

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA CƠ KHÍ
BỘ MÔN CƠ ĐIỆN TỬ



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC
PHÂN TÍCH VÀ THIẾT KẾ
HỆ THỐNG ROBOT CHƠI CỜ TƯỚNG

SVTH: PHƯƠNG TÂN THÀNH – 1811226
NGUYỄN QUANG HÙNG – 1810967
GVHD: TS. PHẠM CÔNG BẰNG

TP. HỒ CHÍ MINH, 2023

NHIỆM VỤ LUẬN VĂN TỐT NGHIỆP*(Chú ý: sinh viên phải dán tờ này vào trang thứ nhất của bản thuyết minh)*

HỌ VÀ TÊN: Phương Tân Thành
Nguyễn Quang Hưng

MSSV: 1811226
1810967

NGÀNH: Cơ điện tử

LỚP: VP18CDT

1. Đầu đề luận văn:

Phân tích và thiết kế hệ thống robot chơi cờ tướng
(Analysis and design of a Chinese chess system)

Số liệu:

- Mô hình robot chơi cờ tướng hiện có trên lab C1

2. Nhiệm vụ:

- Tổng quan về robot chơi cờ
- Đánh giá thực nghiệm chức năng chơi cờ với người dùng

Phương Tân Thành

- Nhận diện quân cờ và vị trí quân trên bàn cờ
- Phân tích và xây dựng thuật toán chơi cờ

Nguyễn Quang Hưng

- Tìm hiểu và hoàn thiện sự kết nối của các thành phần trong hệ thống
- Thiết lập lưu đồ giải thuật điều khiển để thực hiện các chuyển động theo yêu cầu
- Số bản vẽ dự kiến: 5 , gồm:
 - 1 bản vẽ A0, về: kết cấu robot
 - 1 bản vẽ A0, về: sơ đồ kết nối thiết bị điện
 - 2 bản vẽ A0, về: lưu đồ giải thuật
 - 1 bản vẽ A0, về: Phương án

3. Ngày giao nhiệm vụ luận văn: 30 / 01 / 2023

4. Ngày hoàn thành nhiệm vụ: 12 / 05 / 2023

5. Họ và tên người hướng dẫn:

- Phạm Công Bằng , BM Cơ điện tử

Phần hướng dẫn:

100%

Nội dung và yêu cầu LVTN đã được thông qua Bộ môn.

Ngày 30 tháng 01 năm 2023

CHỦ NHIỆM BỘ MÔN
(Ký và ghi rõ họ tên)

NGƯỜI HƯỚNG DẪN CHÍNH
(Ký và ghi rõ họ tên)



Phạm Công Bằng

LỜI CẢM ƠN

Đầu tiên, chúng tôi – những người thực hiện luận văn này – xin bày tỏ lòng biết ơn chân thành và sâu sắc đến TS. Phạm Công Bằng vì sự quan tâm, hỗ trợ và những hướng dẫn tận tình của thầy trong suốt quá trình thực hiện luận văn tốt nghiệp này. Trong suốt thời gian hướng dẫn chúng tôi thực hiện luận văn, thầy không chỉ truyền đạt những kiến thức, kinh nghiệm mà đào tạo, định hướng cho chúng tôi tác phong và thái độ làm việc đúng đắn. Đó là bài học thiết thực với chúng tôi không chỉ với việc thực hiện luận văn tốt nghiệp mà còn trong quá trình công tác sau này.

Bên cạnh đó, chúng tôi cũng muốn thể hiện sự trân trọng đặc biệt với những giúp đỡ, tư vấn, hỗ trợ từ các anh Nguyễn Văn An và anh Đỗ Thành Long, các anh đã tận tình chỉ bảo, giải đáp những thắc mắc và cho chúng tôi nhiều lời khuyên bổ ích.

Chúng tôi cũng xin cảm ơn quý thầy/cô Bộ môn Cơ điện tử - Khoa Cơ khí, cùng quý thầy/cô Chương trình Đào tạo Kỹ sư Chất lượng cao Việt – Pháp, Trường Đại học Bách Khoa – Đại học Quốc gia Thành phố Hồ Chí Minh đã tạo những điều kiện thuận lợi nhất cho chúng tôi trong 5 năm học tập tại đây.

Cuối cùng nhưng không kém phần quan trọng, dù cho bao nhiêu khó khăn vất vả, chính những quan tâm, hỗ trợ của gia đình và bạn bè chúng tôi là những nguồn động viên lớn nhất để chúng tôi hoàn thành luận văn này. Xin chân thành cảm ơn tất cả.

Trân trọng.

TÓM TẮT LUẬN VĂN

Luận văn trình bày về việc nghiên cứu, xây dựng, thiết kế giải thuật điều khiển cho hệ thống robot chơi cờ tướng trên cơ sở khôi phục lại phần cứng robot này ở phòng Phòng Thí nghiệm Điều khiển và Tự động hóa, P202 – C1, Trường Đại học Bách Khoa – Đại học Quốc gia Thành phố Hồ Chí Minh. Mục tiêu luận văn hướng đến kết quả là robot chơi cờ có thể nhận diện quân cờ và tự xếp bàn cờ, có thể nhận diện và đánh giá nước cờ của người chơi, từ đó thực hiện các nước đi phản hồi phù hợp theo các cấp độ chơi từ dễ đến khó.

DANH MỤC HÌNH ẢNH

| | |
|--|----|
| Hình 1.1: Một trận đấu tại Đại hội thể thao Đông Nam Á - bộ môn Cờ tướng. | 13 |
| Hình 1.2: Sự phổ biến của bộ môn cờ tướng với những người cao tuổi. | 14 |
| Hình 1.3: Việc tiếp xúc với các thiết bị công nghệ quá sớm của trẻ em. | 14 |
| Hình 1.4: Robot đánh cờ với giao diện thân thiện với người dùng. | 14 |
| Hình 1.5: Robot của tác giả NG Pui Lam (Hong Kong Metropolitan University). | 15 |
| Hình 1.6: Robot của nhóm tác giả Xu Jiaxuan, Gao Rui và Li Yongheng (Xi'an University of Posts and Telecommunications)..... | 16 |
| Hình 1.7: SenseRobot của công ty SenseTime..... | 16 |
| Hình 1.8: Robot của nhóm tác giả Vũ Văn Khắc và Lê Bạch Quý Cường (Trường Đại học Bách Khoa thành phố Hồ Chí Minh)..... | 17 |
| Hình 1.9: Robot của tác giả Lê Phương Trà và Nguyễn Thành Việt (Trường Đại học Bách Khoa thành phố Hồ Chí Minh)..... | 18 |
| Hình 1.10: Hiện trạng robot chơi cờ tướng tại bộ môn Cơ điện tử, khoa Cơ khí, trường Đại học Bách khoa thành phố Hồ Chí Minh. | 19 |
| Hình 1.11: Webcam Rapoo C200..... | 21 |
| Hình 1.12: Bàn cờ tướng. | 21 |
| Hình 1.13: Quân cờ tướng. | 21 |
| Hình 1.14: Giới hạn không gian đặt quân cờ trước khi xếp. | 22 |
| Hình 2.1: Sơ đồ nguyên lý robot chơi cờ tướng. | 23 |
| Hình 2.2: Chuyển vị của đầu công tác khi M_1 đứng yên, M_2 quay theo chiều âm. | 24 |
| Hình 2.3: Cụm khung đế..... | 26 |
| Hình 2.4: Cụm gá đặt camera. | 26 |
| Hình 2.5: Cụm pulley chủ động. | 27 |
| Hình 2.6: Cụm pulley dẫn hướng. | 27 |
| Hình 2.7: Cụm pulley bị động. | 28 |
| Hình 2.8: Cụm cơ cầu chấp hành. | 28 |
| Hình 2.9: Sơ đồ tín hiệu của các thành phần trong hệ thống điện..... | 29 |
| Hình 2.10: Camera Rapoo C200. | 29 |

| | |
|---|----|
| Hình 2.11: Cảm biến từ SMC D-A93..... | 30 |
| Hình 2.12: Vị trí đặt các cảm biến tiệm cận trên bàn máy..... | 30 |
| Hình 2.13: Cảm biến tiệm cận LJ12A3-4-Z/BX NPN. | 31 |
| Hình 2.14: Các van solenoid được dùng trong hệ thống..... | 31 |
| Hình 2.15: Relay kính OMRON MY4N. | 32 |
| Hình 2.16: Sơ đồ đấu dây cảm biến tiệm cận với relay kính. | 32 |
| Hình 2.17: Driver SGDV 1R6A01. | 33 |
| Hình 2.18: Sơ đồ đấu dây driver..... | 35 |
| Hình 2.19: Động cơ AC Servo SGMPS-02ACA21. | 35 |
| Hình 2.20: Sơ đồ đấu dây động cơ. | 36 |
| Hình 2.21: Xy lanh khí nén MCE987845..... | 36 |
| Hình 2.22: Board PLC FX3U LE3U 24MT. | 37 |
| Hình 2.23: Sơ đồ đấu dây PLC..... | 38 |
| Hình 2.24: Máy tính Gigabyte U4. | 39 |
| Hình 2.25: Bộ nguồn xung 24V DC..... | 40 |
| Hình 2.26: Các loại đèn báo dùng trong hệ thống điện. | 40 |
| Hình 2.27: Cầu dao điện tự động và công tắc điện. | 41 |
| Hình 2.28: Các loại nút nhấn trong hệ thống điện..... | 41 |
| Hình 2.29: Sơ đồ đấu dây với nguồn 220V AC. | 41 |
| Hình 2.30: Sơ đồ đấu dây với nguồn 24V DC. | 42 |
| Hình 2.31: Robot chơi cờ sau khi được khôi phục..... | 43 |
| Hình 3.1: Ma trận 2 chiều của điểm ảnh. | 44 |
| Hình 3.2: Mô hình biểu diễn ảnh nhị phân..... | 44 |
| Hình 3.3: Mô hình biểu diễn ảnh xám. | 45 |
| Hình 3.4: Mô hình màu RGB. | 45 |
| Hình 3.5: Mô hình ảnh RGB. | 46 |
| Hình 3.6: Các tham số của phương trình đường tròn. | 46 |
| Hình 3.7: Chuyển về không gian hough space. | 46 |
| Hình 3.8: Mô hình camera lõi kim. | 47 |
| Hình 3.9: Ảnh bàn cờ bị biến dạng khi chụp qua camera. | 48 |

