i$  là hệ số khuếch đại cơ cấu đo

$T_i$  là hằng số thời gian mạch lọc cảm biến dòng.

Ta nhận thấy đối tượng thực của động cơ không đồng bộ ba pha rotor lồng sóc trong hệ truyền động là cuộn dây stator, nên khi thể hiện với dạng hàm truyền, động cơ chỉ còn dưới dạng là:  $\frac{1}{R_s + sL_s}$ , trong đó  $L_s$  và  $R_s$  lần lượt là hằng số điện từ và điện trở của cuộn stator

Bộ điều khiển ở trên được chọn thông thường là bộ điều khiển PI có dạng là:

$$R_i(s) = K_i \left( 1 + \frac{1}{sT_i} \right)$$

Khi đó hàm truyền hở có dạng là:

$$G_0 = K_i \left( 1 + \frac{1}{sT_i} \right) \frac{k_i}{1 + \tau_i s} \frac{1}{R_s + sL_s} = \frac{K_i k_i}{T_i R_s} \frac{(1 + sT_i)}{s(1 + sT_s)(1 + \tau_i s)}$$

Đặt

$$K = \frac{K_i k_i}{T_i R_s}$$

Chọn  $T_i = T_s = 0.3381$ . Ta rút gọn được hàm truyền thành dạng:

$$G_0 = \frac{K}{s(1 + s\tau_i)}$$

Hàm truyền hệ kín:

$$G_k = \frac{K}{s(1 + s\tau_i) + K} = \frac{K}{\tau_i s^2 + s + K}$$

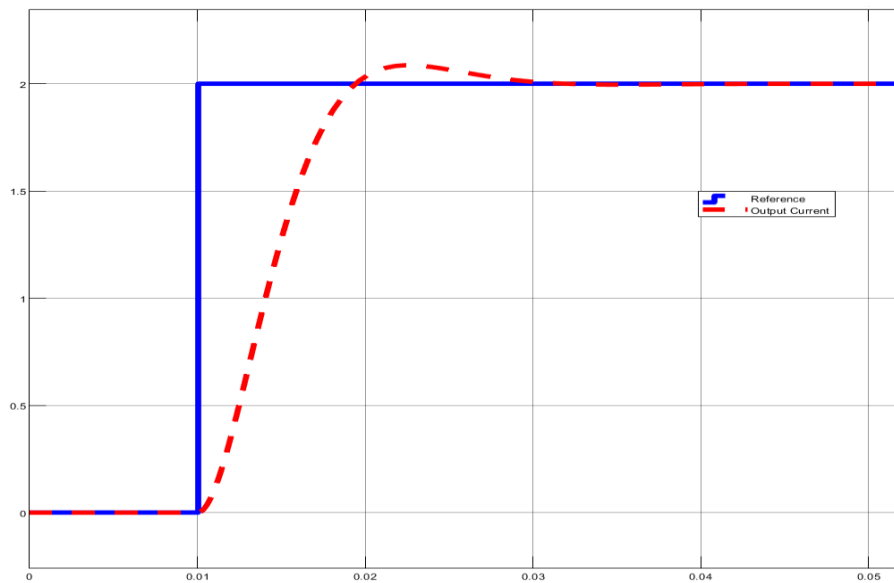
Tổng hợp theo tiêu chuẩn module tối ưu cho hệ bậc hai: Hàm truyền kín phải có dạng:

$$G_k = \frac{1}{1 + 2\tau_i s + 2\tau_i^2 s^2}$$

Đồng nhất hệ số:

$$2\tau_i^2 = \frac{T_i}{K} \rightarrow K_1 = \frac{R_s T_i}{2K_i \tau_i} = 3.5677$$

$$\rightarrow \begin{cases} T_i = 0.3381 \\ K_i = 2.9737 \end{cases}$$



Hình 2.11: Kết quả mô phỏng dòng điện

### Xấp xỉ mạch vòng dòng điện.

Từ hàm truyền đã được rút gọn:

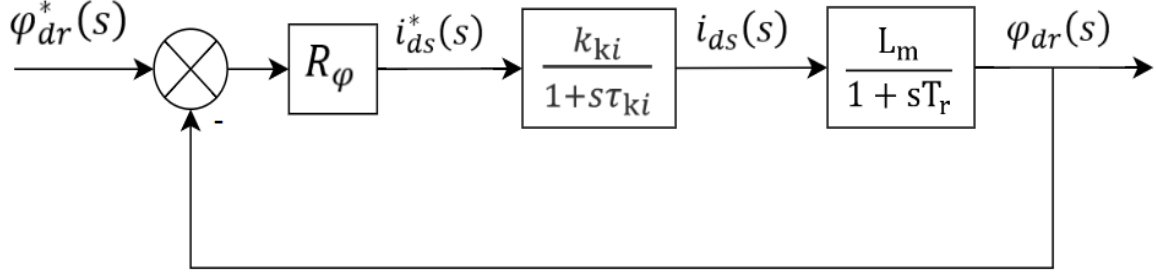
$$G_0 = \frac{K}{s(1 + s\tau_i)}$$

Hàm kín trở thành :

$$G_k = \frac{G_0}{G_0 + 1} = \frac{K}{1 + K + s\tau_i} = \frac{k_{ki}}{1 + s\tau_{ki}}$$

Trong quá trình mô phỏng, các hệ số  $k_p, k_i$  được tính chọn sao cho kết quả đáp ứng được tốt nhất.

### 3.2 Thiết kế bộ điều khiển từ thông.



Hình 2.12: Bộ điều khiển từ thông

Bộ điều khiển ở trên được chọn thông thường là bộ điều khiển PI có dạng là

$$R_\varphi(s) = K_\varphi \left( 1 + \frac{1}{sT_\varphi} \right)$$

Khi đó hàm truyền hở có dạng là:

$$G_0 = K_\varphi \left( 1 + \frac{1}{sT_\varphi} \right) \frac{k_{ki}}{1+s\tau_{ki}} \frac{L_m}{1+sT_r} = \frac{K_\varphi k_{ki} L_m}{T_\varphi} \frac{(1+sT_\varphi)}{s(1+s\tau_{ki})(1+sT_r)}$$

Đặt: 
$$K' = \frac{K_\varphi k_{ki} L_m}{T_\varphi}$$

Chọn  $T_\varphi = T_r = 0.3381$  để khử được hằng số thời gian lớn nhất của mẫu số trong hàm truyền hở của hệ. Ta rút gọn được hàm truyền thành dạng:

$$G_0 = \frac{K'}{s(1+s\tau_{ki})}$$

Hàm truyền hệ kín:

$$G_k = \frac{K'}{s(1+s\tau_{ki}) + K'} = \frac{K'}{\tau_{ki}s^2 + s + K'}$$

Tổng hợp theo tiêu chuẩn module tối ưu cho hệ bậc hai: Hàm truyền kín phải có dạng:

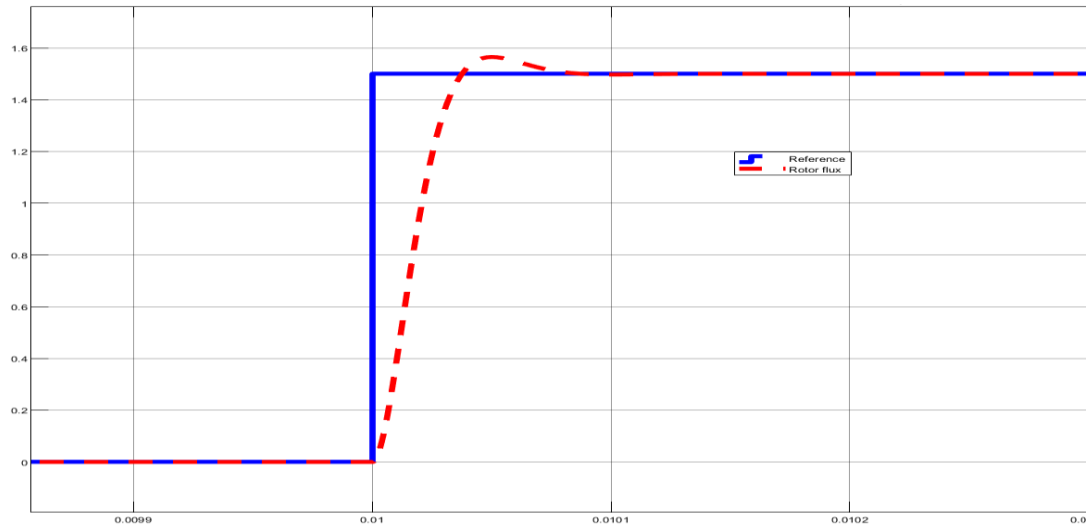
$$G_k = \frac{1}{1 + 2\tau_{ki}s + 2\tau_{ki}^2s^2}$$

Đồng nhất hệ số:

$$2T_i^2 = \frac{\tau_{ki}}{K'} \rightarrow K_\varphi = \frac{T_\varphi}{2\tau_{ki}k_{ki}L_m} = 3.5677$$

Ta có được :

$$\begin{cases} T_{\varphi} = 0.5534 \\ K_{\varphi} = 1.02e + 4 \end{cases}$$



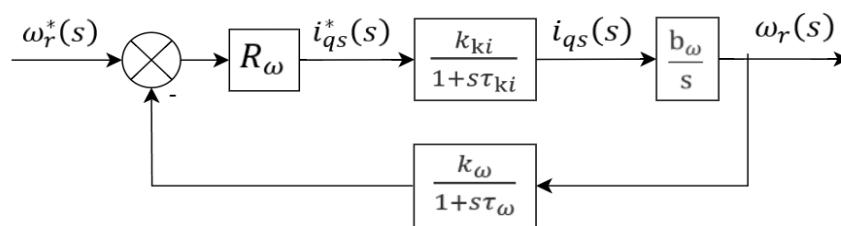
Hình 2.13: Kết quả mô phỏng bộ điều khiển từ thông

### 3.3 Thiết kế bộ điều khiển tốc độ

Ta có:  $\frac{d\omega_r}{dt} = \frac{1}{J} (T_{em} - T_L - B\omega_r) = \frac{3p^2 L_m \psi_{dr}}{2L_r} i_{qs} - \frac{p}{J} T_L$

$$\rightarrow G(s) = \frac{\omega_r(s)}{I_{qs}(s)} = \frac{\frac{3p^2 L_m \psi_{dr}}{2L_r}}{s} = \frac{b_{\omega}}{s}$$

Ta có cấu trúc điều khiển tốc độ động cơ như sau:



Hình 2.14: Bộ điều khiển tốc độ

Bộ điều khiển tốc độ được chọn là bộ điều khiển PI có dạng

$$R_{\omega}(s) = K_{\omega} \left( 1 + \frac{1}{sT_{\omega}} \right)$$

Hàm truyền hở của mạch vòng điều khiển tốc độ có dạng:

$$G_0 = K_{\omega} \left( 1 + \frac{1}{sT_{\omega}} \right) \frac{b_{\omega}}{s} \frac{k_{ki}}{1+s\tau_{ki}} \frac{k_{\omega}}{1+s\tau_{\omega}} = \frac{K_{\omega} k_{ki} k_{\omega} b_{\omega}}{T_{\omega}} \frac{(1+sT_{\omega})}{s^2 (1+s\tau_{ki})(1+s\tau_{\omega})}$$

Xấp xỉ hàm truyền hệ hở viết lại như sau:

$$G_o = \frac{K_\omega k_{s\omega} b_\omega}{T_\omega} \frac{(1 + sT_\omega)}{s^2(1 + s\tau_{s\omega})}$$

$$k_{s\omega} = k_{ki} k_\omega$$

$$\tau_{s\omega} = \tau_{ki} + \tau_\omega$$

Hàm truyền hệ kín:

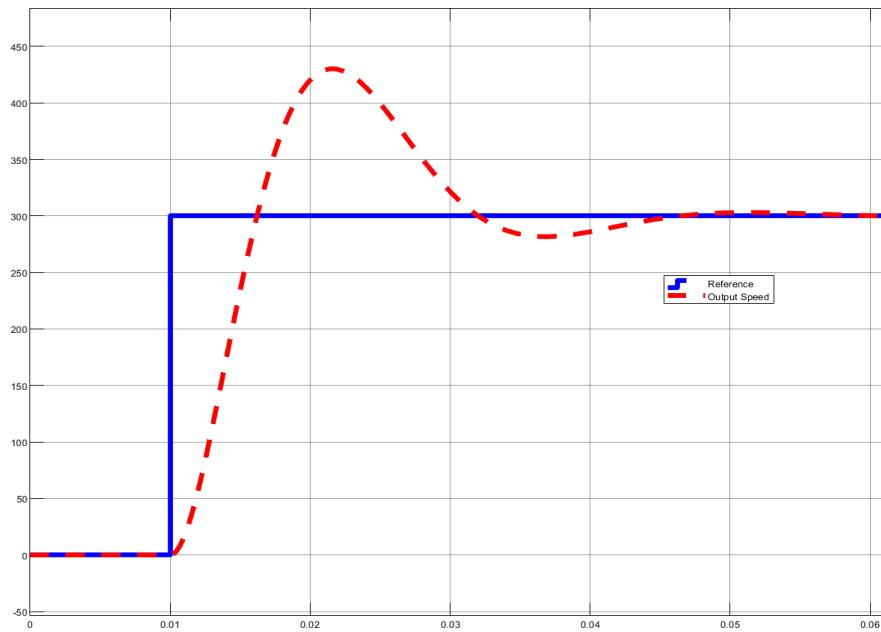
$$G_{kw} = \frac{K''(1 + sT_\omega)}{\tau_{s\omega}s^3 + s^2 + K''\tau_{s\omega}s + K''}$$

Tổng hợp theo module tối ưu đối xứng: Hàm truyền hệ kín có dạng như sau:

$$G_k = \frac{1 + 4\tau_{s\omega}s}{1 + 4\tau_{s\omega}s + 8\tau_{s\omega}^2s^2 + 8\tau_{s\omega}^3s^3}$$

Đồng nhất hệ số:

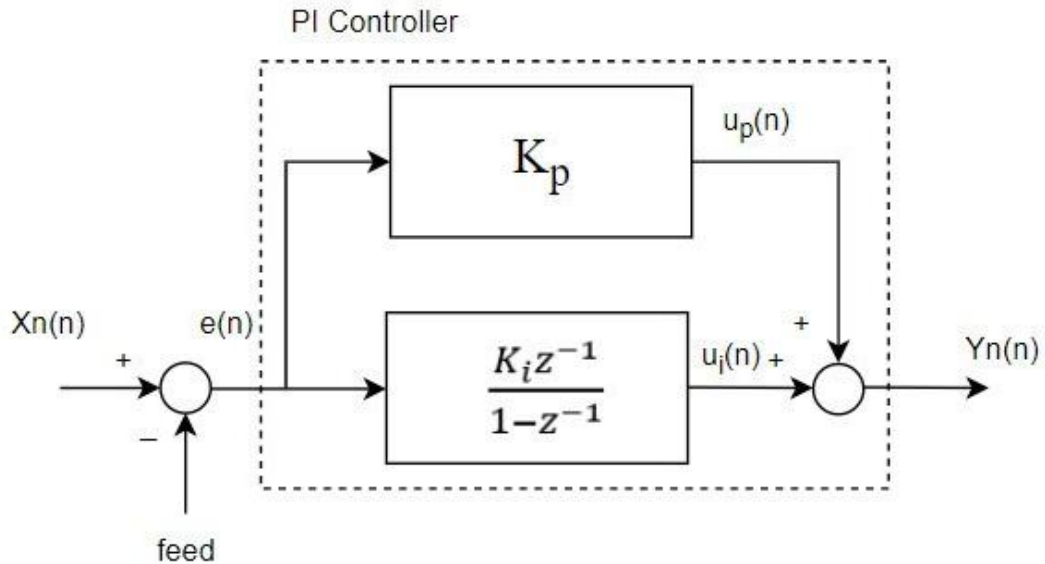
$$\begin{cases} T_\omega = 4\tau_{s\omega} = 0.008 \\ K_\omega = \frac{1}{2\tau_{s\omega}k_{s\omega}b_\omega} = 31.6568 \end{cases}$$



Hình 2.15: Kết quả mô phỏng bộ điều khiển tốc độ

## CHƯƠNG 3: THIẾT KẾ, TRIỂN KHAI FPGA

### 1. Lập trình bộ điều khiển PI



Hình 3.1: Sơ đồ khối bộ điều khiển PI

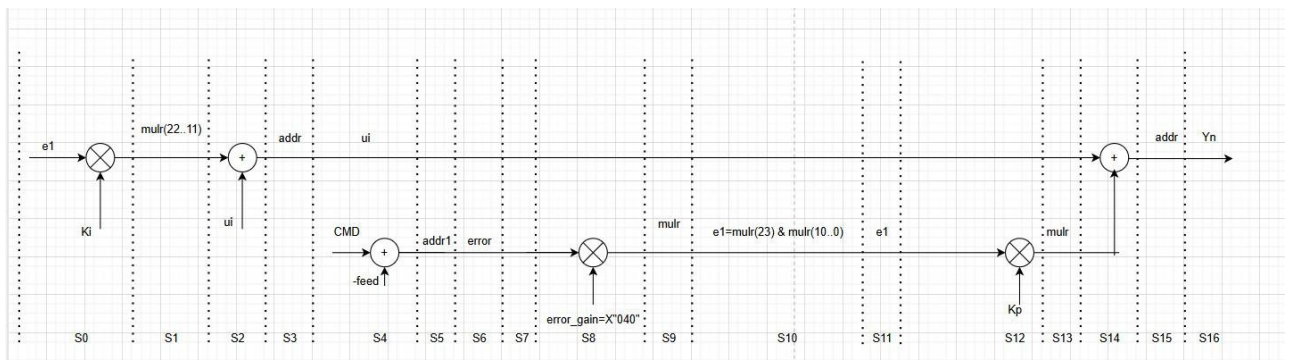
#### 1.1 Công thức của bộ PI:

$$u_p(n) = K_p e(n)$$

$$u_i(n) = u_i(n-1) + K_i e(n-1)$$

$$Y(n) = u_i(n-1) + K_p e(n) + K_i e(n-1)$$

#### 1.2 Lưu đồ thuật toán



Hình 3.2: Lưu đồ thuật toán bộ PI

#### 1.3 Code lập trình bộ điều khiển PI

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
```



```
USE IEEE.std_logic_arith.all;
USE IEEE.std_logic_signed.all;
LIBRARY lpm;
USE lpm.LPM_COMPONENTS.ALL;
ENTITY PI_speed IS
port(
    CLK:IN STD_LOGIC:='0';
    CLK40:IN STD_LOGIC:='0';
    CMD,Kp,Ki:IN STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
    Feed :IN STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
    daura:out STD_LOGIC_VECTOR(11 downto 0):=(others=>'0')
);
END PI_speed;
ARCHITECTURE PI_arch OF PI_speed IS
SIGNAL Info_speed : STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
SIGNAL e1,Un, error, error1, error_gain, error_x_gain :
STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
SIGNAL mula1,mulb1 : STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
SIGNAL mulr1 : STD_LOGIC_VECTOR(23 downto 0):=(others=>'0');
SIGNAL adda1,addb1,addr1,addr : STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
SIGNAL ui : STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
SIGNAL CNT: STD_LOGIC_VECTOR(11 downto 0):=(others=>'0');
SIGNAL overflow, cout : STD_LOGIC;