| | |
|--|----|
| Hình 3.10: Lưu đồ quá trình hiệu chuẩn camera..... | 48 |
| Hình 3.11: Hiệu chuẩn camera..... | 49 |
| Hình 3.12: Kiến trúc mạng neural..... | 49 |
| Hình 3.13: Minh họa kiến trúc một mạng neural tích chập..... | 51 |
| Hình 3.14: Áp dụng phép tích chập trên ảnh..... | 51 |
| Hình 3.15: Minh họa kiến trúc mạng kết nối đầy đủ..... | 52 |
| Hình 3.16: Bộ lọc di chuyển S đơn vị sau mỗi lần thực hiện phép chập..... | 53 |
| Hình 3.17: Quá trình chụp và hiệu chỉnh hình ảnh bàn cờ..... | 56 |
| Hình 3.18: Phát hiện quân cờ thông qua đường tròn khắc trên nó..... | 56 |
| Hình 3.19: Các vị trí đánh giá độ lệch tọa độ quân cờ..... | 58 |
| Hình 3.20: Lưu đồ quá trình xác định tên từng quân cờ đã nhận dạng..... | 61 |
| Hình 3.21: Ảnh quân cờ sau khi được cắt ra từ bàn cờ và biến đổi..... | 62 |
| Hình 3.22: Lưu đồ quá trình lấy dữ liệu để huấn luyện mô hình..... | 63 |
| Hình 3.23: Dữ liệu quân cờ được sinh ra qua quá trình xoay..... | 64 |
| Hình 3.24: Minh họa cấu trúc mạng VGG..... | 65 |
| Hình 3.25: Độ chính xác và giá trị mất mát của mô hình trên tập huấn luyện và tập kiểm tra qua các epochs..... | 67 |
| Hình 3.26: Hình ảnh quân cờ qua biến đổi ở các ngưỡng khác nhau..... | 68 |
| Hình 4.1: Các bước để đầu công tác thực hiện hành động đi quân và ăn quân..... | 70 |
| Hình 4.2: Lưu đồ giải thuật điều khiển đầu công tác..... | 71 |
| Hình 4.3: Mô hình OPC kết nối máy tính với các thiết bị công nghiệp. [18] | 74 |
| Hình 4.4: Giao diện KEPServerEX khi tạo OPC Server cho PLC Mitsubishi FX3U. . | 74 |
| Hình 4.5: Minh họa cho việc tiến hành thực nghiệm 1..... | 75 |
| Hình 4.6: Minh họa cho việc tiến hành thực nghiệm 2..... | 76 |
| Hình 5.1: Lưu đồ hoạt động chơi cờ của robot..... | 79 |
| Hình 5.2: Quân cờ nằm che lấp vị trí cần xếp vào của quân cờ khác. | 80 |
| Hình 5.3: Giới hạn không gian đặt quân cờ trước khi xếp. | 80 |
| Hình 5.4: Lưu đồ quá trình xếp cờ. | 81 |
| Hình 5.5: Mã hóa bàn cờ thành chuỗi FEN. | 84 |
| Hình 5.6: Mô tả trạng thái bàn cờ dưới dạng ma trận. | 85 |

| | |
|---|-----|
| Hình 5.7: Hệ tọa độ bàn cờ..... | 86 |
| Hình 5.8: Các quân cờ nằm không đúng vị trí theo luật (đánh dấu \times). | 87 |
| Hình 5.9: Sơ đồ tổng quan giải thuật chơi cờ..... | 89 |
| Hình 5.10: Sơ đồ minh họa giải thuật Minimax..... | 91 |
| Hình 5.11: Sơ đồ mô tả giải thuật cắt tỉa Alpha – Beta..... | 93 |
| Hình 5.12: Vị trí ban đầu của các quân cờ. | 97 |
| Hình 5.13: Giá trị theo vị trí của các quân cờ..... | 98 |
| Hình 5.14: Bàn cờ bị lặp trạng thái. | 100 |
| Hình 5.15: Thé cờ (1) so sánh nước đi giữa các chế độ chơi. | 103 |
| Hình 5.16: Thé cờ (2) so sánh nước đi giữa các chế độ chơi. | 103 |

DANH MỤC BẢNG BIỂU

| | |
|--|----|
| Bảng 2.1: Các trường hợp di chuyển của đầu công tác..... | 25 |
| Bảng 2.2: Thông số camera Rapoo C200..... | 29 |
| Bảng 2.3: Thông số cảm biến từ SMC D-A93..... | 30 |
| Bảng 2.4: Thông số cảm biến tiệm cận LJ12A3-4-Z/BX NPN..... | 31 |
| Bảng 2.5: Thông số của van solenoid STNC TG2521-08 và STNC TG2321-08..... | 32 |
| Bảng 2.6: Thông số driver SGDV 1R6A01..... | 33 |
| Bảng 2.7: Logic đóng ngắt chân P-OT và N-OT của driver..... | 33 |
| Bảng 2.8: Ký hiệu và chức năng các chân driver..... | 34 |
| Bảng 2.9: Thông số động cơ AC Servo SGMPS-02ACA21..... | 35 |
| Bảng 2.10: Thông số xy lanh MCE987845..... | 36 |
| Bảng 2.11: Thông số PLC FX3U LE3U 24MT..... | 37 |
| Bảng 2.12: Ký hiệu và chức năng các chân PLC..... | 38 |
| Bảng 2.13: Thông số nguồn xung 24V DC..... | 40 |
| Bảng 3.1: Các hàm kích hoạt thường được sử dụng..... | 50 |
| Bảng 3.2: Các kiểu pooling thường được sử dụng..... | 52 |
| Bảng 3.3: Các kiểu 0 – đếm thường được sử dụng..... | 53 |
| Bảng 3.4: Các tham số hàm Houghcircle() | 57 |
| Bảng 3.5: Bảng so sánh độ lệch giữa tọa độ quân cờ thực tế và tọa độ quân cờ tính toán được trên ảnh..... | 58 |
| Bảng 3.6: lựa chọn tham số thresh trong một số môi trường..... | 62 |
| Bảng 3.7: Kết quả nhận diện quân cờ trong các điều kiện ánh sáng khác nhau..... | 68 |
| Bảng 4.1: Kết quả thực nghiệm 1..... | 75 |
| Bảng 4.2: Kết quả thực nghiệm 2..... | 76 |
| Bảng 5.1: Tỉ lệ xếp đúng và thời gian xếp trong các trường hợp bàn cờ khác nhau..... | 82 |
| Bảng 5.2: Quy ước Mã hóa các quân cờ..... | 84 |
| Bảng 5.3: Ví dụ so sánh các ma trận trạng thái để phát hiện nước đi quân cờ..... | 87 |
| Bảng 5.4: Đánh giá chức năng xác định nước đi trong các trường hợp bất kì..... | 88 |
| Bảng 5.5: Số nước đi và thời gian tính toán trung bình giải thuật Minimax..... | 92 |

| | |
|---|-----|
| Bảng 5.6: Số nút phải xét khi dùng thuật toán Minimax và Alpha – Beta..... | 94 |
| Bảng 5.7: Giá trị mỗi quân cờ. | 95 |
| Bảng 5.8: Thời gian giới hạn của giải thuật chơi cờ với các mức độ khó khác nhau. | 101 |
| Bảng 5.9: Phân chia giải thuật chơi cờ ở 3 cấp độ khác nhau. | 102 |
| Bảng 5.10: Nước đi phản hồi thế cờ hình 5.15 của phần mềm theo các chế độ. | 102 |
| Bảng 5.11: Nước đi phản hồi thế cờ hình 5.16 của phần mềm theo các chế độ. | 104 |

MỤC LỤC

| | |
|--|----|
| CHƯƠNG 1. TỔNG QUAN | 13 |
| 1.1. Giới thiệu chung | 13 |
| 1.2. Tình hình nghiên cứu trong và ngoài nước | 15 |
| 1.2.1. Tình hình nghiên cứu ngoài nước..... | 15 |
| 1.2.2. Tình hình nghiên cứu trong nước | 17 |
| 1.3. Mục tiêu – Nhiệm vụ - Phạm vi đề tài..... | 18 |
| 1.3.1. Mục tiêu | 18 |
| 1.3.2. Nhiệm vụ | 20 |
| 1.3.3. Phạm vi đề tài | 21 |
| 1.4. Tổ chức luận văn | 22 |
| CHƯƠNG 2. TÌM HIỂU VÀ KHÔI PHỤC HỆ THỐNG ROBOT CHƠI CỜ TƯỚNG | 23 |
| 2.1. Tìm hiểu về phần cơ khí | 23 |
| 2.1.1. Sơ đồ nguyên lý của robot..... | 23 |
| 2.1.2. Phân tích động học robot..... | 24 |
| 2.1.3. Các cụm cơ khí của robot | 25 |
| 2.2. Tìm hiểu về phần điện | 29 |
| 2.2.1. Camera..... | 29 |
| 2.2.2. Cảm biến từ xy lanh..... | 30 |
| 2.2.3. Cảm biến tiệm cận | 30 |
| 2.2.4. Van điện từ khí nén | 31 |
| 2.2.5. Relay | 32 |
| 2.2.6. Driver động cơ..... | 33 |
| 2.2.7. Động cơ | 35 |