BEGIN
mull: lpm_mult
generic map(LPM_WIDTHA=>12,LPM_WIDTHB=>12,LPM_WIDTHS=>12,LPM_WIDTHHP=>24,
            LPM_REPRESENTATION=>"SIGNED",LPM_PIPELINE=>1)
port map(dataa=> mula1,datab=> mulb1,clock=> clk,result=> mulr1);
adder1: lpm_add_sub
generic map(lpm_width=>12,LPM_REPRESENTATION=>"SIGNED",lpm_pipeline=>1)
port map(dataa=>adda1,datab=>addb1,clock=> clk,result=>addr, overflow => overflow,
cout => cout);

error_gain <= X"040"; --64
process (clk,addr)
```

```
BEGIN
  if clk'event and clk='1' then
    if overflow='1' then
      if cout='1' then
        addr1 <= X"800";
      else
        addr1 <= X"7FF";
      end if;
    else
      addr1 <= addr;
    end if;
  end if;
END PROCESS;

GEN:block
BEGIN
PROCESS(CLK)
BEGIN
  IF CLK40'EVENT and CLK40='1' THEN
    CNT<=CNT+1;
    IF CNT=X"000" THEN
      mula1<= e1;
      mulb1<= Ki;
    ELSIF CNT=X"002" THEN
      adda1<= mulr1(22 downto 11);
      addb1<= ui;
    ELSIF CNT=X"004" THEN
      ui<= addr1;
      adda1<= CMD;
      addb1<= -feed;
    ELSIF CNT=X"006" THEN
      error<= addr1;
    ELSIF CNT=X"008" THEN
      mula1<= error_gain;
      mulb1<= error; -- error
    ELSIF CNT=X"00A" THEN
```

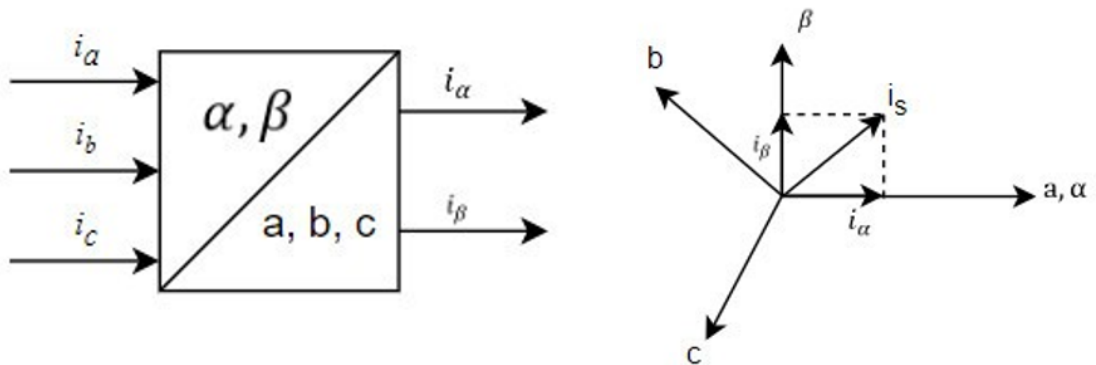
```

        error_x_gain <= mulr1(23) & mulr1(10 downto 0);
ELSIF CNT=X"00C" THEN
    e1<= error_x_gain;
    mula1<= error_x_gain;
    mulb1<= Kp;
ELSIF CNT=X"00E" THEN
    adda1<= mulr1(22 downto 11);
    addb1<= ui;
ELSIF CNT=X"010" THEN
    daura <= addr1;
    CNT<= X"000";

END IF;
END IF;
END PROCESS;
END BLOCK GEN;
END PI_arch;
sửa code PI

```

## 2. Lập trình khối biến đổi clark thuận

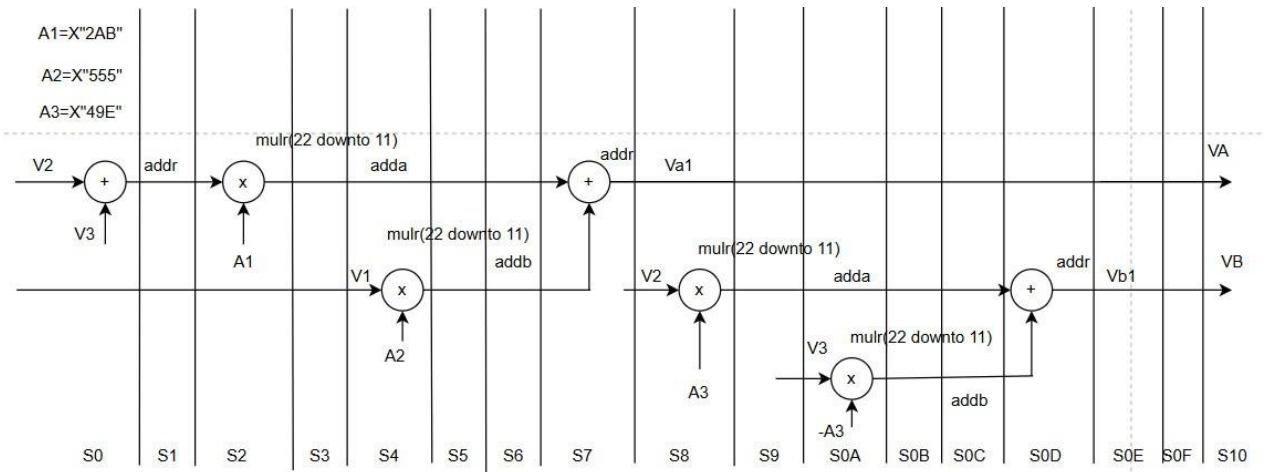


### 2.1 Công thức biến đổi clark thuận:

Công thức clark thuận

$$\begin{bmatrix} i_{\alpha} \\ i_{\beta} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{-1}{3} & \frac{-1}{3} \\ 0 & \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

## 2.2 Lưu đồ thuật toán:



Hình 3.3: Lưu đồ thuật toán Clark thuận

## 2.3. Code mô phỏng bộ biến đổi clark thuận:

```

LIBRARY IEEE;
USE IEEE.std_logic_1164.all;
USE IEEE.std_logic_arith.all;
USE IEEE.std_logic_signed.all;
use IEEE.numeric_std.all;

LIBRARY lpm;
USE lpm.LPM_COMPONENTS.ALL;

ENTITY CLARK is
port (
CLK,clk40 : IN STD_LOGIC:='0';
V1,V2,V3: IN STD_LOGIC_VECTOR(11 DOWNTO 0):=(OTHERS =>'0');
Va,Vb : OUT STD_LOGIC_VECTOR(11 DOWNTO 0):=(OTHERS =>'0');
end entity;

architecture a of CLARK is
signal A1 : std_logic_vector(11 downto 0) :=X"555";--2/3*2048
signal A2 : std_logic_vector(11 downto 0) :=X"2AB";-- 2/3*1/2*2048
signal A3 : std_logic_vector(11 downto 0) :=X"49E"; --SQRT(3)/2*2/3*2048

signal Adda, Addb, Addr, Mula, Mulb,Va1,Vb1 :STD_LOGIC_VECTOR (11 downto 0):=(others =>'0');
signal mulr :std_logic_vector( 23 downto 0):=(others =>'0');
signal CNT : STD_LOGIC_VECTOR (7 DOWNTO 0):=(others =>'0');
BEGIN

```

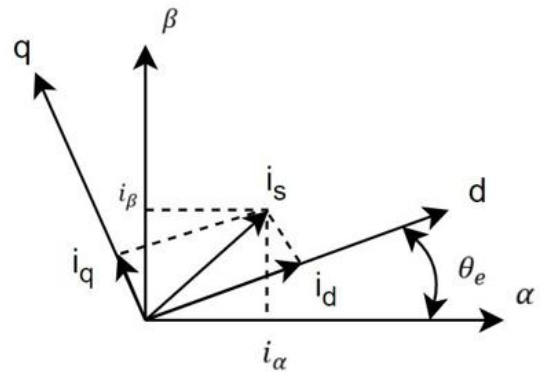
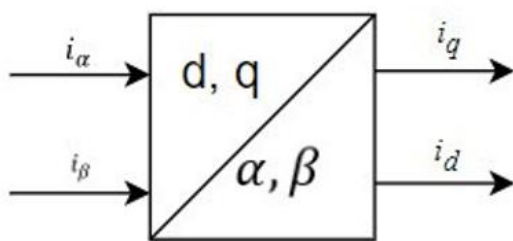
```
mull: lpm_mult
generic
map(LPM_WIDTHA=>12,LPM_WIDTHB=>12,LPM_WIDTHS=>12,LPM_WIDTHP=>24,LPM_REPRESENTATION
=>"SIGNED",LPM_PIPELINE=>1)
port map(dataa=> mula,datab=> mulb,clock=> clk,result=> mulr);
adder1: lpm_add_sub
generic map(lpm_width=>12,LPM_REPRESENTATION=>"SIGNED",lpm_pipeline=>1)
port map(dataa=>adda,datab=>addb,clock=> clk,result=>addr);
GEN: block
begin
process (CLK40)
begin
if clk40'event and clk40 ='1' then
    cnt <= cnt+1;
    if cnt<=x"00" then
        adda<= V3;
        addb<= V2;
    elsif cnt<= x"02" then
        mula<= addr;
        mulb<= -A2;
    elsif cnt<= x"04" then
        adda<= mulr(22 downto 11);
        mula<= V1;
        mulb <= A1;
    elsif cnt<= x"06" then
        addb<=mulr(22 downto 11);
    elsif cnt<= x"08" then
        Va1<=addr;
        mula<= V2;
        MULB<= A3;
    elsif cnt<= x"0A" then
        adda<=mulr(22 downto 11);
        mula<= V3;
        mulb<=-a3;
    elsif cnt<= x"0C" then
        addb<=mulr(22 downto 11);
```

```

elseif cnt<= x"0E" then
    Vb1<=addr;
elseif cnt<= x"10" then
    Va<=Va1;
    Vb<=Vb1;
    CNT<=X"00";
END IF;
END IF;
end process;
end block GEN;
END ARCHITECTURE;

```

### 3. Code biến đổi Park thuận

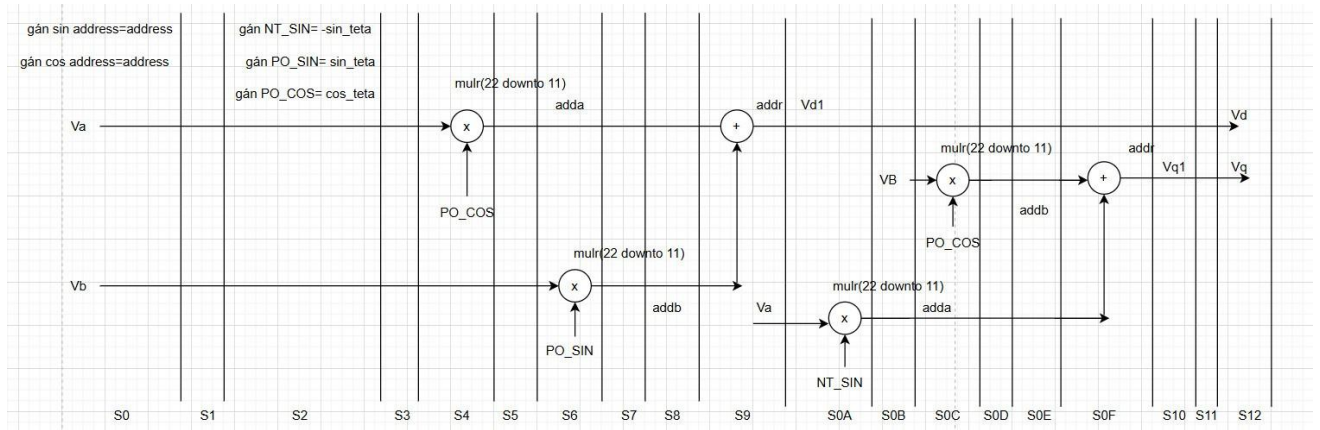


#### 3.1. Công thức biến đổi park thuận:

Công thức park thuận

$$\begin{bmatrix} i_d \\ i_q \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & \sin(\theta_e) \\ -\sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} i_\alpha \\ i_\beta \end{bmatrix}$$

#### 3.2. Lưu đồ thuật toán:



Hình 3.4: Lưu đồ thuật toán Park thuận

### 3.3 Code mô phỏng bộ biến đổi Park thuận

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
LIBRARY lpm;
USE lpm.LPM_COMPONENTS.ALL;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_signed.all;

ENTITY park IS
PORT (
    CLK : IN STD_LOGIC:='0';
    Va,Vb : IN STD_LOGIC_VECTOR(11 downto 0):=(others =>'0');
    ADDRESS : IN STD_LOGIC_VECTOR(11 downto 0):=(others =>'0');
    Vd,Vq : OUT STD_LOGIC_VECTOR(11 downto 0):=(others =>'0'));

END park;

ARCHITECTURE biendoi OF park IS
    signal sin_addr, cos_addr : STD_LOGIC_vector(11 downto 0):=(others =>'0');
    signal sin_teta, cos_teta : STD_LOGIC_VECTOR (11 downto 0):=(others
=>'0');
    signal PO_SIN,PO_COS,NT_SIN: STD_LOGIC_VECTOR (11 downto 0):=(others
=>'0');
    signal CNT : STD_LOGIC_VECTOR (7 DOWNT0 0):=(others =>'0');
    signal Adda, Addb, Addr, Mula, Mulb :STD_LOGIC_VECTOR (11 downto
0):=(others =>'0');
    signal Mulr : STD_LOGIC_vector(23 downto 0):=(others =>'0');
    signal Vd1,Vq1: STD_LOGIC_vector(11 downto 0):=(others =>'0');
BEGIN
    u1 : lpm_rom
        GENERIC MAP(lpm_width => 12, lpm_widthad => 12, lpm_file => "sin_625.mif",
            lpm_address_control => "REGISTERED", lpm_outdata =>
"UNREGISTERED")
        PORT MAP(ADDRESS => sin_addr, inclock => clk, q => sin_teta);
    u2 : lpm_rom
        GENERIC MAP(lpm_width => 12, lpm_widthad => 12, lpm_file => "cos_625.mif",
            lpm_address_control => "REGISTERED", lpm_outdata =>
"UNREGISTERED")
```

```
PORT MAP(ADDRESS => cos_addr, inclock => clk, q => cos_teta);

mull: lpm_mult

    generic
map(LPM_WIDTHA=>12,LPM_WIDTHB=>12,LPM_WIDTHS=>12,LPM_WIDTHP=>24,LPM_REPRESENTATION
=>"SIGNED",LPM_PIPELINE=>1)

    port map(dataa=> mula,datab=> mulb,clock=> clk,result=> mulr);

adder1: lpm_add_sub

    generic map(lpm_width=>12,LPM_REPRESENTATION=>"SIGNED",lpm_pipeline=>1)

    port map(dataa=>adda,datab=>addb,clock=> clk,result=>addr);

-- Enter sin_addr and cos_addr and lpm will give the value at sin_teta and
cos_teta

GEN: block

begin

process (CLK)

begin

if CLK'event and CLK='1' then

    CNT<=cnt+1;

    IF CNT = x"00" then

        sin_addr <= ADDRESS;

        cos_addr <= ADDRESS;

    ELSIF CNT=x"02" THEN

        IF sin_teta = x"000" THEN

            NT_SIN <= x"FFF";

            PO_SIN <= x"000";

        ELSE

            NT_SIN <= -sin_teta;

            PO_SIN <= sin_teta;

        END IF;

        PO_COS <= cos_teta;

    ELSIF CNT =X"04" THEN

        mula<= Va;

        mulb<= PO_COS;

    ELSIF CNT =X"06" THEN

        adda<= mulr(22 downto 11);

        mula<= VB;

        mulb<= PO_SIN;

    ELSIF CNT =X"08" THEN
```



```

        addb<=mulr(22 downto 11);
ELSIF CNT =X"0A" THEN
    Vd1<=addr;
    mula<=Va;
    mulb<= NT_SIN;
ELSIF CNT =X"0C" THEN
    adda<= mulr(22 downto 11);
    mula<= VB;
    mulb<= PO_COS;
ELSIF CNT =X"0E" THEN
    addb<=mulr(22 downto 11);
ELSIF CNT =X"10" THEN
    Vq1<=addr;
ELSIF CNT =X"12" THEN
    Vq<=Vq1;
    Vd<=VD1;
    CNT<=X"00";
END IF;
END IF;
    end process;
end block GEN;
END ARCHITECTURE;