| | |
|--|-----------|
| 2.2.8. Xy lanh khí nén | 36 |
| 2.2.9. PLC | 37 |
| 2.2.10. Máy tính..... | 39 |
| 2.2.11. Nguồn | 39 |
| 2.2.12. Đèn báo..... | 40 |
| 2.2.13. Công tắc và nút bấm | 40 |
| 2.3. Kết quả..... | 42 |
| 2.4. Kết luận: | 43 |
| CHƯƠNG 3. XÁC ĐỊNH VỊ TRÍ VÀ TÊN CÁC QUÂN CỜ..... | 44 |
| 3.1. Các khái niệm | 44 |
| 3.1.1. Xử lý ảnh | 44 |
| 3.1.2. Mạng neural tích chập | 49 |
| 3.2. Chụp và hiệu chỉnh hình ảnh bàn cờ | 55 |
| 3.3. Xác định tọa độ các quân cờ trong hệ tọa độ bàn máy | 56 |
| 3.3.1. Sử dụng giải thuật Hough circle để thu được tọa độ quân cờ | 57 |
| 3.3.2. Đánh giá hoạt động xác định tọa độ quân cờ | 57 |
| 3.4. Xác định tên của các quân cờ | 61 |
| 3.4.1. Chuẩn bị dữ liệu để huấn luyện mô hình..... | 62 |
| 3.4.2. Xác định và xây dựng mô hình..... | 64 |
| 3.4.3. Huấn luyện mô hình và đánh giá | 66 |
| 3.4.4. Đánh giá hoạt động của chương trình trong một số môi trường khác nhau | 67 |
| 3.5. Kết luận..... | 69 |
| CHƯƠNG 4. ĐIỀU KHIỂN ĐẦU CÔNG TÁC | 70 |
| 4.1. Giải thuật điều khiển đầu công tác | 70 |

| | |
|---|-----|
| 4.1.1. Các trạng thái của quá trình về Home | 72 |
| 4.1.2. Các trạng thái của quá trình đi quân..... | 72 |
| 4.1.3. Các trạng thái của quá trình ăn quân | 73 |
| 4.2. Giao tiếp giữa PLC và máy tính | 73 |
| 4.2.1. Chuẩn giao tiếp OPC | 73 |
| 4.2.2. Phần mềm KEPServerEX..... | 74 |
| 4.3. Kết quả..... | 75 |
| 4.4. Kết luận: | 77 |
| CHƯƠNG 5. PHÂN TÍCH VÀ XÂY DỰNG GIẢI THUẬT CHƠI CỜ CHO ROBOT | |
| | 78 |
| 5.1. Sắp xếp quân cờ..... | 79 |
| 5.1.1. Giải thuật sắp xếp quân cờ | 79 |
| 5.1.2. Đánh giá chức năng sắp xếp quân cờ | 82 |
| 5.2. Xác định nước đi của người chơi | 83 |
| 5.2.1. Mô tả trạng thái bàn cờ..... | 84 |
| 5.2.2. Kiểm tra tính hợp lệ của bàn cờ..... | 85 |
| 5.2.3. Phát hiện và kiểm tra nước đi trên bàn cờ | 87 |
| 5.2.4. Đánh giá chức năng xác định nước đi của người chơi | 88 |
| 5.3. Giải thuật chơi cờ của robot | 89 |
| 5.3.1. Giải thuật tìm kiếm..... | 90 |
| 5.3.2. Hàm lượng giá | 94 |
| 5.3.3. Cắt nhánh khi mất quân tướng | 98 |
| 5.3.4. Hiện tượng horizon effect và giải thuật quiescence search | 99 |
| 5.3.5. Cải tiến Alpha – Beta bằng transposition table | 99 |
| 5.3.6. Kiểm soát thời gian duyệt bằng giải thuật iterative depth first search | 101 |

| | |
|--|-----|
| 5.3.7. Giải thuật Zobrist Hashing | 101 |
| 5.3.8. Cơ sở dữ liệu..... | 101 |
| 5.3.9. Phân chia các cấp độ chơi cờ..... | 102 |
| 5.4. Kết luận..... | 105 |
| CHƯƠNG 6. TỔNG KẾT | 106 |
| 6.1. Kết quả..... | 106 |
| 6.2. Hạn chế | 107 |
| 6.3. Hướng phát triển..... | 107 |

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu chung

Cờ tướng là một bộ môn thể thao trí tuệ phổ biến với sự quan tâm đông đảo, cùng lối chơi và thể thức thi đấu đa dạng, hấp dẫn. Loại cờ này phổ biến tại các nước như Trung Quốc, Singapore, Malaysia và Việt Nam. Hàng năm, rất nhiều các giải cờ tướng lớn nhỏ, chuyên nghiệp và không chuyên được tổ chức, thậm chí đây còn là bộ môn tranh tài ở một số đại hội thể thao [1] (hình 1.1), vì vậy nhu cầu thực hành và luyện tập là có. Rất nhiều những trò chơi cờ tướng trên máy tính và điện thoại thông minh được ra đời, phần nào thỏa mãn nhu cầu giải trí và rèn luyện. Trong những năm gần đây, với sự phát triển của học máy và trí tuệ nhân tạo, nhiều phần mềm chơi cờ đã chiến thắng cả những kiện tướng thế giới.



Hình 1.1: Một trận đấu tại Đại hội thể thao Đông Nam Á - bộ môn Cờ tướng.

Tuy nhiên, việc chơi cờ trên máy tính hay điện thoại thông minh vẫn cho thấy nhiều hạn chế trong việc tiếp cận và lan tỏa bộ môn cờ tướng đến nhiều đối tượng. Cụ thể, đối với những đối tượng người cao tuổi - đối tượng được cho là chiếm phần đông trong số những người chơi bộ môn này [2] (hình 1.2), thì việc chơi cờ trên máy tính hay điện thoại không đem lại cảm giác “thật” và việc sử dụng các thiết bị điện thoại, máy tính còn tương đối khó khăn. Bên cạnh đối tượng người cao tuổi, trẻ em cũng là một đối tượng có thể rèn luyện trí thông minh, khả năng suy luận, phán đoán thông qua cờ tướng. Nhưng việc chơi chờ bằng các phần mềm đồng nghĩa với việc cho trẻ tiếp xúc với các thiết bị điện thoại hay máy tính từ sớm [3] (hình 1.3), đây là một mối lo mà phụ huynh muốn tránh.



Hình 1.2: Sự phổ biến của bộ môn cờ tướng với những người cao tuổi.



Hình 1.3: Việc tiếp xúc với các thiết bị công nghệ quá sớm của trẻ em.

Từ những hạn chế trên, robot đánh cờ đã ra đời như một giải pháp hiệu quả. Robot đánh cờ là một robot được lập trình bằng các thuật toán để có thể điều khiển các quân cờ thực tế và chơi cờ một cách tự động. Một số còn được tích hợp giao diện hướng dẫn, hỗ trợ chơi thân thiện như robot trong hình 1.4 [4].



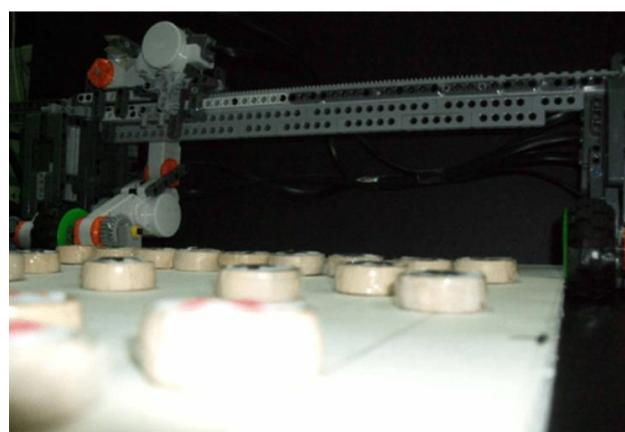
Hình 1.4: Robot đánh cờ với giao diện thân thiện với người dùng.

Robot chơi cờ tướng là một giải pháp giúp những người cao tuổi có thể duy trì sự minh mẫn bằng việc chơi cờ tướng mà không phải quan ngại vấn đề không có người chơi cùng hay không biết sử dụng phần mềm chơi cờ. Đây còn là một cách tiếp cận thú vị với cờ tướng dành cho trẻ em, một cách tiếp cận trực tiếp không thông qua các thiết bị điện thoại hay máy tính.

1.2. Tình hình nghiên cứu trong và ngoài nước

1.2.1. Tình hình nghiên cứu ngoài nước

a. *Robot chơi cờ tướng từ Hong Kong Metropolitan University (2012)*

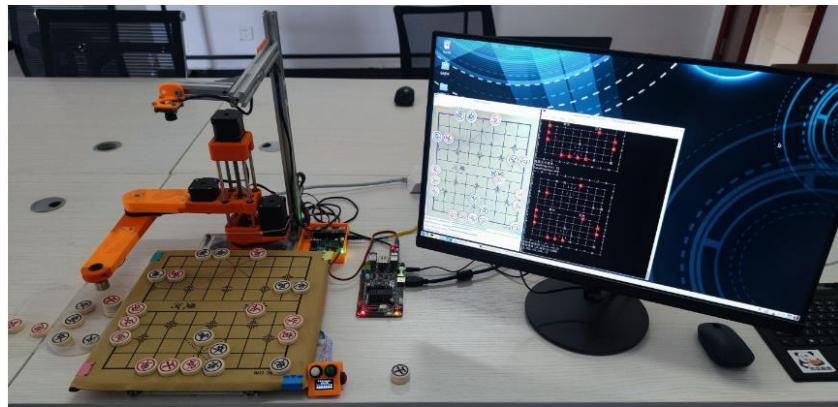


Hình 1.5: Robot của tác giả NG Pui Lam (Hong Kong Metropolitan University).