```

#### 4. Code khối Clark ngược sửa đổi – Park ngược

##### 4.1 Công thức biến đổi Clark ngược sửa đổi và Park ngược:

Công thức chuyển đổi:

$$\text{Address} = \frac{\text{góc}}{2\pi} \times 624$$

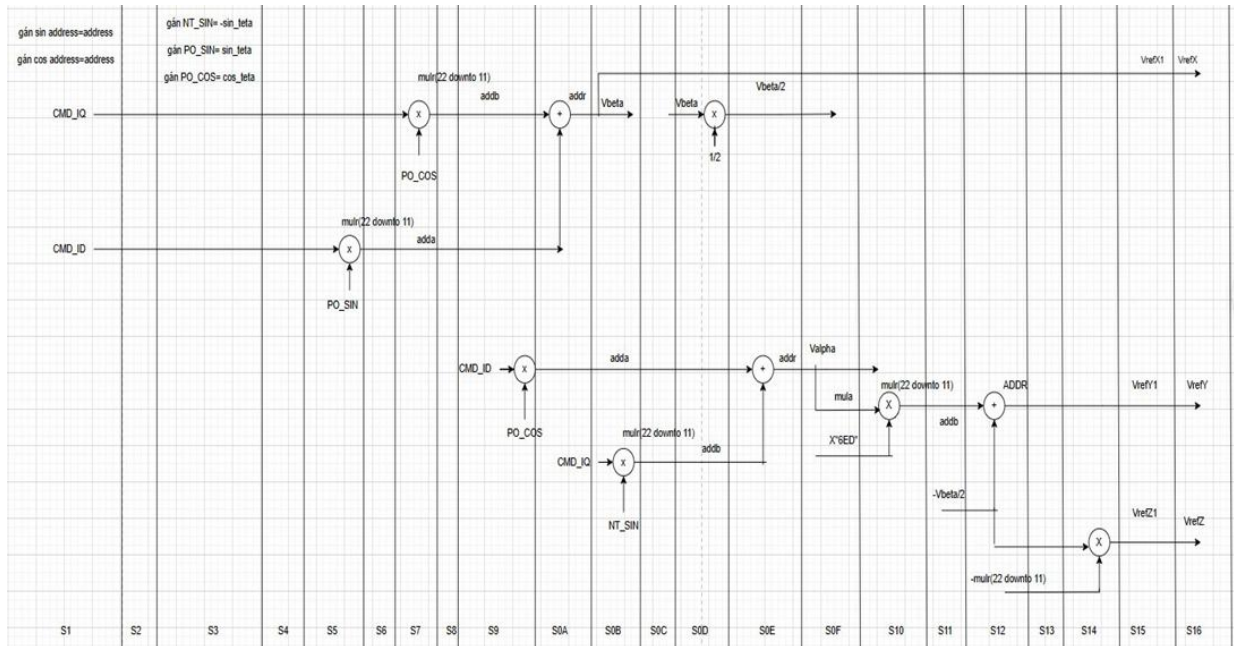
Công thức Clark ngược sửa đổi:

$$\begin{bmatrix} V_{ref1} \\ V_{ref2} \\ V_{ref3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{-1}{2} & \frac{\sqrt{3}}{2} \\ \frac{-1}{2} & \frac{-\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} V_{\beta} \\ V_{\alpha} \end{bmatrix}$$

Công thức Park ngược:

$$\begin{bmatrix} V_\alpha \\ V_\beta \end{bmatrix} = \begin{bmatrix} \cos(\theta_e) & -\sin(\theta_e) \\ \sin(\theta_e) & \cos(\theta_e) \end{bmatrix} \begin{bmatrix} V_d \\ V_q \end{bmatrix}$$

#### 4.2 Lưu đồ thuật toán



Hình 3.5: Lưu đồ thuật toán Clark ngược sửa đổi và Park ngược

#### 4.3 Code mô tả bộ biến đổi clark sửa đổi và park ngược

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
LIBRARY lpm;
USE lpm.LPM_COMPONENTS.ALL;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_signed.all;

ENTITY sin_cos IS
PORT (
    CLK,CLK40: IN STD_LOGIC:='0';
    CMD_ID,CMD_IQ, ADDRESS: IN STD_LOGIC_VECTOR(11 downto 0):=(others =>'0');
    Vref1,Vref2,Vref3: OUT STD_LOGIC_VECTOR(11 downto 0):=(others =>'0');
    done : out std_logic:='0'
);
END sin_cos;

ARCHITECTURE sin_cos_arch OF sin_cos IS
    signal sin_addr, cos_addr: STD_LOGIC_vector(11 downto 0):=(others =>'0');

```

```
    signal sin_teta, cos_teta: STD_LOGIC_VECTOR (11 downto 0):=(others =>'0');
    signal PO_SIN,PO_COS,NT_SIN: STD_LOGIC_VECTOR (11 downto 0):=(others
=>'0');
    signal CNT: STD_LOGIC_VECTOR (7 DOWNTO 0):=(others =>'0');
    signal Adda, Addb, Addr, Mula, Mulb: STD_LOGIC_VECTOR (11 downto
0):=(others =>'0');
    signal Mulr: STD_LOGIC_vector(23 downto 0):=(others =>'0');
    signal CMD_Vbeta, CMD_Vbetaa, CMD_Valpha: STD_LOGIC_vector(11 downto
0):=(others =>'0');
    signal Vrefx, Vrefy, Vrefz: STD_LOGIC_vector(11 downto 0):=(others =>'0');

BEGIN

u1 : lpm_rom
    GENERIC MAP(lpm_width => 12, lpm_widthad => 12, lpm_file => "sin_625.mif",
        lpm_address_control => "REGISTERED", lpm_outdata =>
"UNREGISTERED")
    PORT MAP(ADDRESS => sin_addr, inclock => clk, q => sin_teta);
u2 : lpm_rom
    GENERIC MAP(lpm_width => 12, lpm_widthad => 12, lpm_file => "cos_625.mif",
        lpm_address_control => "REGISTERED", lpm_outdata =>
"UNREGISTERED")
    PORT MAP(ADDRESS => cos_addr, inclock => clk, q => cos_teta);

mull: lpm_mult
generic
map(LPM_WIDTHHA=>12,LPM_WIDTHHB=>12,LPM_WIDTHHS=>12,LPM_WIDTHHP=>24,LPM_REPRESENTATION
=>"SIGNED",LPM_PIPELINE=>1)
port map(dataa=> mula,datab=> mulb,clock=> clk,result=> mulr);
adder1: lpm_add_sub
generic map(lpm_width=>12,LPM_REPRESENTATION=>"SIGNED",lpm_pipeline=>1)
port map(dataa=>adda,datab=>addb,clock=> clk,result=>addr);
-- Enter sin_addr and cos_addr and lpm will give the value at sin_teta and
cos_teta
GEN : block
begin
process (CLK40)
begin
if CLK40'event and CLK40='1' then
```

```
CNT<=cnt+1;
IF CNT = x"00" then
    sin_addr <= ADDRESS;
    cos_addr <= ADDRESS;
ELSIF CNT=x"02" THEN
IF sin_teta = x"000" THEN
    NT_SIN <= x"FFF";
    PO_SIN <= x"000";
ELSE
    NT_SIN <= -sin_teta;
    PO_SIN <= sin_teta;
END IF;
    PO_COS <= cos_teta;
ELSIF CNT=x"04" THEN
    MULA <= CMD_ID;
    MULB <= PO_SIN;
ELSIF CNT=x"06" THEN
    ADDA <= MULR(22 DOWNT0 11); --CMD_ID*POSIN
    MULA <= CMD_IQ;
    MULB <= PO_COS;
ELSIF CNT=x"08" THEN
    ADDB <= MULR(22 DOWNT0 11); -- CMD_ID*POCOS
    MULA <= CMD_ID;
    MULB <= PO_COS;
ELSIF CNT=x"0A" THEN
    ADDA <= MULR(22 DOWNT0 11); -- cmd_id*po_cos :B1 TÍNH VALPHA
    MULA <= CMD_IQ;
    MULB <= NT_SIN;
    CMD_Vbeta <= ADDR; -- cmd_id*posin+cmd_iq*po_cos
    Vrefx <= ADDR; -- cmd_Vbeta

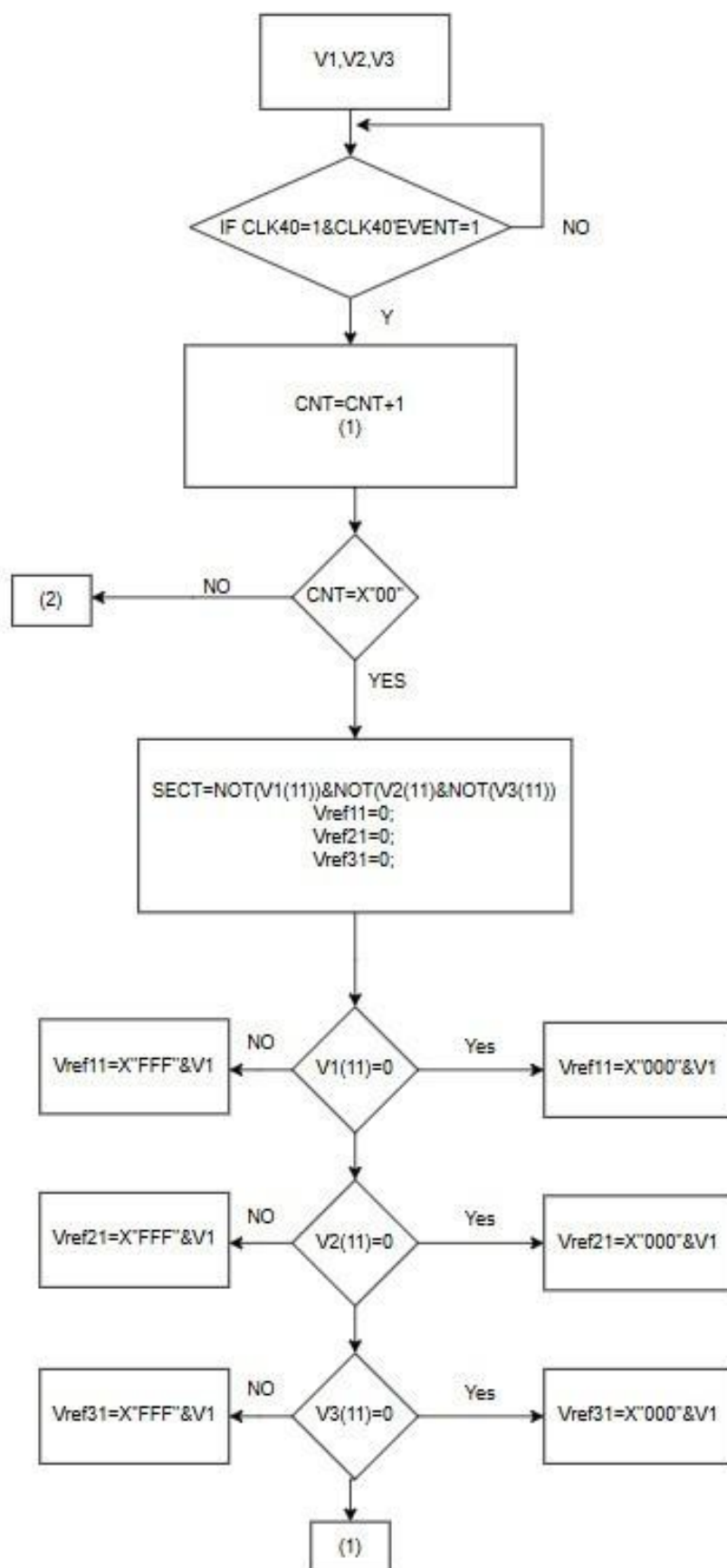
ELSIF CNT=x"0C" THEN
    ADDB <= MULR(22 DOWNT0 11); -- cmd_iq*nt_sin
    CMD_Vbetaa <= CMD_Vbeta(11)& CMD_Vbeta(11 downto 1); -- CMD_Vbeta/2
ELSIF CNT=x"0E" THEN
    CMD_Valpha <= ADDR; -- cmd_id*po_cos+cmd_iq*nt_sin
```