Robot trong hình 1.5 [5] là thành quả nghiên cứu và phát triển của tác giả NG Pui Lam. Phần cứng robot được xây dựng dựa vào bộ kit robot LEGO Mindstorms. Ngôn ngữ lập trình là Java và phần mềm sử dụng là leJOS. Đầu công tác sử dụng nam châm để hút quân cờ. Trong dự án này, robot sẽ không nhận diện quân cờ mà sẽ tính toán và đối chiếu giá trị RGB của mỗi ô cờ để từ đó nhận biết nước đi của người chơi và tính toán nước cờ đánh lại.

- **Ưu điểm:**
 - Việc xây dựng phần cứng dễ dàng nhờ sử dụng bộ kit có sẵn.
 - Thuật toán nhận diện nước đi đơn giản không yêu cầu nhiều về camera.
- **Nhược điểm:**
 - Độ cứng, vững không cao, hành trình đầu công tác chưa tối ưu.
 - Đầu công tác là nam châm nên không dùng được cho tất cả loại quân cờ
 - Chưa nhận diện được quân cờ, yêu cầu người chơi phải xếp cờ trước.

b. Robot chơi cờ tướng từ Xi'an University of Posts and Telecommunications (2022)

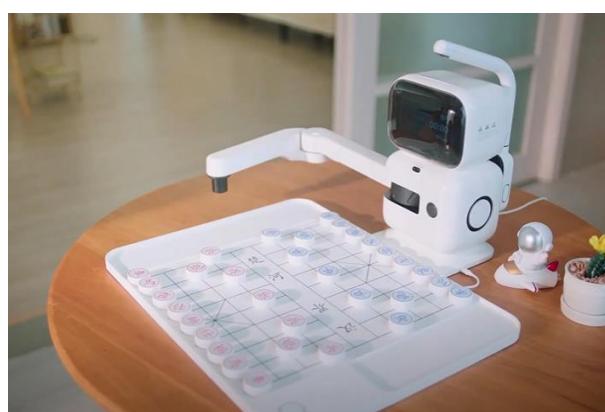


Hình 1.6: Robot của nhóm tác giả Xu Jiaxuan, Gao Rui và Li Yongheng (Xi'an University of Posts and Telecommunications).

Dự án của nhóm tác giả Xu Jiaxuan, Gao Rui và Li Yongheng [6] trong hình 1.6 đã thắng giải nhất cuộc thi “Anmou Technology Enterprise Proposition”. Họ sử dụng cánh tay robot SCARA với đầu công tác nam châm, kit phát triển là EAIDK-610. Thuật toán chơi cờ “Elephantfish Chess Search” ứng dụng trí tuệ nhân tạo. Ngoài ra, đi kèm với robot còn là một phần mềm mô phỏng để giám sát việc nhận diện quân cờ và đánh cờ.

- **Ưu điểm:**
 - Kích thước nhỏ gọn, thời gian đáp ứng ngắn.
 - Phần mềm chơi cờ hỗ trợ nhiều chế độ và mức độ khó khác nhau.
- **Nhược điểm:**
 - Đầu công tác là nam châm nên không dùng được cho tất cả loại quân cờ.
 - Hành trình đầu công tác chưa tối ưu.

c. Robot chơi cờ tướng SenseRobot (2022)



Hình 1.7: SenseRobot của công ty SenseTime.

SenseRobot [4] trong hình 1.7 là một sản phẩm thường mại ra mắt vào ngày 9 tháng 8 năm 2022 của tập đoàn SenseTime. Robot này ứng dụng Deep Learning, có thể tương tác với người dùng thông qua gương mặt và giọng nói, sử dụng cánh tay robot 3 bậc tự do với đầu công tác nam châm để di chuyển quân cờ. Robot có thể cài đặt và kết nối thông qua ứng dụng di động. Nó sở hữu giao diện thân thiện và hoạt động an toàn với người sử dụng.

- **Ưu điểm:**

- Kích thước nhỏ gọn.
- Thời gian đáp ứng ngắn.
- Giao diện người dùng thân thiện và dễ sử dụng
- An toàn cho người dùng.
- Hành trình đầu công tác tối ưu

- **Nhược điểm:**

- Đầu công tác là nam châm nên không dùng được cho tất cả loại quân cờ.
- Giá thành cao, robot trong giai đoạn phát triển, chưa được phổ biến rộng rãi.

1.2.2. Tình hình nghiên cứu trong nước

a. *Robot chơi cờ tướng từ trường Đại học Bách Khoa Thành phố Hồ Chí Minh (2013)*
 Ở Việt Nam, robot chơi cờ chưa thực sự phổ biến. Tại trường Đại học Bách Khoa thành phố Hồ Chí Minh, robot chơi cờ tướng được lên ý tưởng và thực hiện lần đầu tiên bởi hai sinh viên Vũ Văn Khắc và Lê Bạch Quý Cường [7] vào năm 2013. (hình 1.8)



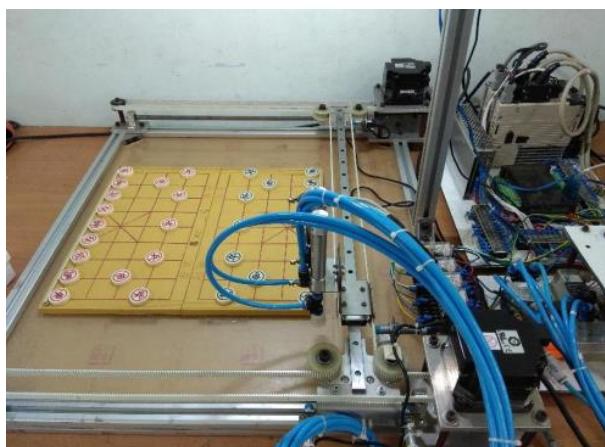
Hình 1.8: Robot của nhóm tác giả Vũ Văn Khắc và Lê Bạch Quý Cường (Trường Đại học Bách Khoa thành phố Hồ Chí Minh).

- Ưu điểm:
 - Thời gian đáp ứng khá tốt.
- Nhược điểm:
 - Hành trình đầu công tác chưa tối ưu và tốc độ còn chậm.
 - Chưa nhận diện được quân cờ.
 - Độ cứng, vững chưa cao.

b. Robot chơi cờ tướng từ trường Đại học Bách Khoa Thành phố Hồ Chí Minh (2017)

Vào năm 2017, hai sinh viên Lê Phương Trà và Nguyễn Thành Việt [8] đã hoàn thành đề tài về robot chơi cờ tướng với kết cấu di chuyển quân cờ như hình 1.9, sử dụng bàn máy CNC 2D được thiết kế theo mô hình H-Robot.

- Ưu điểm:
 - Độ cứng, vững tốt.
 - Hành trình đi của đầu công tác được tối ưu.
- Nhược điểm:
 - Nhận diện quân cờ vẫn chưa tốt.
 - Chưa có khả năng nhận diện nước đi sai.



Hình 1.9: Robot của tác giả Lê Phương Trà và Nguyễn Thành Việt (Trường Đại học Bách Khoa thành phố Hồ Chí Minh).

1.3. Mục tiêu – Nhiệm vụ - Phạm vi đề tài

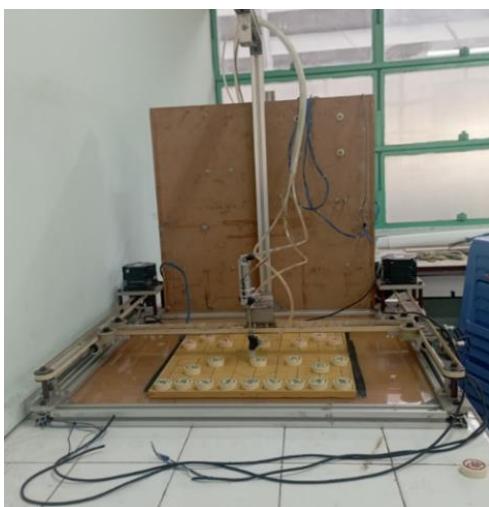
1.3.1. Mục tiêu

Hiện nay, robot chơi cờ tướng đã có sẵn tại bộ môn Cơ điện tử, khoa Cơ khí, trường Đại học Bách khoa – ĐHQG TPHCM. Theo báo cáo luận văn gần nhất năm 2018 của nhóm

tác giả Nguyễn Văn An – Hoàng Minh Cường [9] cùng các video hoạt động thực tế thì robot chơi cờ tướng đã hoạt động được, có khả năng nhận diện và gấp thả quân cờ đúng theo tính toán từ máy tính, đồng thời đưa ra các nước cờ đối phó với người chơi.

Về hạn chế, việc nhận diện quân cờ chưa ổn định dẫn đến việc dừng chơi cờ giữa chừng. Thêm nữa, phần mềm chơi cờ chưa thể tính toán những nước đi có độ sâu lớn do hạn chế về phần cứng.

Vào năm 2019, trong quá trình thực hiện đề tài luận văn tốt nghiệp, một sinh viên đã tháo dỡ một phần robot, nhưng vì tình hình dịch bệnh Covid 19 nên không thể tiếp tục. Từ đó robot chưa được phục hồi đúng hiện trạng ban đầu như hình 1.10, một số linh kiện cũng đã hư hỏng hoặc bị thiêu.



a: Mặt trước robot.



b: Mặt sau robot.

Hình 1.10: Hiện trạng robot chơi cờ tướng tại bộ môn Cơ điện tử, khoa Cơ khí, trường Đại học Bách khoa thành phố Hồ Chí Minh.

Do đó, đề tài này có hướng tới ba mục tiêu lớn để cải thiện và giải quyết:

- *Thứ nhất* là khôi phục hoạt động của robot, đầu công tác có thể di chuyển đến đúng vị trí mục tiêu và gấp – thả quân cờ được.
- *Thứ hai* là tăng tốc độ và độ chính xác nhận diện quân cờ trên bàn cờ, đảm bảo luôn nhận diện đúng trong suốt quá trình chơi.
- *Thứ ba* là xây dựng giải thuật chơi cờ, đưa ra những nước cờ nhanh và “mạnh”. Phát triển nhiều chế độ chơi với những cấp độ chơi từ dễ đến khó để phù hợp với những đối tượng chơi khác nhau.

1.3.2. Nhiệm vụ

Từ ba mục tiêu lớn đã đề cập, mục này sẽ trình bày những nhiệm vụ cụ thể như sau:

- Phân tích kết cấu cơ khí, hệ thống điện của robot chơi cờ.
 - Tìm hiểu sơ đồ nguyên lý bàn máy CNC 2D dẫn động bằng đai thiết kế theo mô hình H-bot (gồm cấu tạo và phân tích động học).
 - Tìm hiểu các cụm chức năng: cụm khung, cụm pulley dẫn hướng, cụm pulley bị động, cụm động cơ, cụm cơ cấu chấp hành.
 - Tìm hiểu sơ đồ đấu dây của hệ thống.
 - Tìm hiểu các bộ phận cấu thành hệ thống điện, gồm: driver, động cơ servo, cảm biến, bộ điều khiển.
- Xây dựng giải thuật điều khiển đầu công tác.
 - Tìm hiểu và thiết lập các thông số điều khiển động cơ cho driver.
 - Tính toán tần số và số xung cần thiết cho việc điều khiển động cơ.
 - Viết chương trình giao tiếp truyền nhận giữa bộ điều khiển và máy tính.
 - Viết chương trình điều khiển đầu công tác tới đúng vị trí của quân cờ.
- Xác định vị trí thực và tên của các quân cờ
 - Tìm hiểu các khái niệm cơ bản về xử lý ảnh: các loại ảnh thông dụng và một số thuật toán xử lý ảnh.
 - Xác định tọa độ các quân cờ trên ảnh.
 - Xây dựng cấu trúc mạng neuron tích chập, thu thập bộ dữ liệu ảnh và huấn luyện mạng để nhận diện tên các quân cờ đã tìm được.
 - Hiệu chỉnh camera để xác định các thông số nội, ngoại của camera; từ đó biến đổi tọa độ ảnh thành tọa độ thực trên bàn cờ.
- Phát triển giải thuật chơi cờ
 - Viết chương trình điều khiển quá trình sắp xếp bàn cờ.
 - Tìm hiểu các giải thuật chơi cờ cơ bản: giải thuật Minimax, cắt tỉa Alpha – Beta, các kĩ thuật khai cuộc và tàn cuộc.
 - Viết chương trình chơi cờ, cho phép lựa chọn 3 chế độ: dễ, trung bình và khó, cho phép chơi từ thế cờ có sẵn.
 - Phát triển khả năng nhận diện nước đi sai của người chơi.

1.3.3. Phạm vi đề tài

a. Đối với robot

Đề tài sử dụng webcam Rapoo C200 độ phân giải 720p, góc ống kính 89° (hình 1.11).



Hình 1.11: Webcam Rapoo C200.

Môi trường thực hiện đề tài: môi trường trong nhà với ánh sáng thích hợp và cường độ sáng thay đổi không đáng kể.

Các thông số bộ cờ (hình 1.12):

- Kích thước bàn cờ: 410×365 (mm \times mm).
- Kích thước mỗi ô cờ: 41×41 (mm \times mm).



Hình 1.12: Bàn cờ tướng.

- Kích thước quân cờ (hình 1.13): bán kính 16 mm, chiều cao 12 mm.



Hình 1.13: Quân cờ tướng.

b. Đối với phần mềm

- Nhận diện quân cờ: Trong giới hạn đề tài, nhóm tác giả chỉ khảo sát 3 môi trường với ánh sáng tốt và ổn định là môi trường trong Phòng Thí nghiệm Điều khiển và tự động hóa P202C1, khu vực trước xưởng C1 và khu vực ghé đá trước tòa B1 trong những ngày trời có nắng.

- Xếp cờ: Các quân cờ được lật ngửa mặt khắc biểu tượng quân, đặt đầy đủ trên bàn cờ, và trong khu vực xác định như hình 1.14.



Hình 1.14: Giới hạn không gian đặt quân cờ trước khi xếp.

- Nhận diện nước đi của người chơi: Các lỗi được kiểm tra khi nhận diện nước đi là các lỗi vi phạm theo luật cờ tướng.

1.4. Tổ chức luận văn

Từ các mục tiêu và nhiệm vụ đã trình bày, đề tài được chia thành 5 chương. Các chương tiếp theo này sẽ được triển khai để thực hiện các nhiệm vụ đã đề ra:

- Chương 2 tìm hiểu về robot, bao gồm các nội dung: kết cấu cơ khí, hệ thống điện, khí nén.
- Chương 3 phân tích và xây dựng giải thuật nhận diện vị trí và tên quân cờ.
- Chương 4 phân tích và xây dựng giải thuật điều khiển đầu công tác di chuyển và gấp thả quân cờ, xây dựng giao thức truyền thông giữa bộ điều khiển và máy tính.
- Chương 5 phân tích và xây dựng giải thuật chơi cờ, cùng chương trình mô phỏng để kiểm nghiệm trên máy tính.
- Chương 6 tổng kết đề tài, phân tích những điểm đã làm được và cần cải thiện, từ đó đưa ra hướng phát triển tiếp theo.

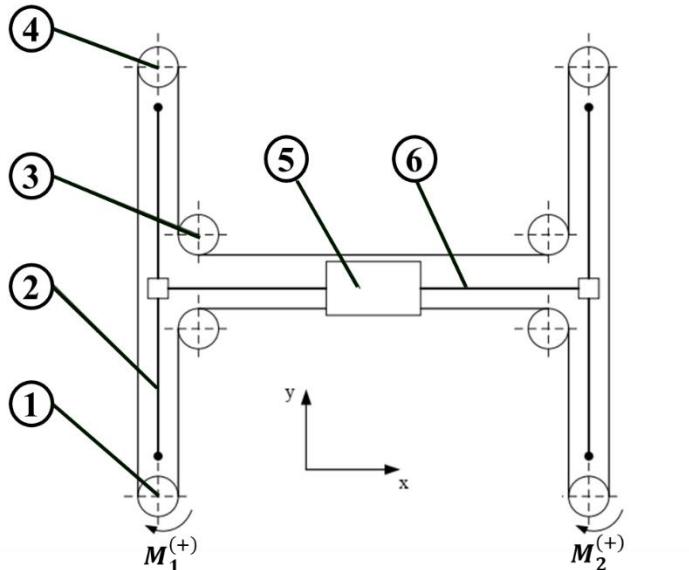
CHƯƠNG 2. TÌM HIỂU VÀ KHÔI PHỤC HỆ THỐNG ROBOT CHƠI CỜ TƯỚNG

Chương này tập trung vào quá trình tìm hiểu và khôi phục hoạt động của các kết cấu cơ khí và hệ thống điện của robot chơi cờ tướng. Về tình trạng phần cơ khí, các cụm cơ khí vẫn hoạt động bình thường; dây đai có dấu hiệu nứt, mát độ đàn hồi và không thể sử dụng tiếp; một số mối lắp bị lỏng. Về tình trạng phần điện, các thành phần của hệ thống điện như động cơ, relay, bộ nguồn,... vẫn hoạt động tốt nhưng các thành phần này chưa được đấu nối với nhau hoàn chỉnh. Mục tiêu cần đạt được của chương này là chuẩn bị về mặt lý thuyết và khôi phục lại hoạt động của robot để tạo cơ sở cho việc điều khiển robot ở chương 4.

2.1. Tìm hiểu về phần cơ khí

2.1.1. Sơ đồ nguyên lý của robot

Robot chơi cờ hiện có hoạt động theo nguyên lý của mô hình H-bot dẫn động bằng một đai như hình 2.1.



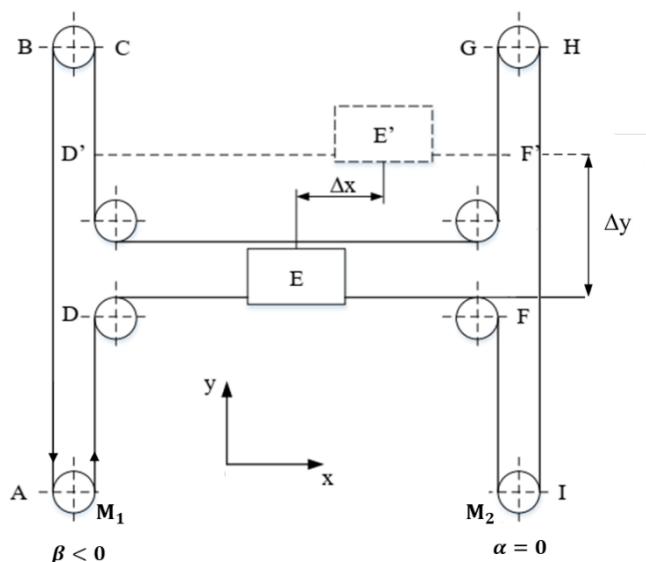
Hình 2.1: Sơ đồ nguyên lý robot chơi cờ tướng.

Hai đầu dây đai được nối với nhau thông qua bộ phận gá đai (5) tạo thành một vòng đai khép kín, vòng đai này sẽ cuốn qua các pulley chủ động (1), pulley bị động (4) và pulley tròn dẫn hướng (3). Vòng đai này kết hợp với 2 trục dẫn hướng (2) và (6) theo phương x và y tạo nên hình dạng chữ H như trong tên của mô hình. Khi hai động cơ gắn với 2

pulley chủ động quay thì đầu công tác gắn với bộ phận gá đai sẽ di chuyển theo quỹ đạo thẳng trong mặt phẳng XY. Để xác định mối quan hệ giữa góc quay và chiều quay của hai động cơ với quỹ đạo di chuyển của đầu công tác trong từng trường hợp cụ thể thì cần phân tích động học của robot.