```
MULA <= ADDR;          -- Cmd_Valpha
MULB <= x"6ED";         -- sqrt(3)/2 (0.866*2048 (2048=2^11: 12 bits))
ELSIF CNT=x"10" THEN
    ADDA <= -CMD_Vbetaa;
    ADDB <= MULR(22 DOWNT0 11); -- cmd_Valpha*sqrt(3)/2
ELSIF CNT=x"12" THEN
    Vrefy <= ADDR; -- -CMD_Vbeta/2+cmd_Valpha*sqrt(3)/2
    ADDA <= -CMD_Vbetaa;
    ADDB <= -MULR(22 DOWNT0 11);
ELSIF CNT=x"14" THEN
    Vrefz <= ADDR;
ELSIF CNT = x"16" THEN
    Vref1 <= Vrefx;
    Vref2 <= Vrefy;
    Vref3 <= Vrefz;
    done <= '1' , '0' after 50 ns;
    CNT <= x"00";
END IF;
END IF;
end process;
end block GEN;
END sin_cos_arch;
```

## **5. Lập trình thuật toán SVPWM**

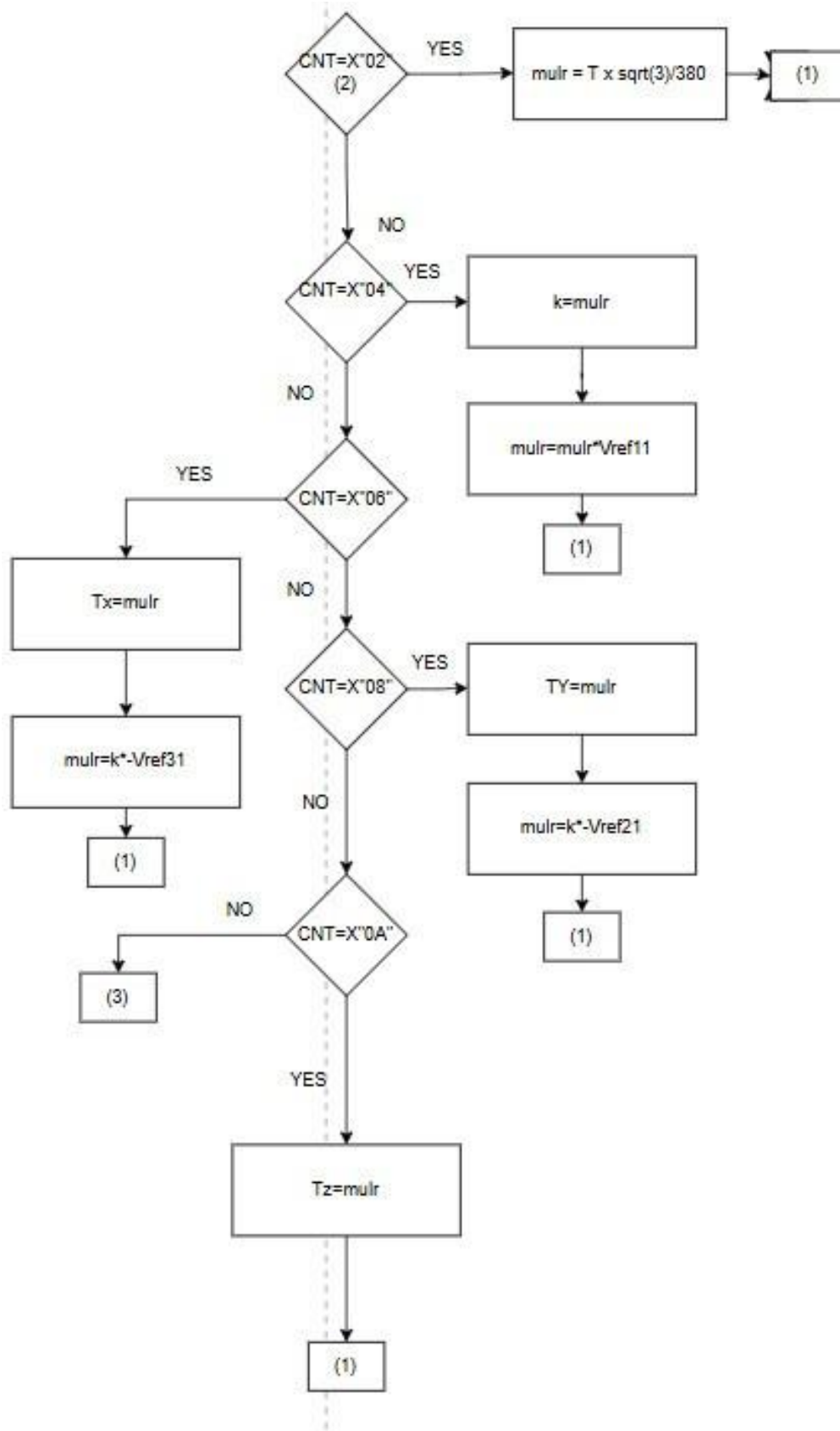
### ***5.1 Lưu đồ thuật toán***

Bước 1 : Từ  $V_{ref1}$ ,  $V_{ref2}$ ,  $V_{ref3}$  tính section



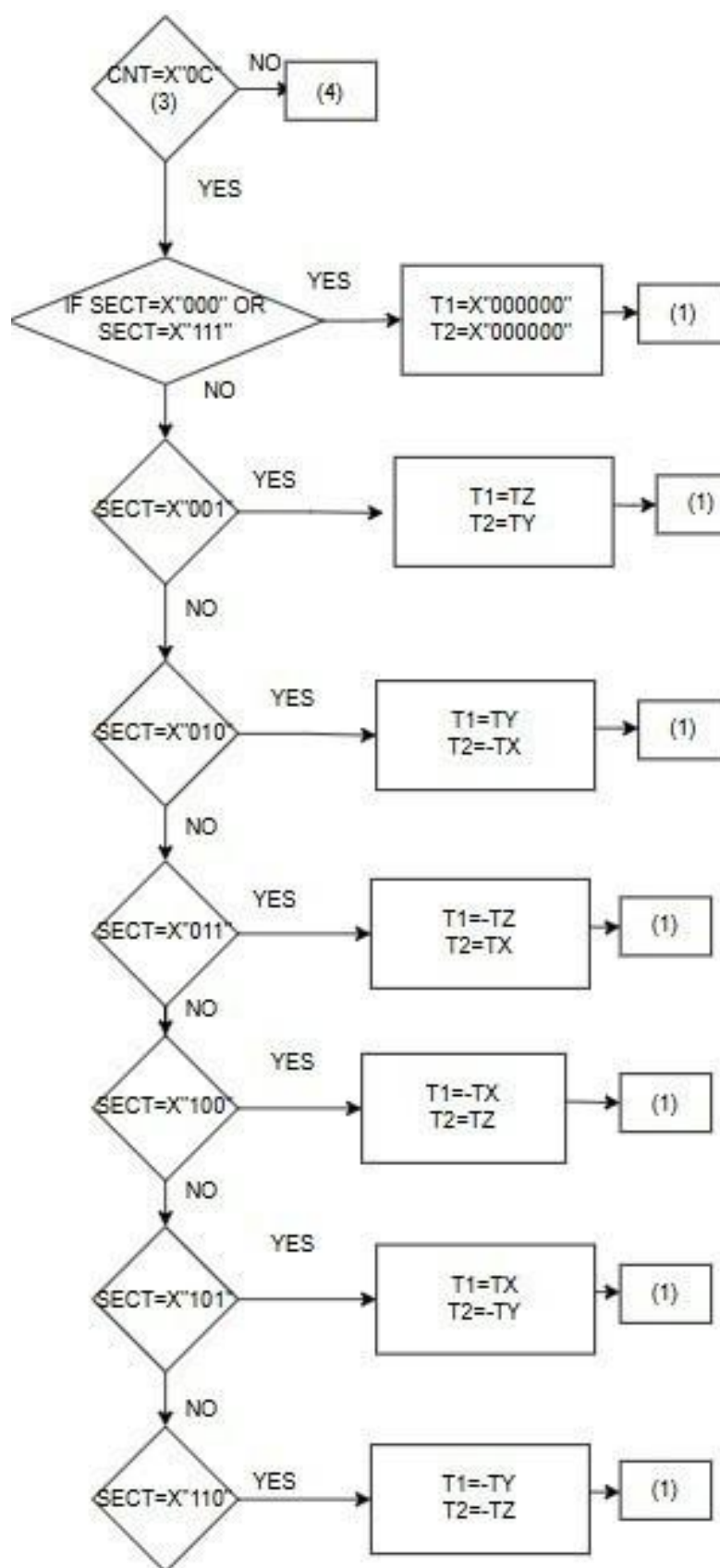
Hình 3.6: Bước 1 lập trình thuật toán SVPWM

Bước 2 : Tính Tx, Ty, Tz



Hình 3.7: Bước 2 lập trình thuật toán SVPWM

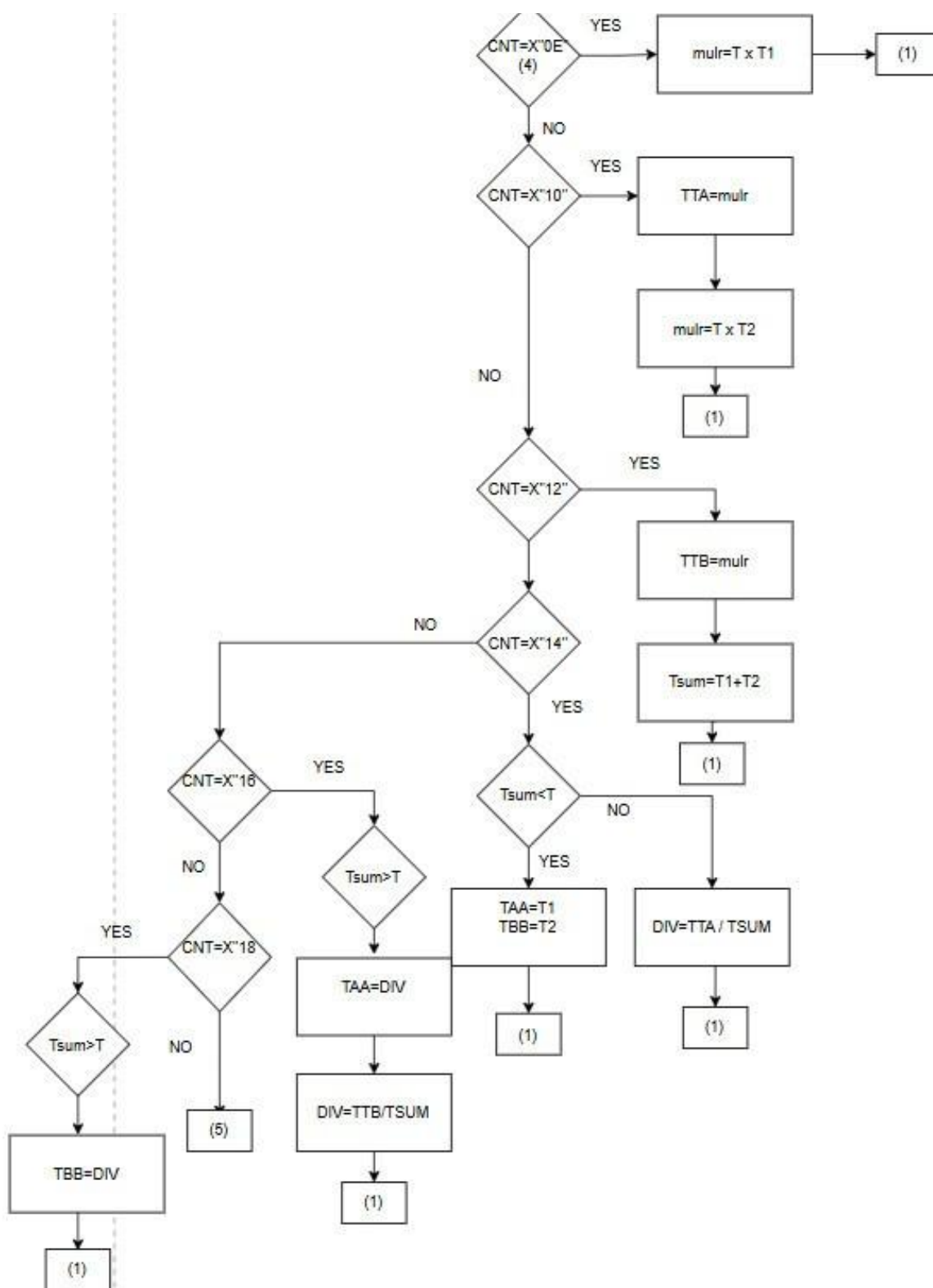
Bước 3 : Tính  $T_1, T_2$



Hình 3.8: Bước 3 lập trình thuật toán SVPWM

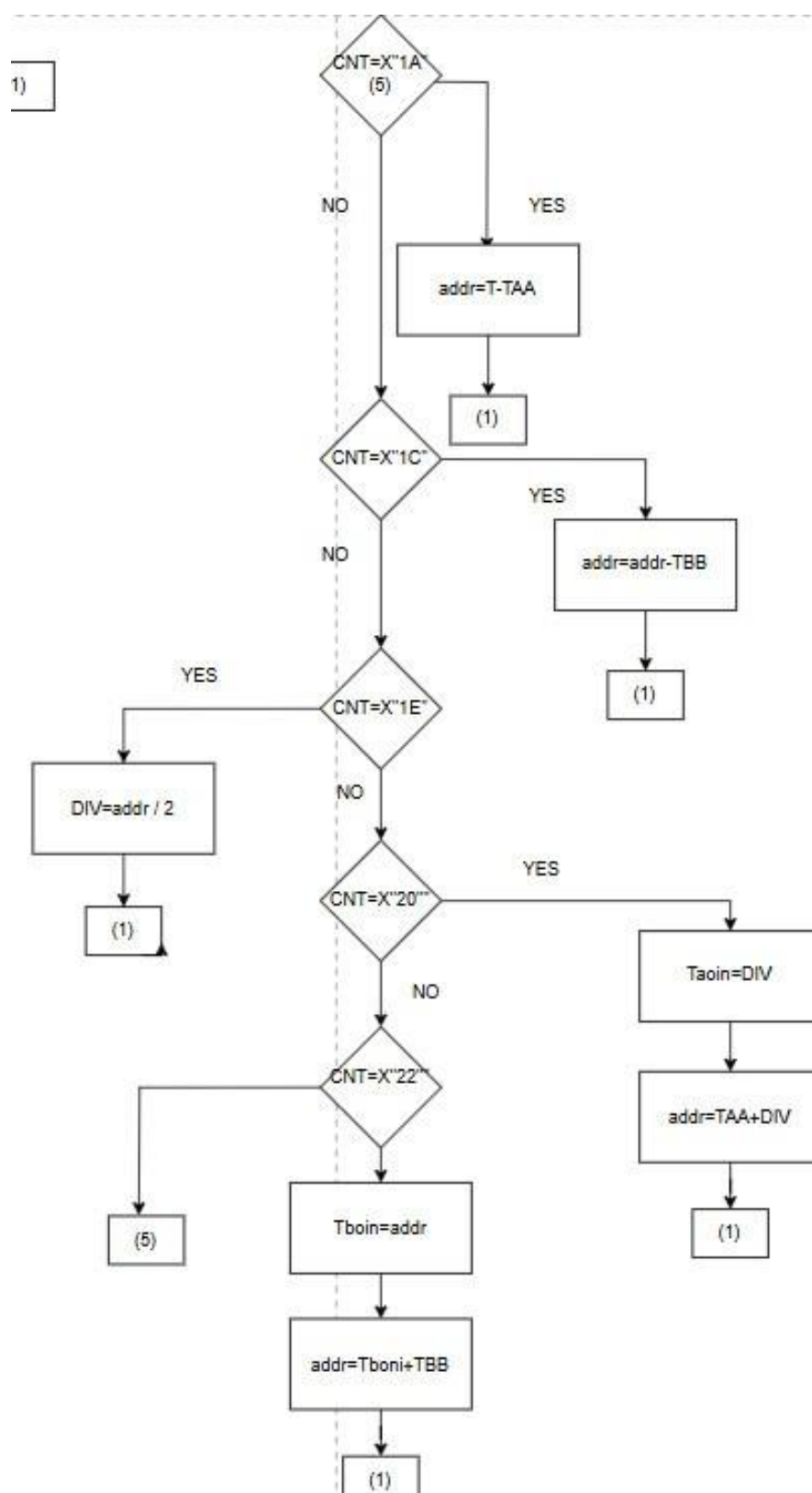


Bước 4 : Tính TAA, TBB



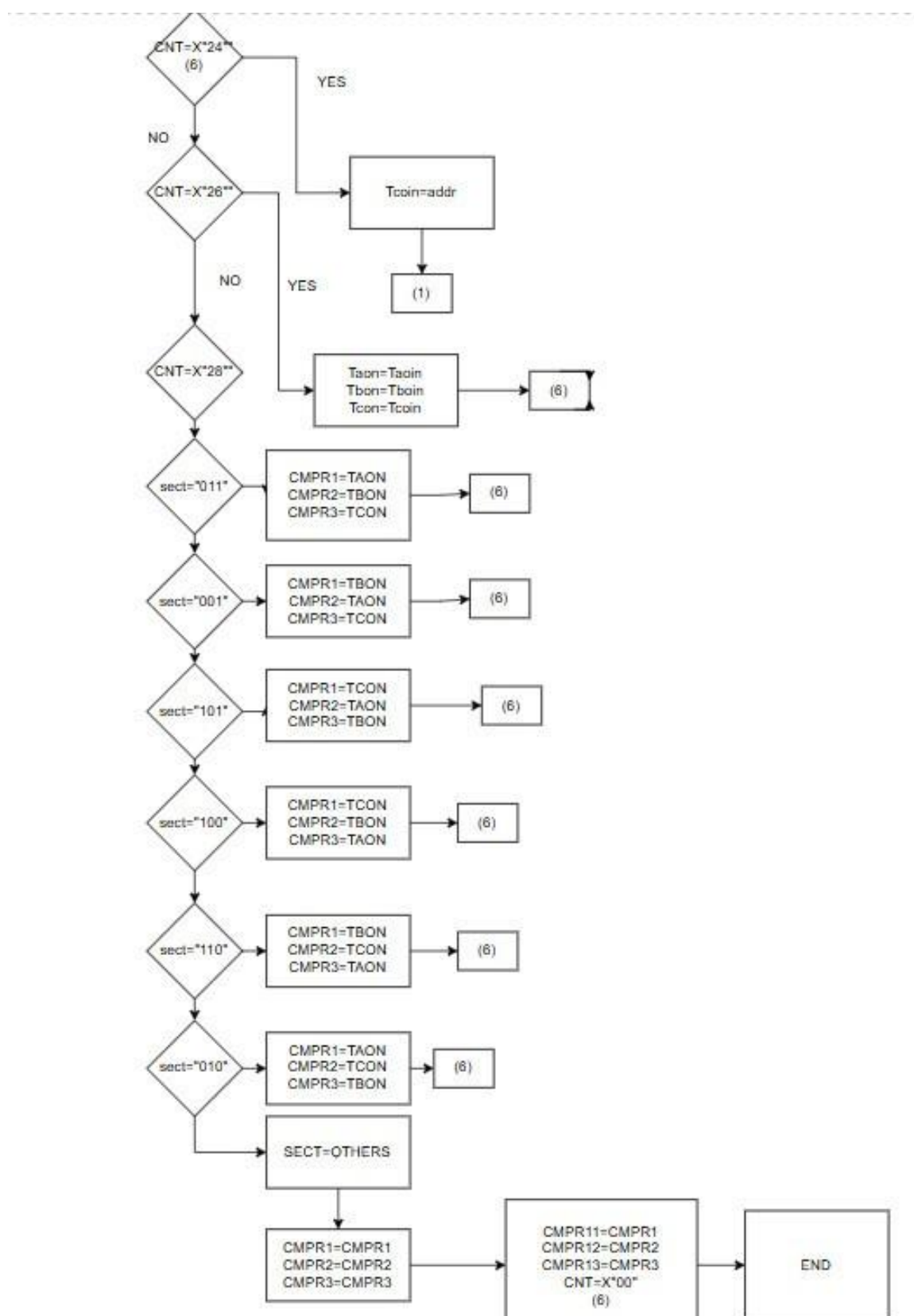
Hình 3.9: Bước 4 lập trình thuật toán SVPWM

Bước 5 : Tính  $T_{aon}$ ,  $T_{bon}$ ,  $T_{con}$



Hình 3.10: Bước 5 lập trình thuật toán SVPWM

Bước 6 : Tính CMPR1, CMPR2, CMPR3



Hình 3.11: Bước 6 lập trình thuật toán SVPWM

## 5.2 Code mô phỏng

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE IEEE.std_logic_signed.all;
USE IEEE.std_logic_signed.all;
USE IEEE.numeric_std.all;
LIBRARY lpm;
USE lpm.LPM_COMPONENTS.ALL;
USE ieee.std_logic_arith.all;

ENTITY CMPR IS
PORT (
    CLK,clk40: IN STD_LOGIC:='0';
    Vref1,Vref2,Vref3: IN STD_LOGIC_VECTOR(11 downto 0):=(others =>'0');
    CMPR11,CMPR12,CMPR13 : OUT STD_LOGIC_VECTOR(23 DOWNT0 0):=(OTHERS =>'0')
);
END entity;

architecture arch of CMPR IS
    signal Vref11,Vref21,Vref31,CMPR1,CMPR2,CMPR3: STD_LOGIC_VECTOR(23 downto 0):=(others =>'0');
    signal TTA,TTB,TSUM,A,B,SAT,TAM: STD_LOGIC_VECTOR(23 downto 0):=(others =>'0');
    SIGNAL Taoni,TBoni,TConi,adda2,addb2,addr2,A2,B2,SAT2,sat3 :
STD_LOGIC_VECTOR(23 downto 0):=(others =>'0');
    signal CNT: STD_LOGIC_VECTOR (7 DOWNT0 0):=(others =>'0');
    signal Adda, Addb, Addr, Mula, Mulb: STD_LOGIC_VECTOR (23 downto 0):=(others
=>'0');
    signal Mulr: STD_LOGIC_VECTOR(47 downto 0):=(others =>'0');
    signal k: STD_LOGIC_VECTOR( 23 DOWNT0 0):=(OTHERS =>'0');
    signal Tx,Ty,Tz,T1,T2,TAA,TBB,TAO,TBO,TCO: STD_LOGIC_VECTOR( 23 DOWNT0
0):=(OTHERS =>'0');
    signal sect: STD_LOGIC_VECTOR(2 downto 0):=(others =>'0');
    signal a1: std_logic_vector(23 downto 0):= X"00955C"; -- SQRT(3)/380;
    signal T: std_logic_vector(23 downto 0):= X"000347"; -- 0.0001;
    signal PWMEA_1,PWMEA_2,PWMEB_1: STD_LOGIC:='0';
    signal PWMEB_2, PWMEC_1,PWMEC_2: STD_LOGIC:='0';
```

```
signal CMPAA,CMPBB,CMPCC: STD_LOGIC_VECTOR(23 downto 0):=(others =>'0');

begin

    mull: lpm_mult
        generic
        map(LPM_WIDTHA=>24,LPM_WIDTHB=>24,LPM_WIDTHS=>24,LPM_WIDTHP=>48,LPM_REPRESENTATION
=>"SIGNED",LPM_PIPELINE=>1)
        port map(dataa=> mula,datab=> mulb,clock=> CLK,result=> mulr);

    adder1: lpm_add_sub
        generic map(lpm_width=>24,LPM_REPRESENTATION=>"SIGNED",lpm_pipeline=>1)
        port map(dataa=>adda,datab=>addb,clock=> CLK,result=>addr);

    adder2: lpm_add_sub
        generic map(lpm_width=>24,LPM_REPRESENTATION=>"SIGNED",lpm_pipeline=>1)
        port map(dataa=>adda2,datab=>addb2,clock=> CLK,result=>addr2);

    m0 : lpm_divide-----divide component
        GENERIC MAP (LPM_WIDTHN=>24, LPM_WIDTHD =>24,
LPM_PIPELINE=>1,LPM_NREPRESENTATION =>"SIGNED", LPM_DREPRESENTATION =>"SIGNED")
        port map (numer=>A,denom=>B,clock=> CLK,quotient=>sat);

    m1 : lpm_divide-----divide component
        GENERIC MAP (LPM_WIDTHN=>24, LPM_WIDTHD =>24,
LPM_PIPELINE=>1,LPM_NREPRESENTATION =>"SIGNED", LPM_DREPRESENTATION =>"SIGNED")
        port map (numer=>A2,denom=>B2,clock=> CLK,quotient=>sat2,remain=>sat3);

    GEN : BLOCK
    BEGIN
        PROCESS (CLK40,sect)
        BEGIN
            IF CLK40'EVENT AND CLK40='1' THEN
                CNT <= CNT + 1;
                IF CNT = x"00" THEN
                    sect <= not( Vref3(11)) & not (Vref2(11))& not(Vref1(11));
                    vref11 <= (OTHERS => '0');
                    vref21 <= (OTHERS => '0');
                    vref31 <= (OTHERS => '0');
```

```
IF Vref1(11) = '0' THEN
    vref11 <= X"000" & Vref1;
ELSE
    vref11 <= X"FFF" & Vref1;
END IF;
IF Vref2(11) = '0' THEN
    vref21 <= X"000" & Vref2;
ELSE
    vref21 <= X"FFF" & Vref2;
END IF;
IF Vref3(11) = '0' THEN
    vref31 <= X"000" & Vref3;
ELSE
    vref31 <= X"FFF" & Vref3;
END IF;
ELSIF CNT = x"02" THEN
    mula <= a1;
    mulb <= T;
ELSIF CNT = x"04" THEN
    mula <= mulr(46 downto 23);
    k <= mulr(46 downto 23);
    mulb <= Vref11;
ELSIF CNT = x"06" THEN
    Tx <= mulr(23 downto 0);
    mula <= k;
    mulb <= -Vref31;
ELSIF CNT = x"08" THEN
    Ty <= mulr(23 downto 0);
    mula <= k;
    mulb <= -Vref21;
ELSIF cnt = x"0A" THEN
    Tz <= mulr(23 downto 0);
ELSIF cnt = x"0C" THEN
    IF sect = "000" OR sect = "111" THEN
        T1 <= X"000000";
        T2 <= X"000000";
```

```
ELSIF sect = "001" THEN
    T1 <= Tz;
    T2 <= Ty;
ELSIF sect = "010" THEN
    T1 <= Ty;
    T2 <= -Tx;
ELSIF sect = "011" THEN
    T1 <= -Tz;
    T2 <= Tx;
ELSIF sect = "100" THEN
    T1 <= -Tx;
    T2 <= Tz;
ELSIF sect = "101" THEN
    T1 <= Tx;
    T2 <= -Ty;
ELSIF sect = "110" THEN
    T1 <= -Ty;
    T2 <= -Tz;
END IF;
ELSIF cnt = x"0E" THEN
    mula <= T;
    mulb <= T1;
ELSIF cnt = x"10" THEN
    TTA <= mulr(23 downto 0);
    mula <= T;
    mulb <= T2;
ELSIF cnt = x"12" THEN
    TTb <= mulr(23 downto 0);
    Tsum <= T1+T2;
ELSIF cnt = x"14" THEN
    IF TSUM < T THEN
        TAA <= T1;
        TBB <= T2;
    ELSE
        A <= TTA;
        B <= Tsum;
```