2.1.2. Phân tích động học robot

Chọn hệ tọa độ như hình 2.2. Trong đó, các giá trị Δx , Δy lần lượt là chuyển vị theo phương x và phương y. Chiều quay dương quy ước là thuận chiều kim đồng hồ, với động cơ M_1 nằm bên phải có góc quay α và động cơ M_2 nằm bên trái có góc quay β . Xét trường hợp động cơ M_1 đứng yên, động cơ M_2 quay một góc $\beta = -\theta_0$ ($\theta_0 > 0$):



Hình 2.2: Chuyển vị của đầu công tác khi M_1 đứng yên, M_2 quay theo chiều âm.

Trong trường hợp trên, chiều dài CD và GF sẽ thay đổi một lượng Δy thành CD' và GF' do động cơ M_2 quay ngược chiều kim đồng hồ và AB, DF, HI có chiều dài không thay đổi. Do đó, chiều dài của đoạn CDFG giảm một lượng $\theta_0 R$ thành CD'F'G. (R là bán kính pulley chủ động)

$$(CD - CD') + (GF - GF') = \theta_0 R \quad (2.1)$$

Biết rằng $CD - CD' = GF - GF' = \Delta y$, từ đó tính được:

$$\Delta y = CD - CD' = 0,5\theta_0 R \quad (2.2)$$

Đồng thời, đoạn ADEFI sẽ dài thêm một đoạn $\theta_0 R$ trong khi chiều dài EFI không đổi (do M_1 đứng yên) nên E sẽ dịch chuyển sang phải một đoạn Δx . Biết rằng AD dịch chuyển một đoạn bằng Δy , khi đó:

$$\Delta x = \theta_0 R - \Delta y = 0,5\theta_0 R \quad (2.3)$$

Các trường hợp với góc quay và chiều quay động cơ khác được xét tương tự như cách xét trường hợp trên và cho ra kết quả như bảng 2.1.

Bảng 2.1: Các trường hợp di chuyển của đầu công tác.

| α | β | Δ_x | Δ_y |
|-------------|-------------|------------------|------------------|
| a | 0 | $-0,5\theta_0 R$ | $0,5\theta_0 R$ |
| $-\theta_0$ | 0 | $0,5\theta_0 R$ | $-0,5\theta_0 R$ |
| 0 | θ_0 | $-0,5\theta_0 R$ | $-0,5\theta_0 R$ |
| 0 | $-\theta_0$ | $0,5\theta_0 R$ | $0,5\theta_0 R$ |
| θ_0 | θ_0 | $-\theta_0 R$ | 0 |
| $-\theta_0$ | $-\theta_0$ | $\theta_0 R$ | 0 |
| θ_0 | $-\theta_0$ | 0 | $\theta_0 R$ |
| $-\theta_0$ | θ_0 | 0 | $-\theta_0 R$ |

Dựa vào bảng 2.1, hệ phương trình động học thuận của robot được rút ra như sau:

$$\begin{cases} \Delta_x = -0,5R(\alpha + \beta) \\ \Delta_y = 0,5R(\alpha - \beta) \end{cases} \quad (2.4)$$

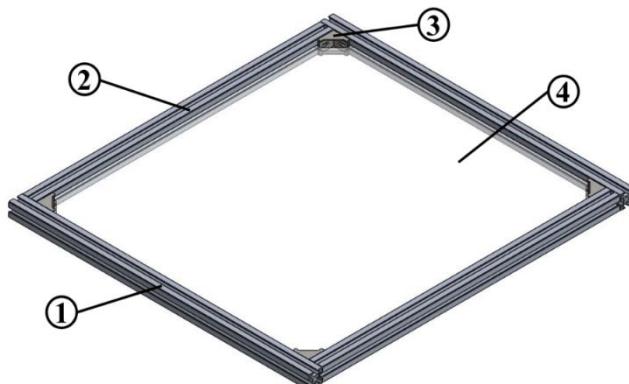
Hệ phương trình động học nghịch được suy ra từ công thức 2.4 như sau:

$$\begin{cases} \alpha = \frac{-\Delta_x + \Delta_y}{R} \\ \beta = \frac{-\Delta_x - \Delta_y}{R} \end{cases} \quad (2.5)$$

2.1.3. Các cụm cơ khí của robot

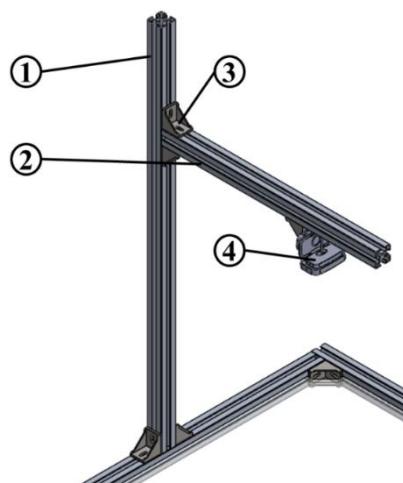
a. Cụm khung đế

Cụm khung đế được hình thành từ hai thanh nhôm định có kích thước là $30 \times 30 \times 650$ (mm) (1) và hai thanh nhôm định hình có kích thước là $30 \times 30 \times 644$ (mm) (2) ghép nối với nhau bởi các miếng ke góc (3) như hình 2.3. Chính giữa là tấm đế mica (4).



Hình 2.3: Cụm khung đế.

b. Cụm gá đặt camera

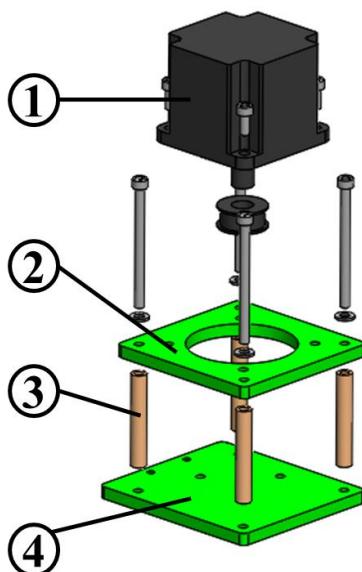


Hình 2.4: Cụm gá đặt camera.

Cụm gá camera được hình thành từ một thanh nhôm định hình có kích thước $30 \times 30 \times 760$ (mm) (1), một thanh nhôm định hình có kích thước là $30 \times 30 \times 440$ (mm) (2) ghép nối với nhau bởi các miếng ke góc (3) và dùng để gá camera (4) như hình 2.4.

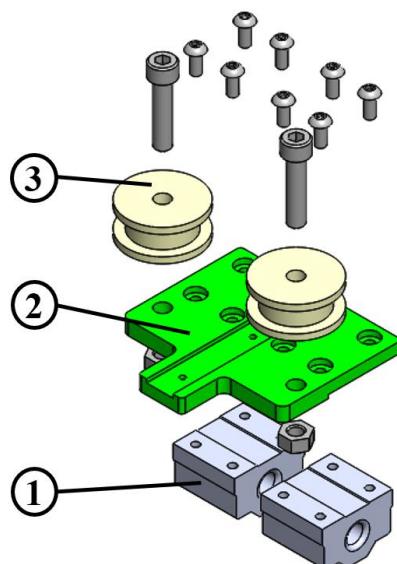
c. Cụm pulley chủ động

Kết cấu cụm pulley chủ động như thể hiện trong hình 2.5 bao gồm các thành phần như sau: động cơ (1) với trục động cơ hướng xuống có gắn pulley chủ động (5); miếng gá mặt bích động cơ (2); miếng đế nhôm (4) để gá cả cụm chi tiết với cụm khung đế; bốn trụ đồng (3) ghép nối miếng đế nhôm với miếng gá mặt bích động cơ. Robot sẽ có hai cụm pulley chủ động giống nhau đặt ở hai góc của cụm khung đế tương ứng với hai động cơ M_1 và M_2 .



Hình 2.5: Cụm pulley chủ động.

d. Cụm pulley dẫn hướng

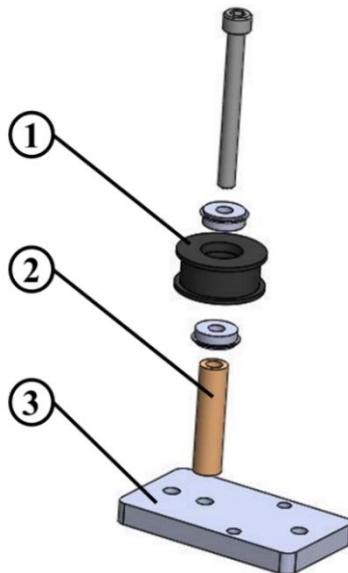


Hình 2.6: Cụm pulley dẫn hướng.

Cụm pulley dẫn hướng có kết cấu như hình 2.6, bao gồm: hai con trượt (1) chuyển động dọc theo phương y; tấm nhôm chữ T (2) vừa có vai trò gá hai con trượt vừa có vai trò gắn hai pulley trượt dẫn hướng (3). Ngoài ra, mặt trên của tấm chữ T còn được phay bậc để gá đặt thanh trượt theo phương x.

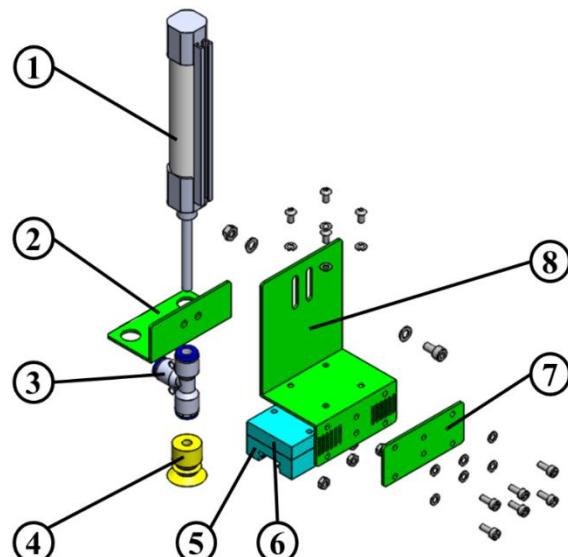
e. Cụm pulley bị động

Cụm pulley bị động được hình thành từ pulley (1) gắn đồng trực với ống trụ đồng (2) và gắn trên miếng đế nhôm (3) như hình 2.7.