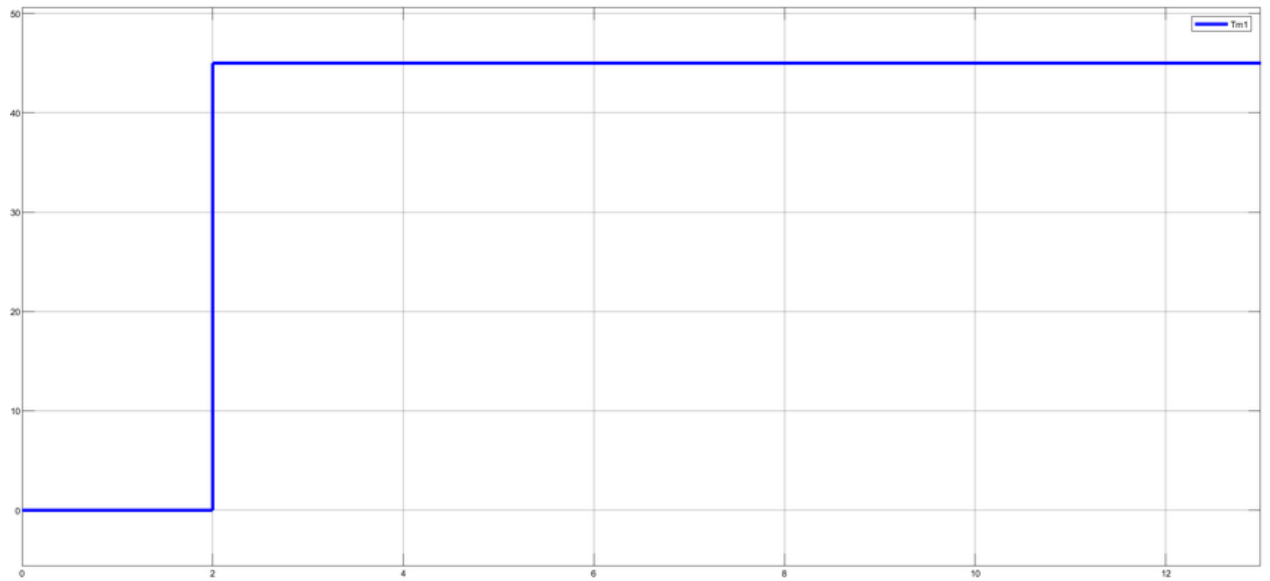
```
END IF;
ELSIF cnt = x"16" THEN
  IF TSUM >= T THEN
    TAA <= SAT(23 DOWNT0 0)+X"00003C";
    A <= TTB;
    B <= TSUM;
  END IF;
ELSIF cnt = x"18" THEN
  IF TSUM > T THEN
    TBB <= SAT(23 DOWNT0 0)+X"00003C";
  END IF;
ELSIF cnt = x"1A" THEN
  adda2 <= T;
  addb2 <= -(TAA);
ELSIF cnt = x"1C" THEN
  adda2 <= addr2;
  addb2 <= -(TBB);
ELSIF cnt = x"1E" THEN
  A2 <= addr2;
  B2 <= X"000002";
ELSIF cnt = x"20" THEN
  IF sat2 > X"000000" THEN
    Taoni <= SAT2(23 DOWNT0 2)&Sat3(1 downto 0);
    adda2 <= SAT2(23 DOWNT0 2)& sat3(1 downto 0);
  ELSE
    Taoni <= SAT2(23 DOWNT0 0);
    adda2 <= SAT2(23 DOWNT0 0) ;
  END IF;
  addb2 <= TAA;
ELSIF cnt = x"22" THEN
  TBoni <= addr2;
  adda2 <= Tboni;
  addb2 <= TBB;
ELSIF cnt = x"24" THEN
  TConi <= addr2;
ELSIF cnt = x"26" THEN
```



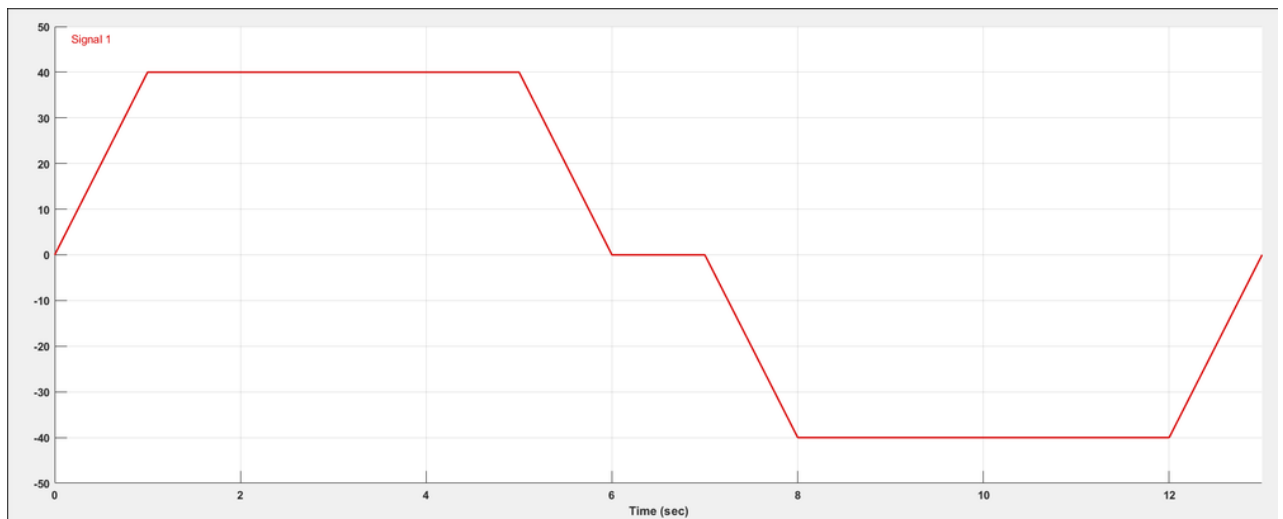
```
Ta0 <= Taoni;
Tb0 <= Tboni;
Tc0 <= Tconi;
ELSIF cnt = x"28" THEN
CASE sect IS
WHEN "011" =>
    CMPR1 <= TA0;
    CMPR2 <= TB0;
    CMPR3 <= TC0;
WHEN "001" =>
    CMPR1 <= TB0;
    CMPR2 <= TA0;
    CMPR3 <= TC0;
WHEN "101" =>
    CMPR1 <= TC0;
    CMPR2 <= TA0;
    CMPR3 <= TB0;
WHEN "100" =>
    CMPR1 <= TC0;
    CMPR2 <= TB0;
    CMPR3 <= TA0;
WHEN "110" =>
    CMPR1 <= TB0;
    CMPR2 <= TC0;
    CMPR3 <= TA0;
WHEN "010" =>
    CMPR1 <= TA0;
    CMPR2 <= TC0;
    CMPR3 <= TB0;
WHEN OTHERS =>
    CMPR1 <= CMPR1;
    CMPR2 <= CMPR2;
    CMPR3 <= CMPR3;
END CASE;
CMPR11 <= CMPR1;
CMPR12 <= CMPR2;
```

```
        CMPR13 <= CMPR3;  
        CNT <= x"00";  
    END IF;  
END IF;  
END PROCESS;  
END BLOCK gen;  
END architecture;
```



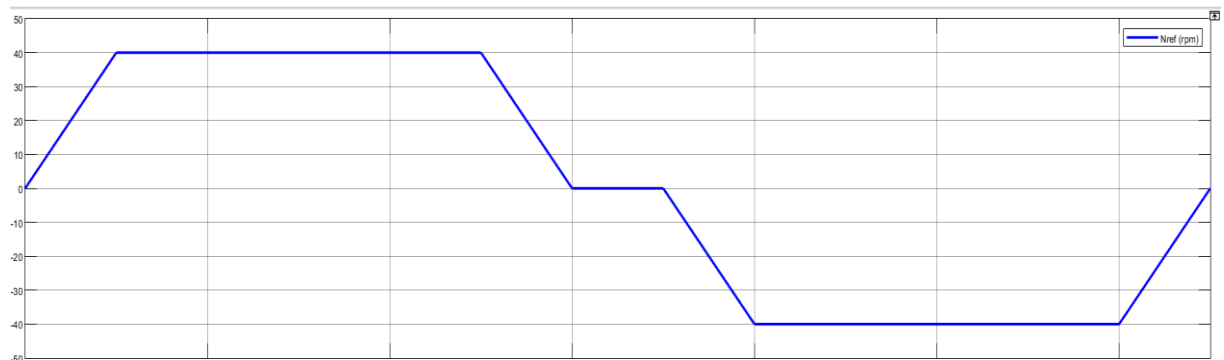


Hình 4.2: Momen tải yêu cầu

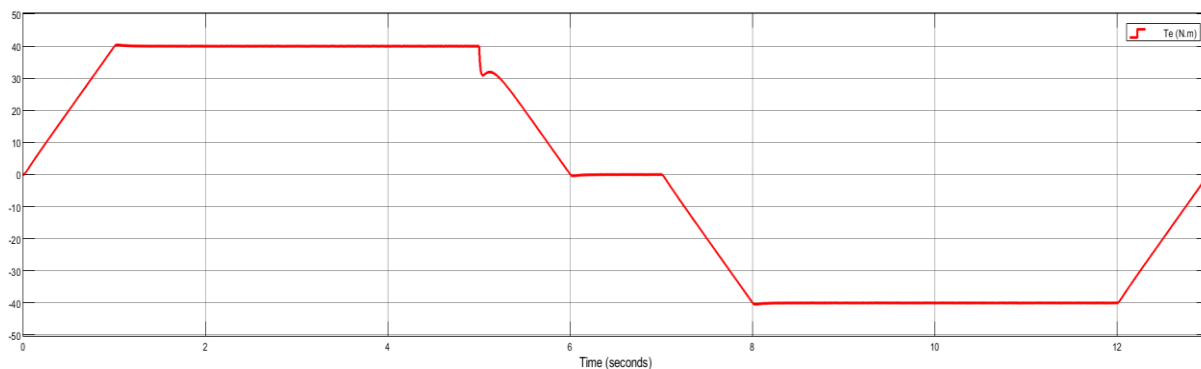


Hình 4.3: Tốc độ đặt của động cơ

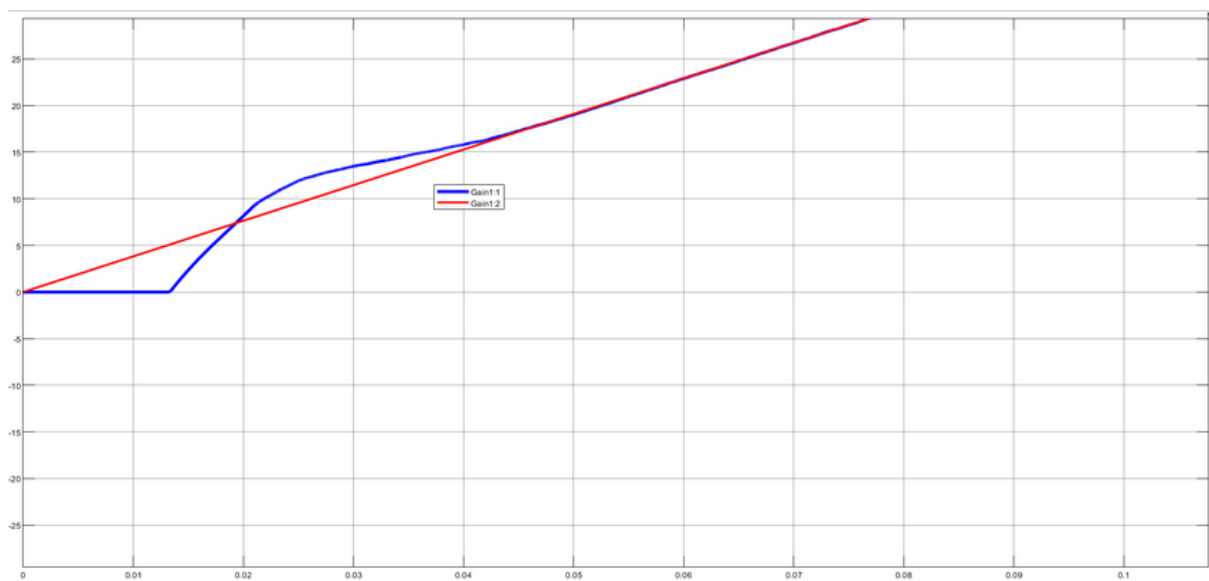
## 1.2 Kiểm nghiệm và đánh giá kết quả mô phỏng



Hình 4.4: Kết quả mô phỏng mong muốn



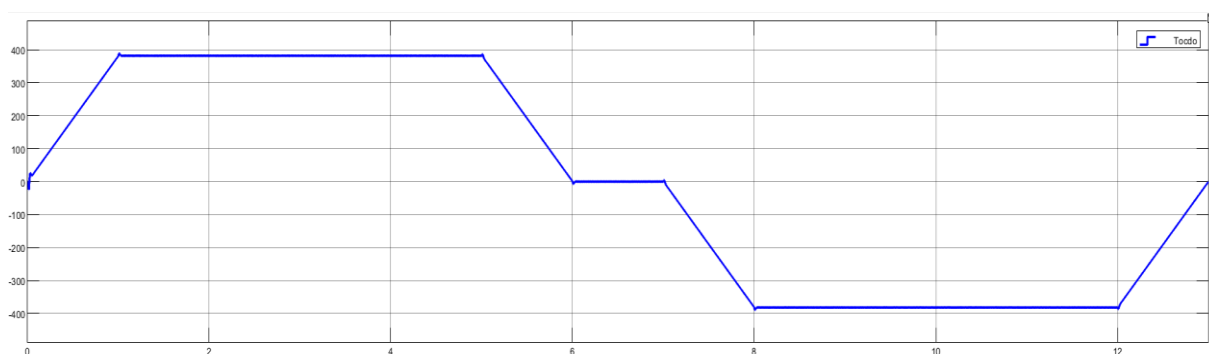
Hình 4.5: Kết quả mô phỏng tốc độ thực



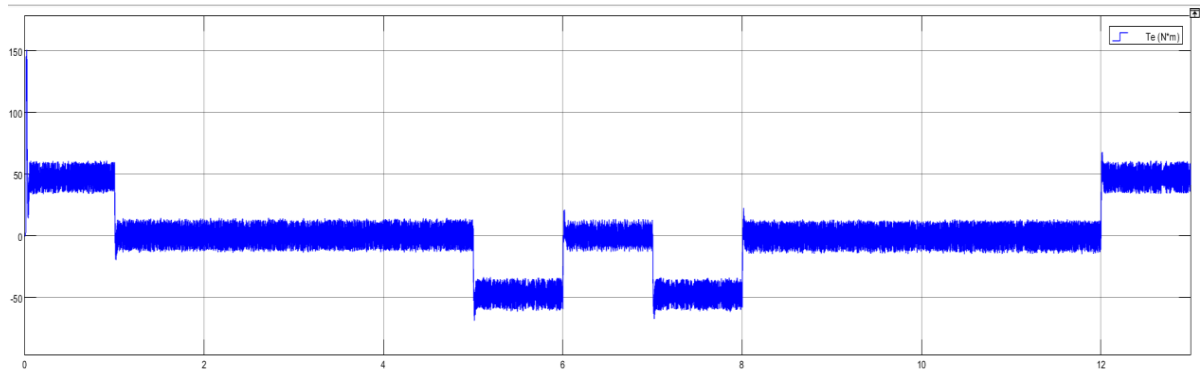
Hình 4.6: Thời gian xác lập của tốc độ

**Nhận xét:** Tốc độ động cơ bám sát tốc độ đặt và đạt xác lập nhanh ( $t < 0,05s$ ). Tại các thời điểm momen và tốc độ thay đổi vẫn có độ quá điều chỉnh nhỏ.

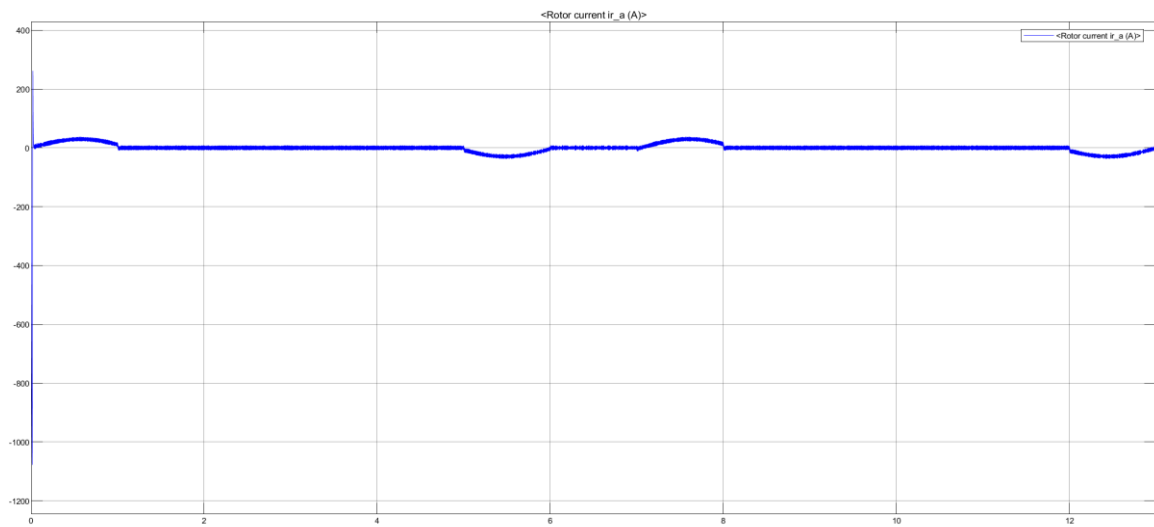
### 1.3 Tốc độ, momen và dòng điện



Hình 4.7: Tốc độ động cơ



Hình 4.8: Momen động cơ

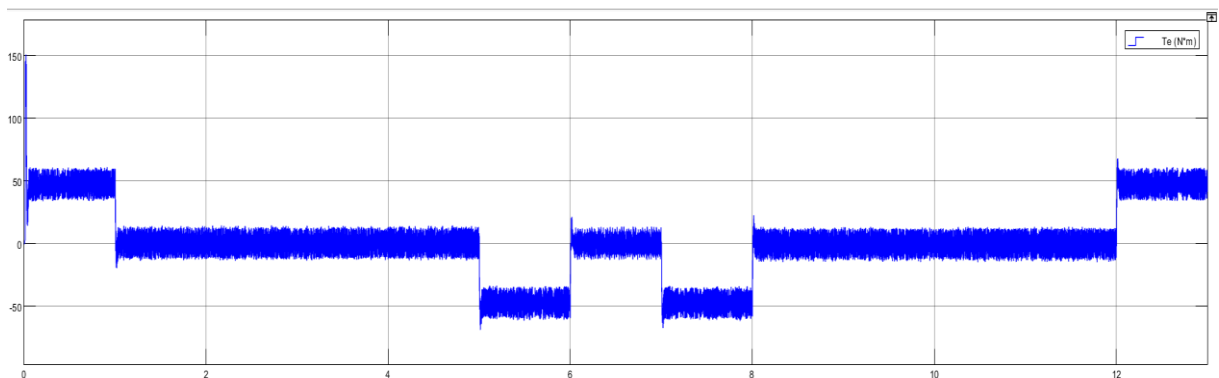


Hình 4.9: Dòng điện động cơ

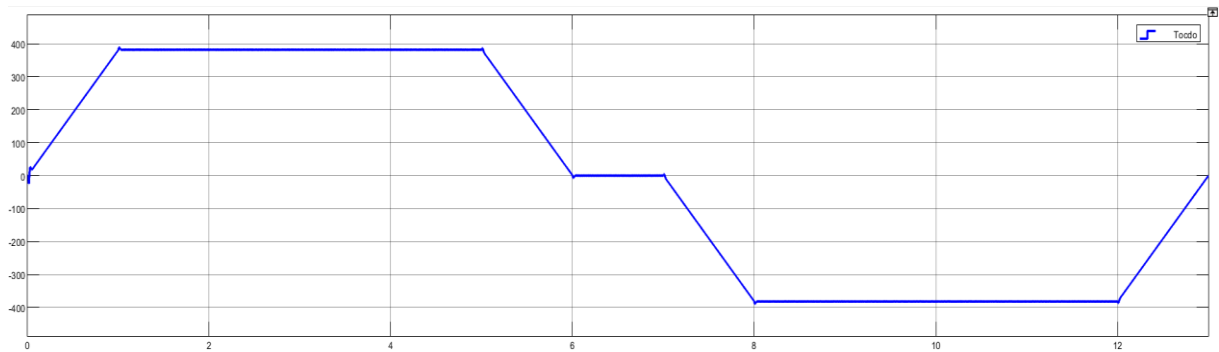
### Nhận xét:

- Các khoảng thời gian tốc độ yêu cầu tăng thì momen và dòng điện cũng tăng để đưa tốc độ thực đáp ứng với yêu cầu.
- Khi tốc độ động cơ không biến thiên momen và dòng điện cũng không thay đổi.
- Momen và dòng điện đảo chiều khi tốc độ đảo chiều.

### 1.4 Tương quan giữa momen và tốc độ



Hình 4.10: Mô phỏng momen



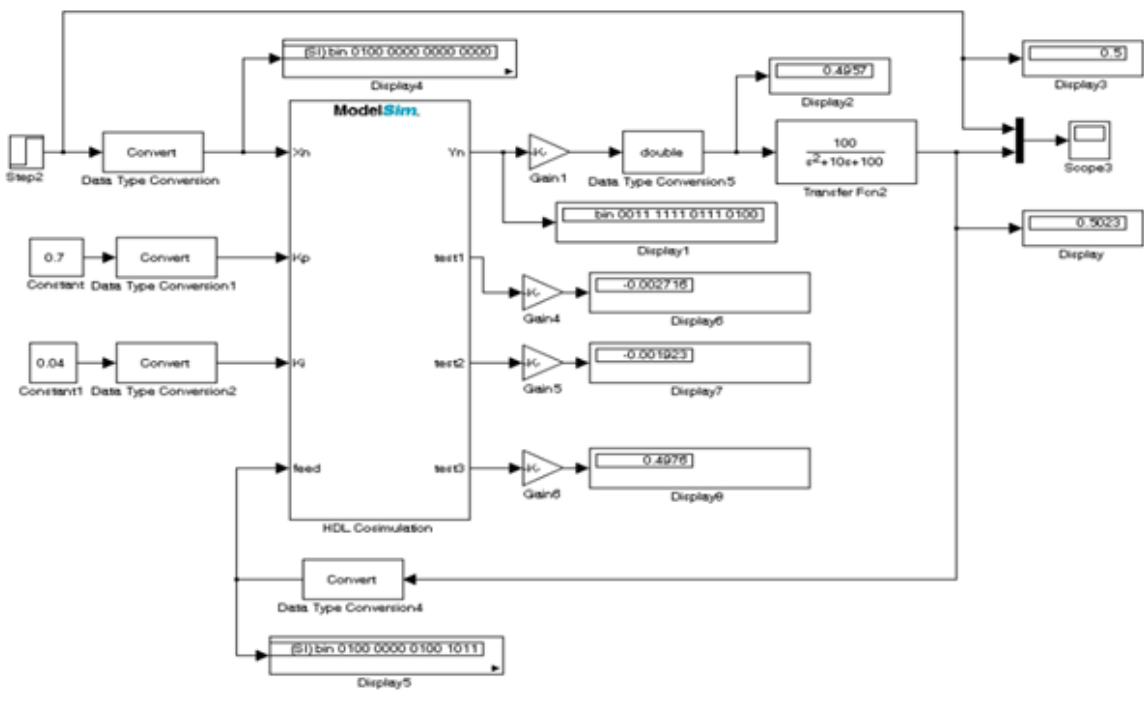
Hình 4.11: Mô phỏng tốc độ

### Nhận xét:

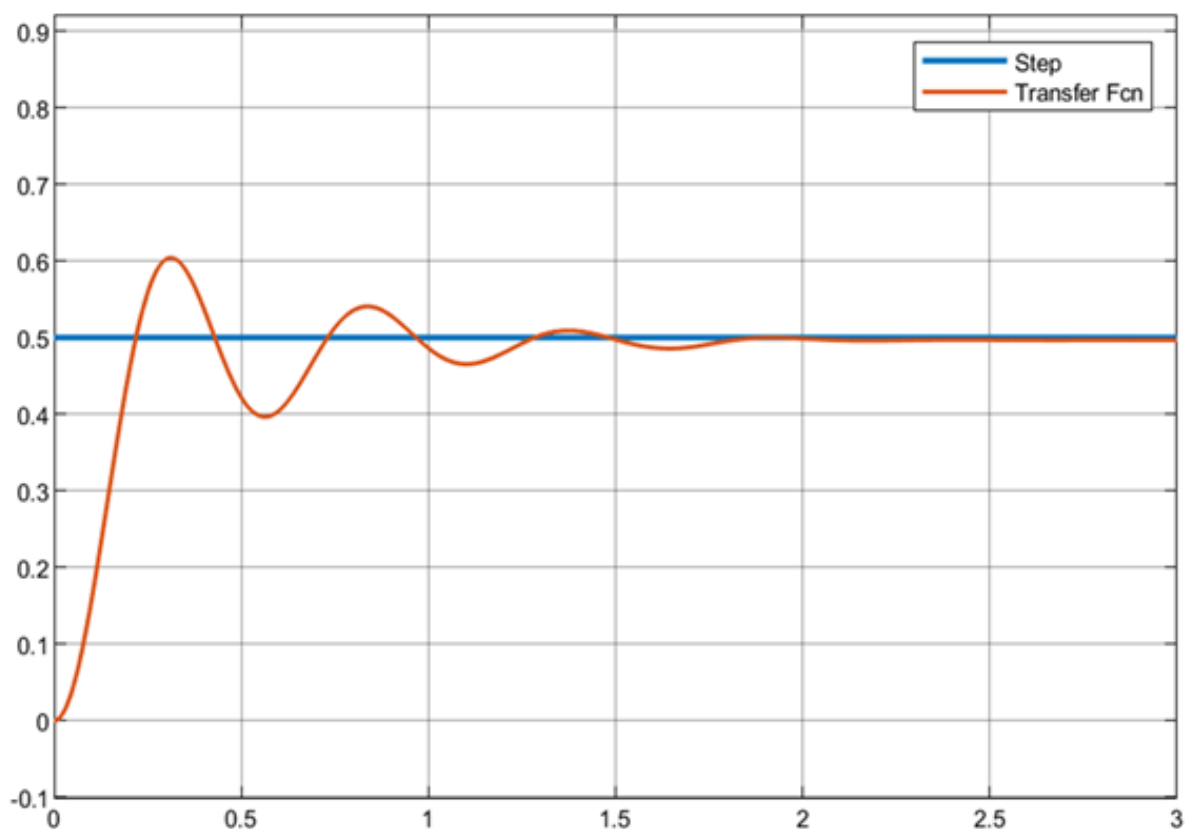
- Khi tốc độ yêu cầu của động cơ tăng, điện áp đặt vào phần ứng của động cơ cũng tăng để tốc độ thực đạt yêu cầu.
- Khi tốc độ không đổi, điện áp đặt vào động cơ và momen không đổi.
- Khi tốc độ đảo chiều, bộ chỉnh lưu cũng đảo chiều điện áp cấp cho động cơ, momen đảo chiều tương ứng.

## 2. Mô phỏng và kiểm nghiệm code Modelsim trên Matlab/Simulink

### 2.1. Mô phỏng và kiểm nghiệm bộ điều khiển PI

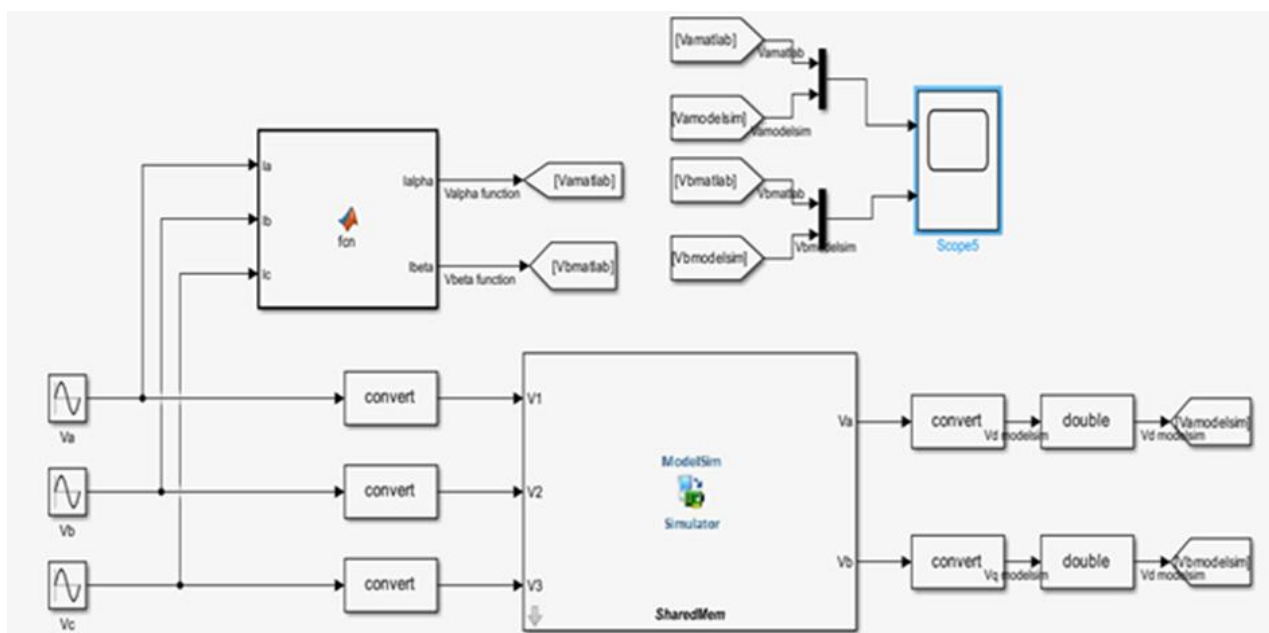


Hình 4.12: Mô phỏng Matlab/Simulink bộ điều khiển PI



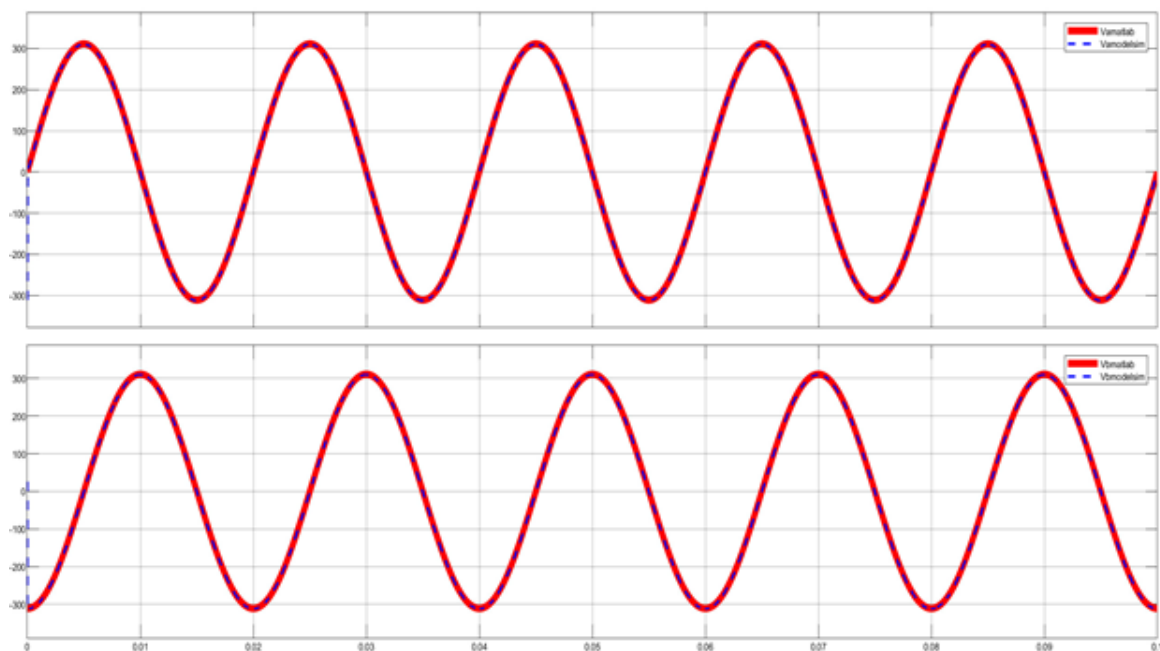
Hình 4.13: Kết quả mô phỏng bộ PI

## 2.2. Mô phỏng và kiểm nghiệm khối biến đổi Clark thuận



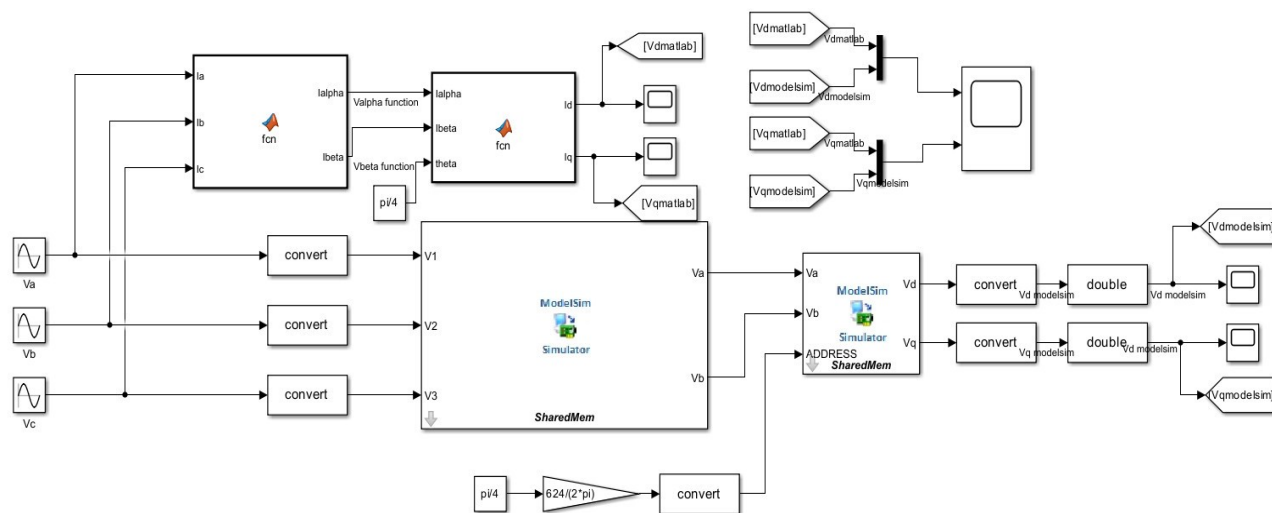
Hình 4.14: Mô phỏng Matlab/Simulink bộ biến đổi Clark thuận