Hình 2.7: Cụm pulley bị động.

f. Cụm cơ cầu cháp hành

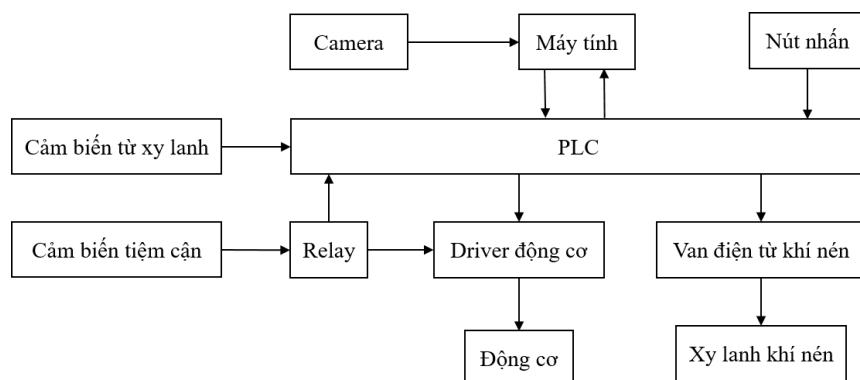


Hình 2.8: Cụm cơ cầu cháp hành.

Hình 2.8 thể hiện kết cấu của cụm cơ cầu cháp hành, bao gồm xylanh (1) được gá lên miếng nhôm (2), giác hút (4) được gắn vào đầu xy lanh thông qua van chữ T (3); miếng gá đỡ (8) là chi tiết trung gian kết nối miếng nhôm gá xy lanh ở trên với con trượt (5) và tám kẹp đai (7). Giữa miếng gá đỡ này và con trượt còn có một miếng mica (6). Cả cụm cơ cầu này sẽ di chuyển dọc theo phương y.

2.2. Tìm hiểu về phần điện

Hệ thống điện của robot bao gồm các thành phần như: PLC, máy tính, nguồn, relay, driver động cơ, van điện từ khí nén, xy lanh khí nén, driver động cơ, động cơ, đèn báo, công tắc, nút nhấn, camera, cảm biến tiệm cận, cảm biến từ xy lanh. Sự tương tác về mặt tín hiệu giữa khối trung tâm gồm PLC và máy tính với các thiết bị khác trong hệ thống điện được khái quát như hình 2.9.



Hình 2.9: Sơ đồ tín hiệu của các thành phần trong hệ thống điện.

2.2.1. Camera



Hình 2.10: Camera Rapoo C200.

Camera trong robot chơi cờ đóng vai trò như thị giác của máy tính, cung cấp cho máy tính hình ảnh theo thời gian thực của bàn cờ và các quân cờ để từ đó máy tính xác định các quân cờ và vị trí của chúng thông qua các thuật toán xử lý ảnh. Camera được sử dụng trong đề tài này là camera Rapoo C200 như hình 2.10 cùng các thông số cụ thể như bảng 2.2.

Bảng 2.2: Thông số camera Rapoo C200.

| | |
|---------------|------------------|
| Độ phân giải | HD 720p |
| Chuẩn kết nối | USB 2.0 |
| Kích thước | 126 × 60 × 118mm |
| Góc nhìn | 78° |

2.2.2. Cảm biến từ xy lanh

Cảm biến xy lanh đóng vai trò kiểm soát hành trình ra vào của xy lanh khí nén. Cảm biến này hoạt động theo nguyên lý của cảm biến từ, khi nam châm gắn bên trong xy lanh đi ngang qua cảm biến thì công tắc từ sẽ đóng lại. Thông thường, mỗi xy lanh sẽ sử dụng 2 cảm biến này cho hành trình ra và vào. Trong đề tài này, chỉ có một cảm biến được sử dụng để kiểm soát hành trình ra của xy lanh khi đầu công tác gấp thả quân cờ. Cảm biến được sử dụng là SMC D-A93 như hình 2.11 với các thông số như bảng 2.3.



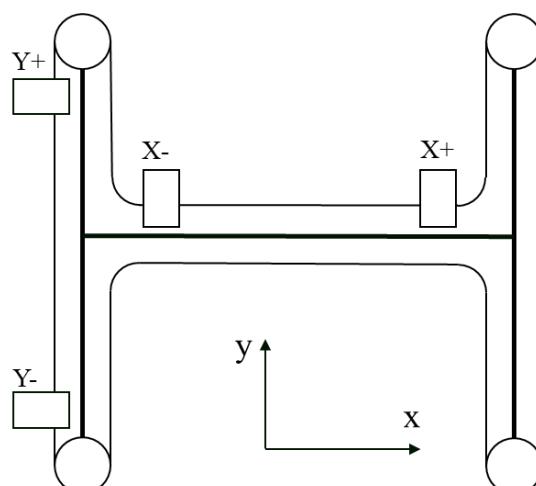
Hình 2.11: Cảm biến từ SMC D-A93.

Bảng 2.3: Thông số cảm biến từ SMC D-A93.

| | |
|---------------|--------------------------|
| Nguồn vào | 24V DC |
| Dòng tiêu thụ | 5-40mA |
| Tín hiệu ra | NO |
| Sơ đồ dây | Nâu: Tín hiệu; Xanh: GND |

2.2.3. Cảm biến tiệm cận

Cảm biến tiệm cận trong robot này làm nhiệm vụ giới hạn hành trình bàn máy, đảm bảo đầu công tác chỉ hoạt động trong vùng không gian cho phép, tránh va chạm và làm hư hỏng bàn máy. Có tất cả 4 cảm biến tiệm cận để giới hạn hành trình theo chiều dương và âm của hai phương x và y như hình 2.12.



Hình 2.12: Vị trí đặt các cảm biến tiệm cận trên bàn máy.

Loại cảm biến tiệm cận sử dụng trong đề tài này là LJ12A3-4-Z/BX NPN như hình 2.13. Đây là loại cảm biến được sử dụng để phát hiện vật kim loại trong khoảng cách từ 0 đến 4mm. Ở trạng thái bình thường dây tín hiệu của cảm biến sẽ ở mức 24V, khi phát hiện vật thể thì chuyển sao mức 0V. Các thông số của cảm biến được thể hiện trong bảng 2.4.



Hình 2.13: Cảm biến tiệm cận LJ12A3-4-Z/BX NPN.

Bảng 2.4: Thông số cảm biến tiệm cận LJ12A3-4-Z/BX NPN.

| | |
|-----------------------|--|
| Nguồn vào | 6-36V DC |
| Dòng tiêu thụ | 300mA |
| Khoảng cách phát hiện | 4mm |
| Vật liệu phát hiện | Kim loại |
| Tín hiệu ra | NPN, NO |
| Sơ đồ dây | Nâu: 6 – 36V DC; Xanh dương: GND; Đen: dây tín hiệu ra |

2.2.4. Van điện từ khí nén



(a) Van solenoid STNC TG2521-08. (b) Van solenoid STNC TG2321-08.

Hình 2.14: Các van solenoid được dùng trong hệ thống.

Van điện từ khí nén (solenoid valve) được sử dụng để điều khiển việc nạp xả khí nén thông qua tín hiệu điện. Trong đề tài này, loại van được sử dụng là van STNC TG2521-08 (5 cửa, 2 vị trí) dành cho xy lanh khí nén và van STNC TG2321-08 (3 cửa, 2 vị trí)

dành cho giắc hút. Hình 2.14 là hình ảnh thực tế của 2 loại van này, thông số chi tiết của 2 loại van lần lượt thể hiện trong bảng 2.5.

Bảng 2.5: Thông số của van solenoid STNC TG2521-08 và STNC TG2321-08.

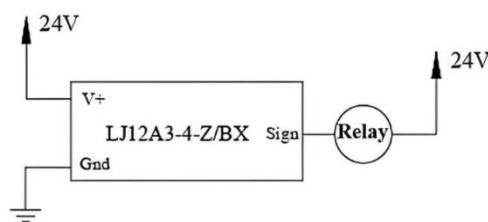
| | STNC TG2521-08 | STNC TG2321-08 |
|--------------------|----------------------------------|----------------------------------|
| Loại van | 5 cửa 2 vị trí (1 đầu coil điện) | 3 cửa 2 vị trí (1 đầu coil điện) |
| Kích thước công | 1/4" (ren 13mm) | 1/4" (ren 13mm) |
| Áp suất hoạt động | 0.2 – 0.8 MPa | 0.15 – 0.8 MPa |
| Nhiệt độ hoạt động | 5 – 60°C | 5 – 60°C |

2.2.5. Relay



Hình 2.15: Relay kính OMRON MY4N.

Các tín hiệu từ cảm biến tiệm cận được sử dụng để đóng ngắt các chân N-OT và P-OT của driver, từ đó khóa chiều quay thuận hay nghịch của hai động cơ khi đầu công tác đi đến giới hạn hành trình. Ngoài ra, các tín hiệu này còn được đưa về ngõ vào của PLC để phục vụ cho việc lập trình. Vì vậy, hệ thống cần 4 relay với đầu vào là tín hiệu từ 4 cảm biến tiệm cận, một đầu ra thường đóng để dùng cho driver và một đầu ra thường hở để đưa vào PLC. Relay được sử dụng trong đề tài này là relay kính OMRON MY4N như hình 2.15.



Hình 2.16: Sơ đồ đấu dây cảm biến tiệm cận với relay kính.

Sơ đồ đấu dây của relay và cảm biến tiệm cận được thể hiện trong hình 2.16. Relay kính được đấu với cảm biến tiệm cận theo phương x có ký hiệu là X- và X+, các relay đấu với cảm biến tiệm cận theo phương y có ký hiệu là Y- và Y+.