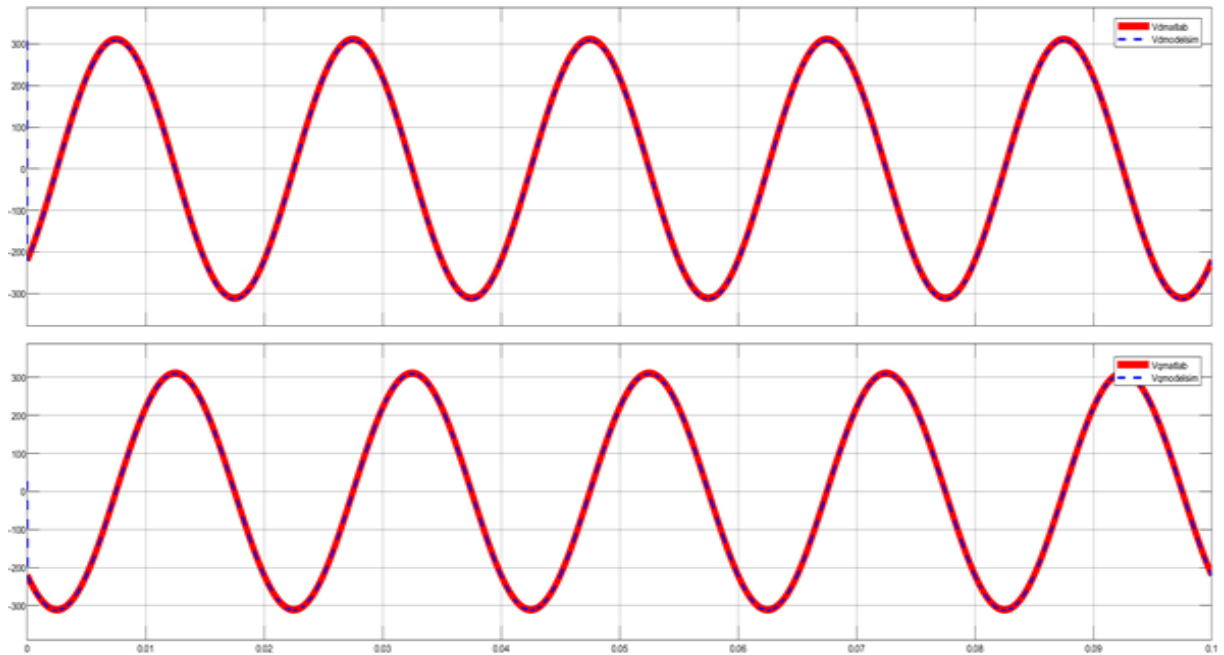


Hình 4.15: Kết quả mô phỏng Clark thuận

### 2.3. Mô phỏng và kiểm nghiệm khối biến đổi Clark, Park thuận

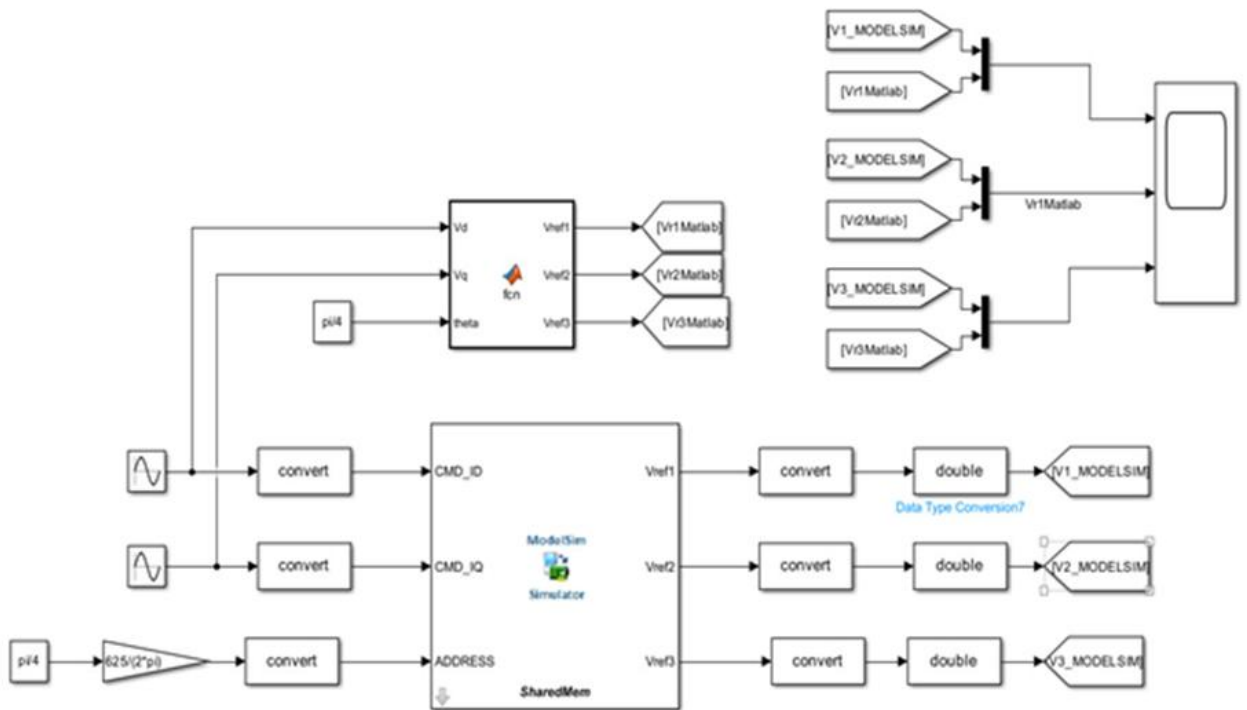


Hình 4.16: Mô phỏng Matlab/Simulink bộ biến đổi Clark, Park thuận

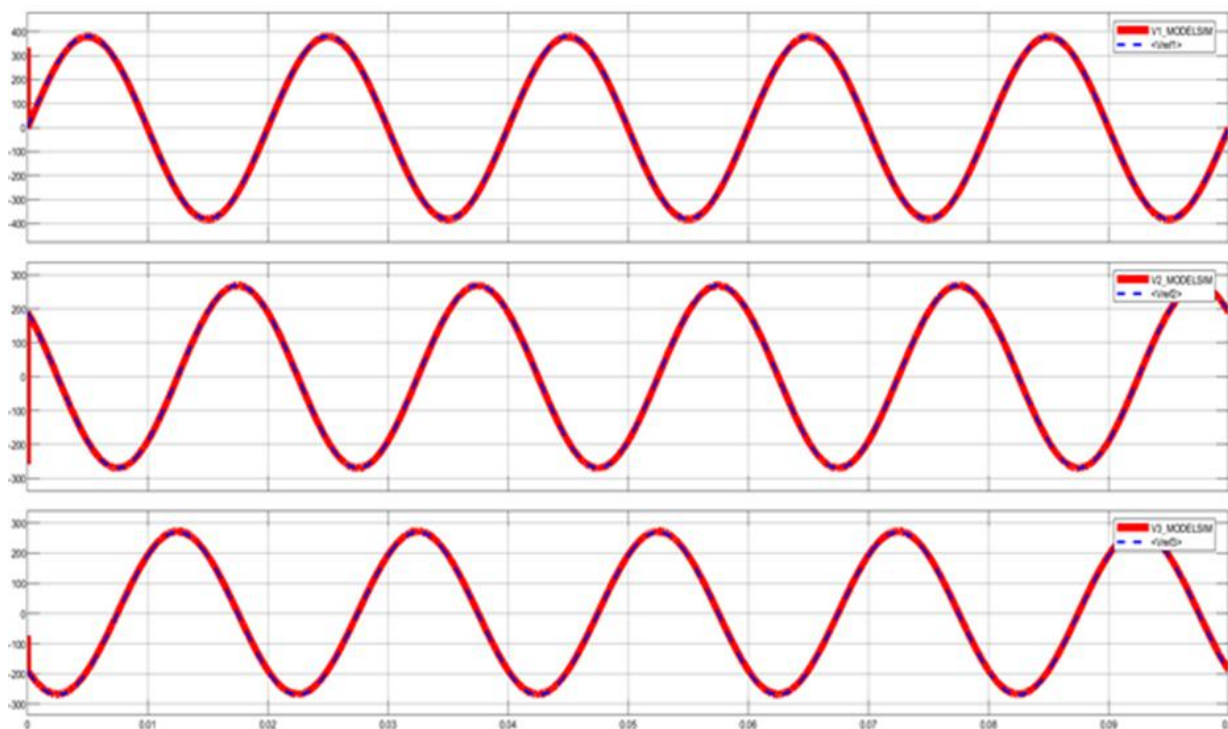


Hình 4.17: Kết quả mô phỏng Park, Clark thuận

#### 2.4. Mô phỏng và kiểm nghiệm khối Clark ngược sửa đổi – Park ngược

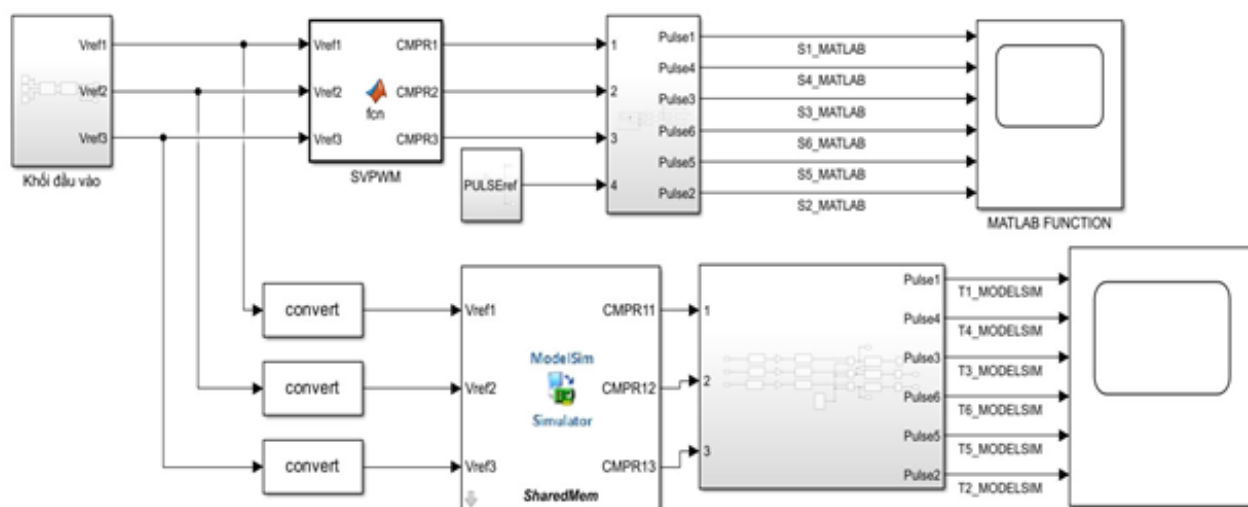


Hình 4.18: Mô phỏng Matlab/Simulink Clark ngược sửa đổi – Park ngược

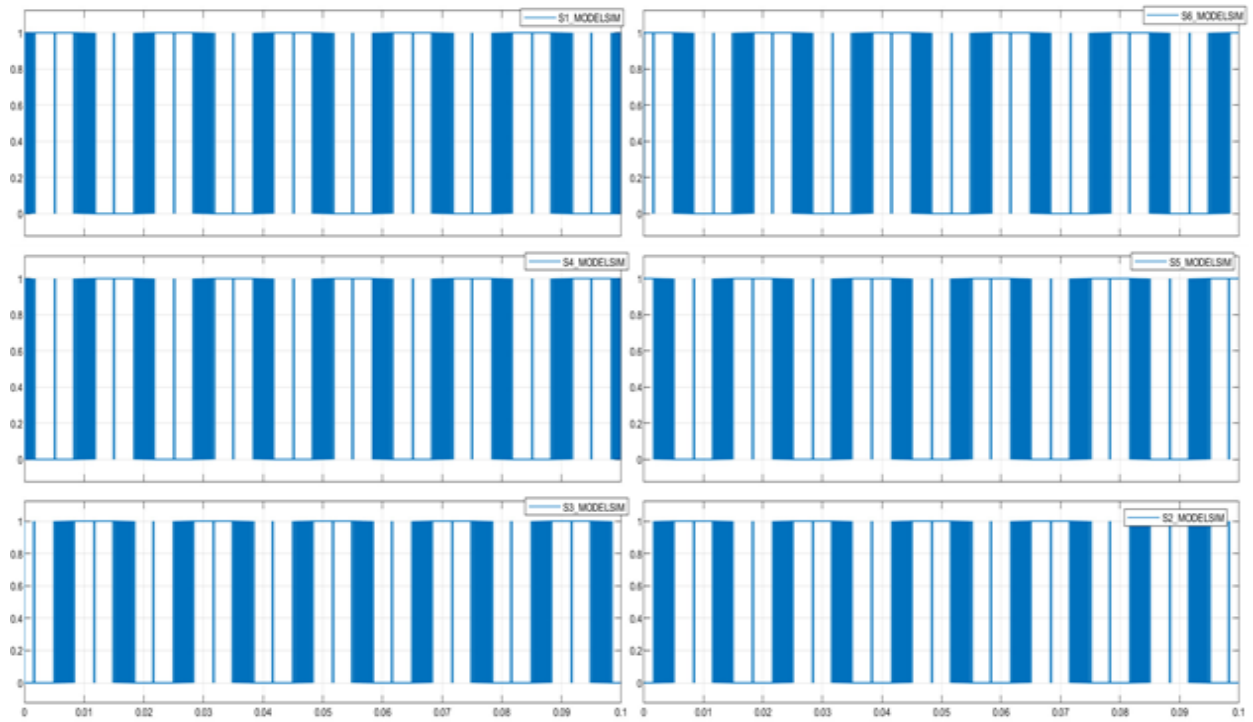


Hình 4.19: Kết quả mô phỏng Clark ngược sửa đổi – Park ngược

## 2.5. Mô phỏng và kiểm nghiệm thuật toán SVPWM



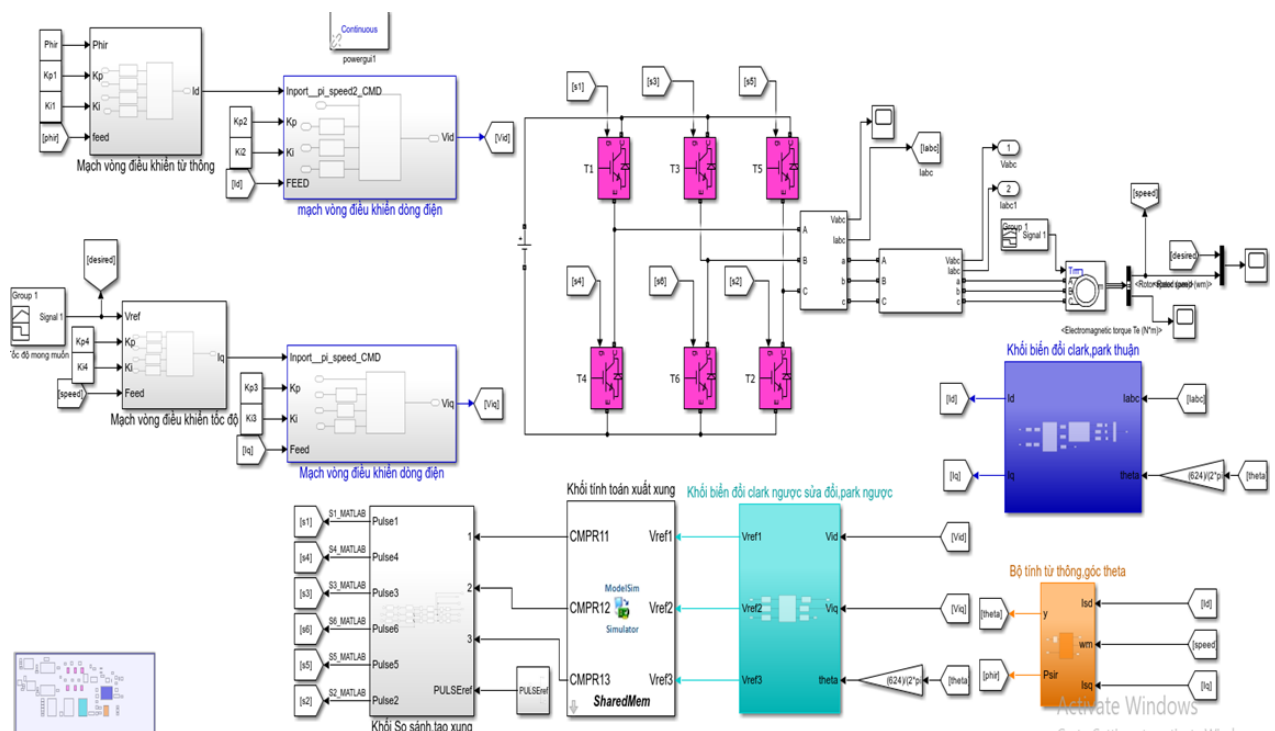
Hình 4.20: Mô phỏng SVPWM



Hình 4.21: Kết quả mô phỏng SVPWM

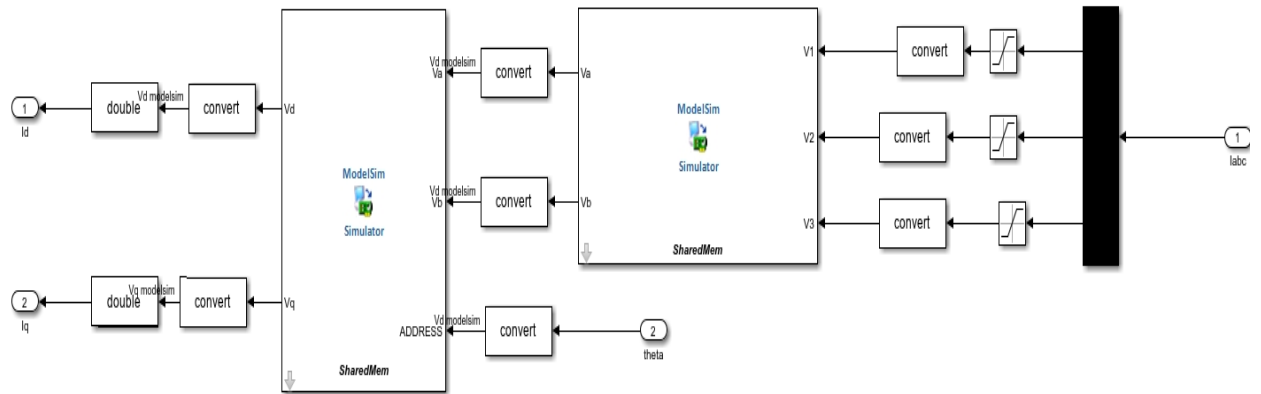
## 2.6. Mô phỏng sơ đồ toàn hệ thống

Sơ đồ mô phỏng



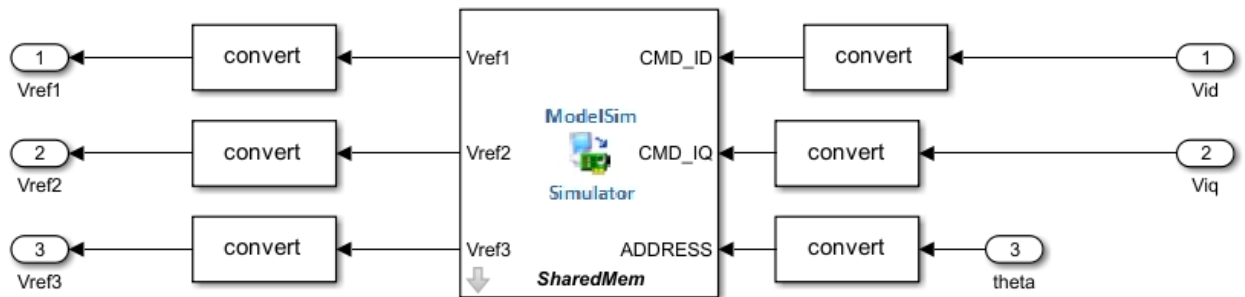
Hình 4.22: Sơ đồ mô phỏng

Bên trong khối biến đổi Clark, Park thuận



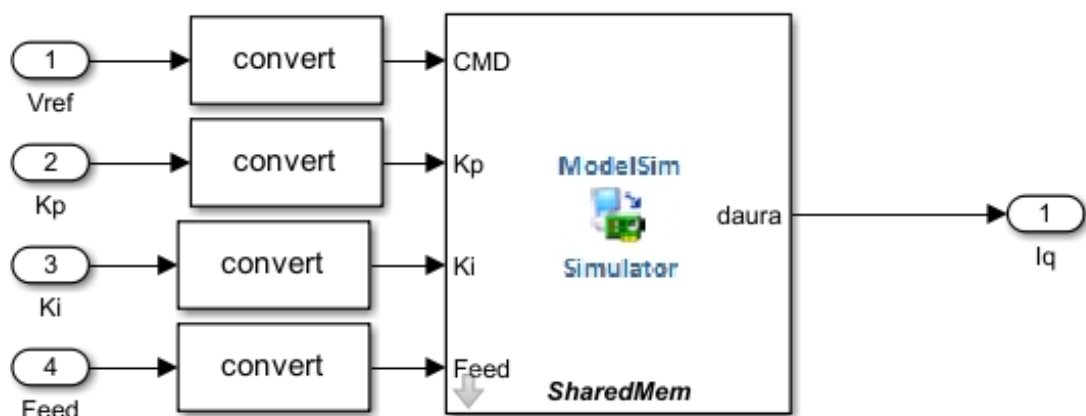
Hình 4.23: Sơ đồ khối mô phỏng

Bên trong khối mạch Biến đổi Clark ngược sửa đổi, Park ngược



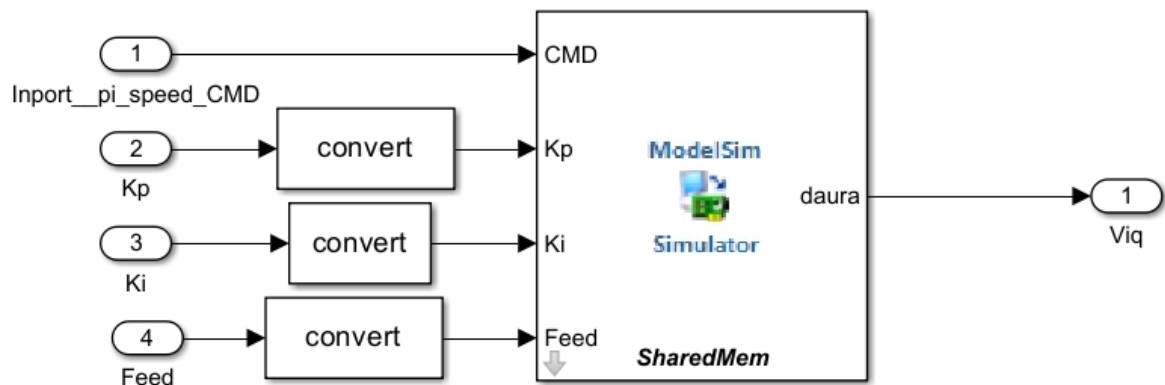
Hình 4.24: Clark ngược sửa đổi – Park ngược

Bên trong khối mạch vòng điều khiển tốc độ



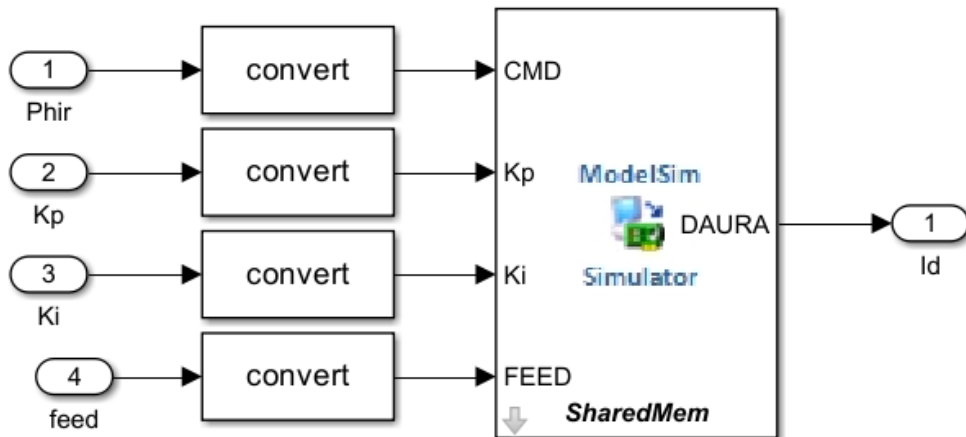
Hình 4.25: Khối mạch vòng điều khiển tốc độ

Bên trong khối mạch vòng điều khiển dòng điện  $I_q$



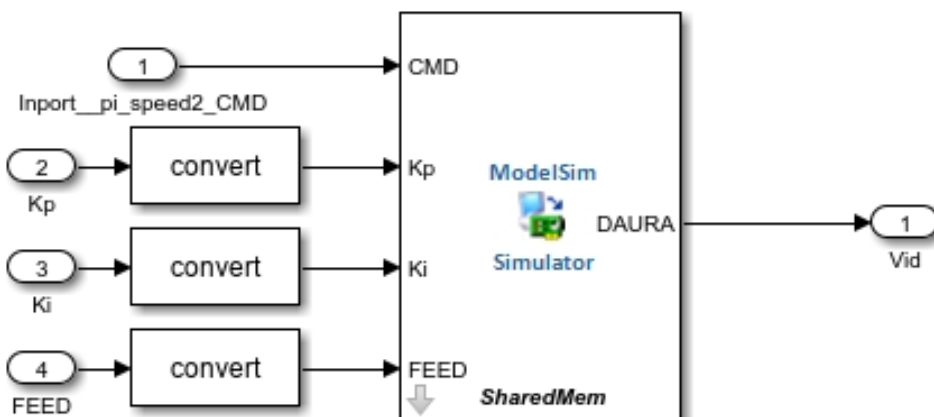
Hình 4.26: Khối mạch vòng điều khiển dòng điện  $I_q$

Bên trong khối điều khiển từ thông



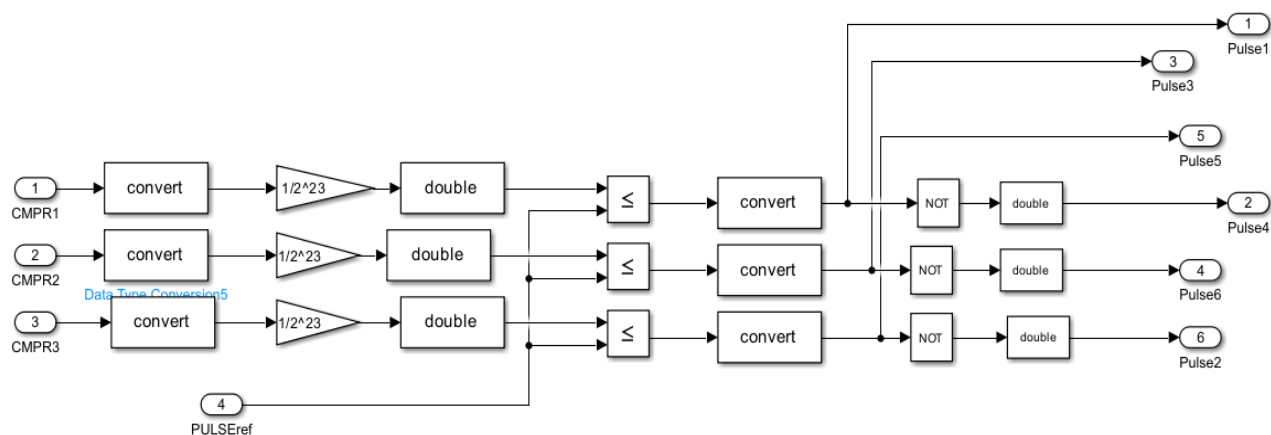
Hình 4.27: Khối điều khiển từ thông

Bên trong khối điều khiển dòng điện  $I_d$



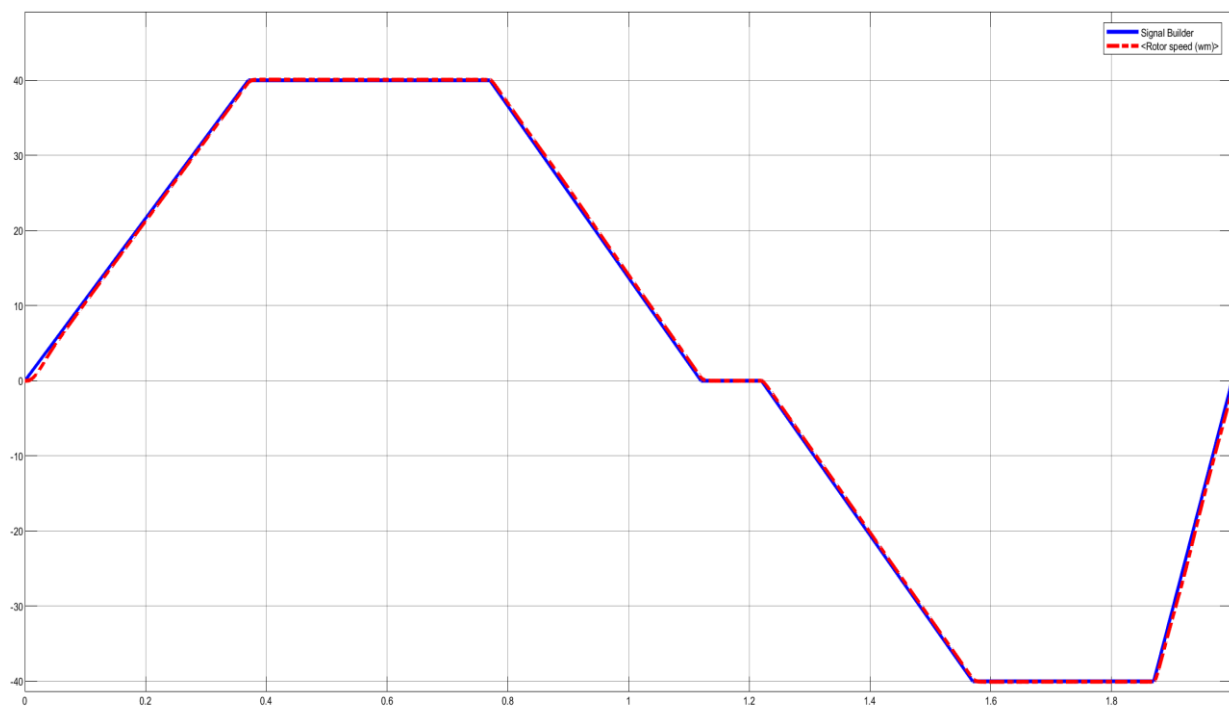
Hình 4.28: Khối điều khiển dòng điện  $I_d$

## Bên trong khối tạo xung

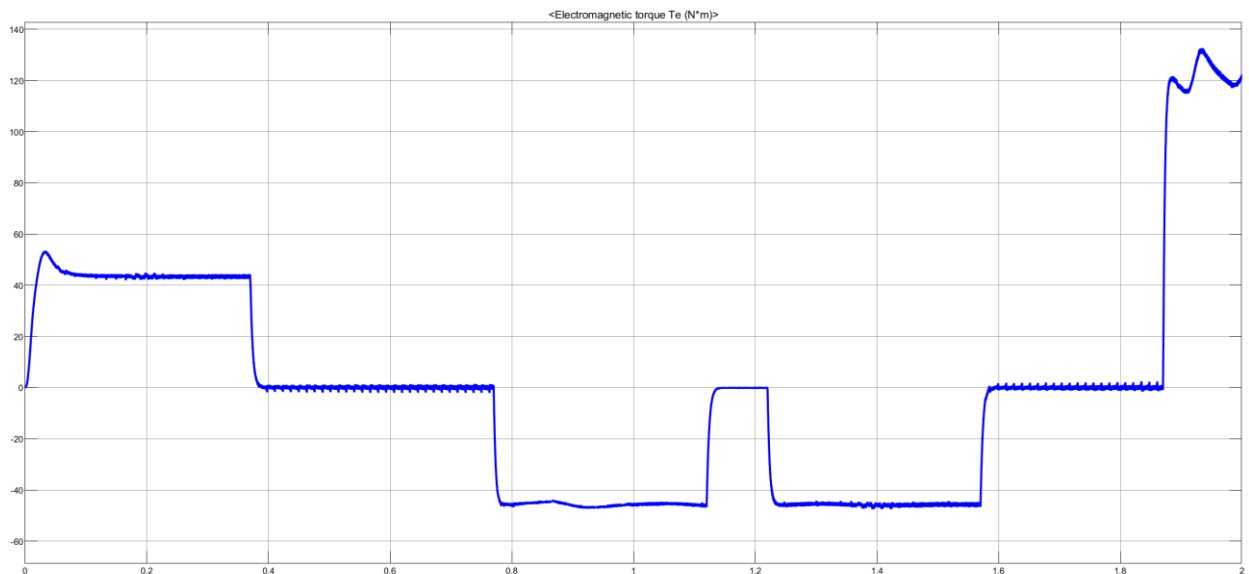


Hình 4.29: Khối tạo xung

## Kết quả mô phỏng



Hình 4.30: Kết quả mô phỏng tốc độ của động cơ



Hình 4.31: Kết quả mô phỏng momen động cơ

#### Nhận xét:

- Do thời gian mô phỏng Modelsim kết hợp Matlab/Simulink rất lâu nên nhóm em giảm thời gian chạy về còn 2s để mô phỏng, việc này dẫn đến momen tăng cao (do thay đổi tốc độ đột ngột dẫn đến dòng lớn).
- Từ đồ thị tốc độ của động cơ ta thấy sai số xác lập là  $0.4 = 1\% < 3\%$  nên điều này là chấp nhận được, sai số này xuất hiện một phần do thông số PI chưa được chuẩn xác, ngoài ra nó còn do việc sử dụng số bit 12Q để tính toán số thập phân còn hạn chế về sự chính xác, nhưng nhìn chung đáp ứng đầu ra của hệ thống vẫn có thể bám theo được giá trị tốc độ mong muốn.
- Việc sử dụng 12bit Q ban đầu ở mạch vòng tốc độ còn có 1 sự hạn chế là vì số bit không đủ lớn nên nếu chọn số bit Q quá nhỏ nó sẽ làm chậm đáp ứng của hệ thống.
- Từ đồ thị Momen của động cơ, ta thấy vẫn còn nhiều sự dao động lớn, nếu hoạt động lâu dài có thể làm hư hại động cơ, tuy nhiên vẫn đáp ứng được yêu cầu của tải về momen.

### 3. Kết luận chung

#### Ưu điểm:

- Hệ thống hoạt động tương đối tốt với giá trị trung bình của các đại lượng tốc độ, momen, dòng điện và điện áp theo đúng với giá trị tính toán trên lý thuyết.
- Thời gian xác lập của tốc độ đạt  $t = 0.04s$ . Cho kết quả phản ứng nhanh, tốc độ bám tốt và thời gian đáp ứng nhỏ, sử dụng tốt cho các đối tượng động cơ hoạt động ở vùng dưới tốc độ định mức và cận 0.



- Khâu lọc chưa hoạt động hiệu quả nên kết quả đầu ra của các đại lượng chưa hoàn toàn phẳng.
- Tại các thời điểm momen tải thay đổi, vẫn còn độ quá điều chỉnh nhỏ ở tốc độ động cơ.

**Nhược điểm:**

- Do sử dụng thuật toán điều chế vector không gian nên khối lượng tính toán lớn, gây ra thời gian trễ lớn cho quá trình tính toán.
- Sử dụng thuật toán điều chế vector không gian dễ gây ra hệ số đập mạch lớn, gây quá trình tỏa nhiệt và tổn hao lớn cho thiết bị.

## **TÀI LIỆU THAM KHẢO**

1. <https://www.slideshare.net/slideshow/iu-khin-tc-ng-c-khng-ng-b-3-pha-dng-b-iu-khin-pid-m-lai/251262223>
2. <https://www.slideshare.net/slideshow/iu-khin-thit-b-in-cng-nghip-trn-quang-th-nguyn-vinh-quanpdf/254162813>
3. [https://www.youtube.com/watch?v=sKIKnOvUmxI&list=PLOLMLP7vpGRj5\\_1N-2cXQRtbzkxiXeQHZ&index=7](https://www.youtube.com/watch?v=sKIKnOvUmxI&list=PLOLMLP7vpGRj5_1N-2cXQRtbzkxiXeQHZ&index=7)
4. [https://www.youtube.com/watch?v=EravApASDuk&list=PLOLMLP7vpGRj5\\_1N-2cXQRtbzkxiXeQHZ&index=8](https://www.youtube.com/watch?v=EravApASDuk&list=PLOLMLP7vpGRj5_1N-2cXQRtbzkxiXeQHZ&index=8)
5. [https://www.youtube.com/watch?v=06XVqbVk2pI&list=PLOLMLP7vpGRj5\\_1N-2cXQRtbzkxiXeQHZ&index=9](https://www.youtube.com/watch?v=06XVqbVk2pI&list=PLOLMLP7vpGRj5_1N-2cXQRtbzkxiXeQHZ&index=9)
6. <https://ieeexplore.ieee.org/document/5955638>