2.2.6. Driver động cơ

Driver động cơ được sử dụng là AC Servo SGDV 1R6A01 như hình 2.17. Driver nhận xung từ ngõ ra PLC để điều khiển tốc độ và chiều động cơ AC Servo. Các thông số cơ bản của driver thể hiện trong bảng 2.6.



Hình 2.17: Driver SGDV 1R6A01.

Bảng 2.6: Thông số driver SGDV 1R6A01.

| | |
|-----------------------|--|
| Nguồn vào | 200-230V AC |
| Công suất | 200W |
| Truyền thông | RS442-A, USB |
| Dải điều khiển tốc độ | 1:5000 |
| Chức năng | bảo vệ quá dòng, quá áp, thấp áp, quá tải, lỗi phát sinh |

Bảng 2.7: Logic đóng ngắt chân P-OT và N-OT của driver.

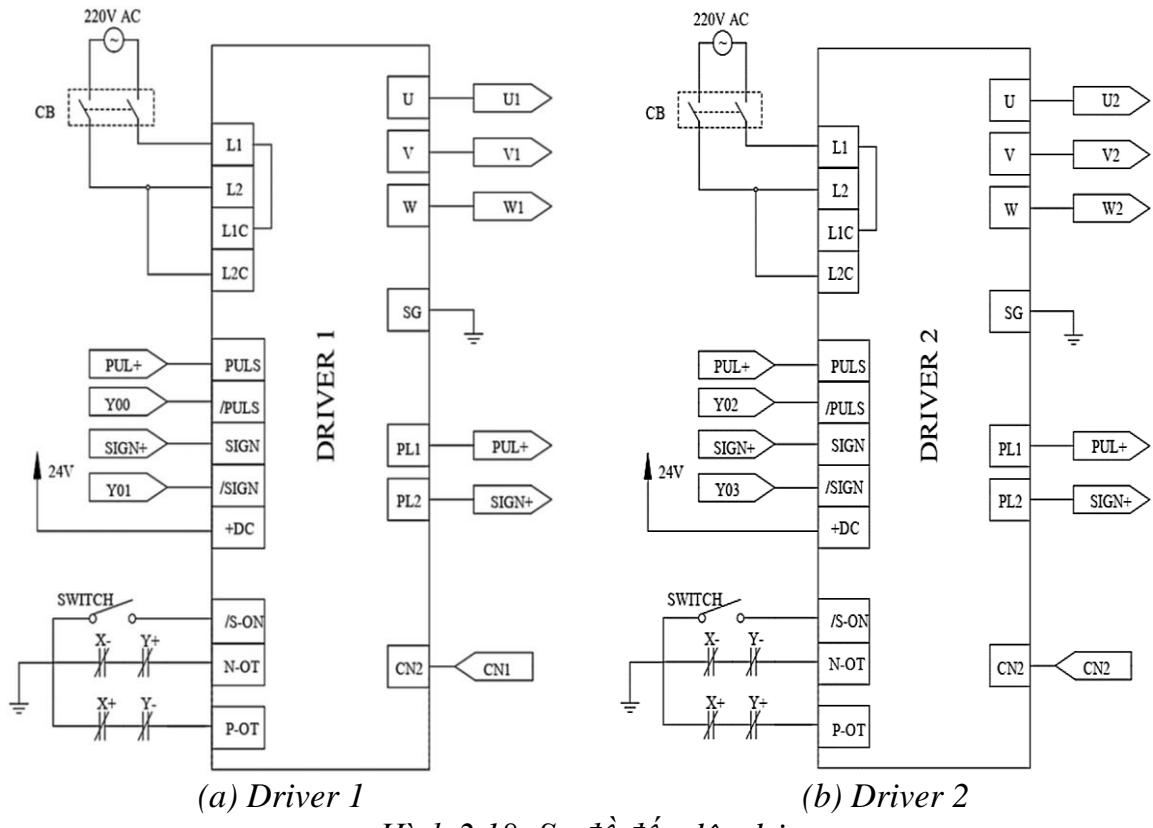
| Cảm biến tiệm cận | | | | Driver 1 | | Driver 2 | |
|-------------------|----|----|----|----------|------|----------|------|
| X+ | Y+ | X- | Y- | N-OT | P-OT | N-OT | P-OT |
| 1 | 0 | 0 | 0 | | x | | x |
| 0 | 1 | 0 | 0 | x | | | x |
| 0 | 0 | 1 | 0 | x | | x | |
| 0 | 0 | 0 | 1 | | x | x | |

Tín hiệu đưa vào driver sẽ là tín hiệu từ các chân phát xung và chiều của PLC. Ngoài ra, chân /S-ON, P-OT và N-OT khi được đưa vào điện áp 0V thì động cơ sẽ hoạt động bình thường. Khi /S-ON bị ngắt, động cơ sẽ ngừng quay, P-OT bị ngắt thì động cơ sẽ bị khóa chiều thuận, N-OT bị ngắt thì động cơ sẽ bị khóa chiều nghịch. Thông qua việc khảo sát các phương trình động học của robot, logic đóng ngắt các chân P-OT và N-OT được rút ra thành bảng 2.7. Những quy ước về ký hiệu và chức năng các chân driver được thể hiện đầy đủ ở bảng 2.8.

Bảng 2.8: Ký hiệu và chức năng các chân driver.

| Chân | Ký hiệu | Chức năng |
|----------|---------|--------------------------------------|
| 1 | SG | Nối đất |
| 7 | PULS | Chân PL1 (3) của driver |
| 8 | /PULS | Ngõ ra của PLC |
| 11 | SIGN | Chân PL2 (13) của driver |
| 12 | /SIGN | Ngõ ra của PLC |
| 40 | S-ON | Công tắc điện (switch) |
| 42 | P-OT | Tiếp điểm thường đóng của relay kính |
| 43 | N-OT | Tiếp điểm thường đóng của relay kính |
| 47 | Vin | Chân nguồn 24VDC |
| L1, L2 | | Nguồn 1 pha 220V AC |
| L1C, L2C | | Nguồn 1 pha 220V AC hoặc 24V DC |
| U,V,W | | Đầu với 3 dây động cơ |
| CN2 | | Chân đọc encoder của động cơ |

Từ những thông tin về các chân và chức năng của từng chân, kết hợp với logic đóng ngắt P-OT và N-OT, sơ đồ đấu dây của driver 1 (driver điều khiển động cơ M_1) với các thiết bị như PLC, relay và động cơ M_1 sẽ được thể hiện ở hình 2.18a. Tương tự, hình 2.18b thể hiện cách đấu dây của driver 2 (driver điều khiển động cơ M_2) với PLC, relay và động cơ M_2 .



Hình 2.18: Sơ đồ đầu dây driver.

2.2.7. Động cơ



Hình 2.19: Động cơ AC Servo SGMPS-02ACA21.

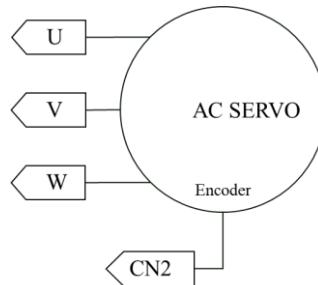
Robot sử dụng 2 động cơ AC Servo SGMPS-02ACA21 như hình 2.19 để truyền động cho đai và di chuyển đầu công tác trong mặt phẳng XY. Các thông số quan trọng của động cơ được tóm tắt trong bảng 2.9.

Bảng 2.9: Thông số động cơ AC Servo SG MPS-02ACA21.

| | |
|------------------------|-------|
| Công suất (W) | 200 |
| Momen xoắn: [N.m] | 0.637 |
| Điện áp cung cấp (VAC) | 200 |
| Dòng điện [A] | 2 |

| | |
|----------------------------|--------|
| Encoder | 17 bit |
| Vận tốc tối đa (vòng/phút) | 3000 |
| Cân nặng (Kg) | 1 |

Các dây nguồn của động cơ được nối với 3 chân U, V, W của driver như hình 2.20. Dây encoder được nối vào kênh CN2 của driver.



Hình 2.20: Sơ đồ đấu dây động cơ.

2.2.8. Xy lanh khí nén

Xy lanh khí nén đóng vai trò là cơ cấu tịnh tiến lên xuống để gấp và thả quan cờ với trực xy lanh được gắn một giác hút. Trong robot đánh cờ hiện tại, xy lanh được sử dụng là MCE987845 như hình 2.21. Các thông số xy lanh này được thể hiện trong bảng 2.10.

Bảng 2.10: Thông số xy lanh MCE987845.

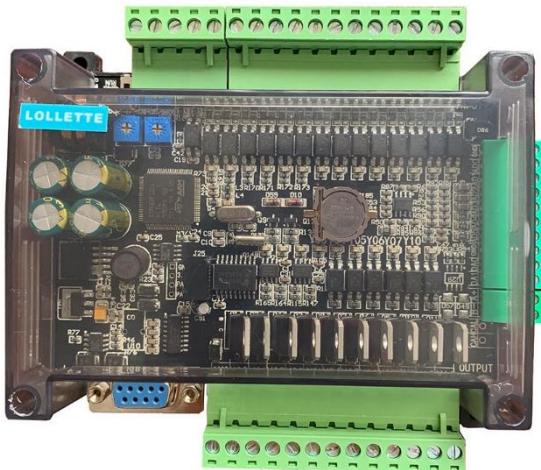
| | |
|--------------------------------|------|
| Đường kính nòng (mm) | 16 |
| Đường kính trục (mm) | 5 |
| Hành trình (mm) | 45 |
| Áp suất hoạt động tối đa (Mpa) | 0,07 |



Hình 2.21: Xy lanh khí nén MCE987845.

2.2.9. PLC

PLC FX3U LE3U 24MT như hình 2.22 với những thông số như bảng 2.11 được sử dụng trong đề tài này với vai trò là bộ điều khiển nhận các tín hiệu từ các cảm biến và nút bấm, giao tiếp với máy tính và điều khiển các cơ cấu chấp hành.



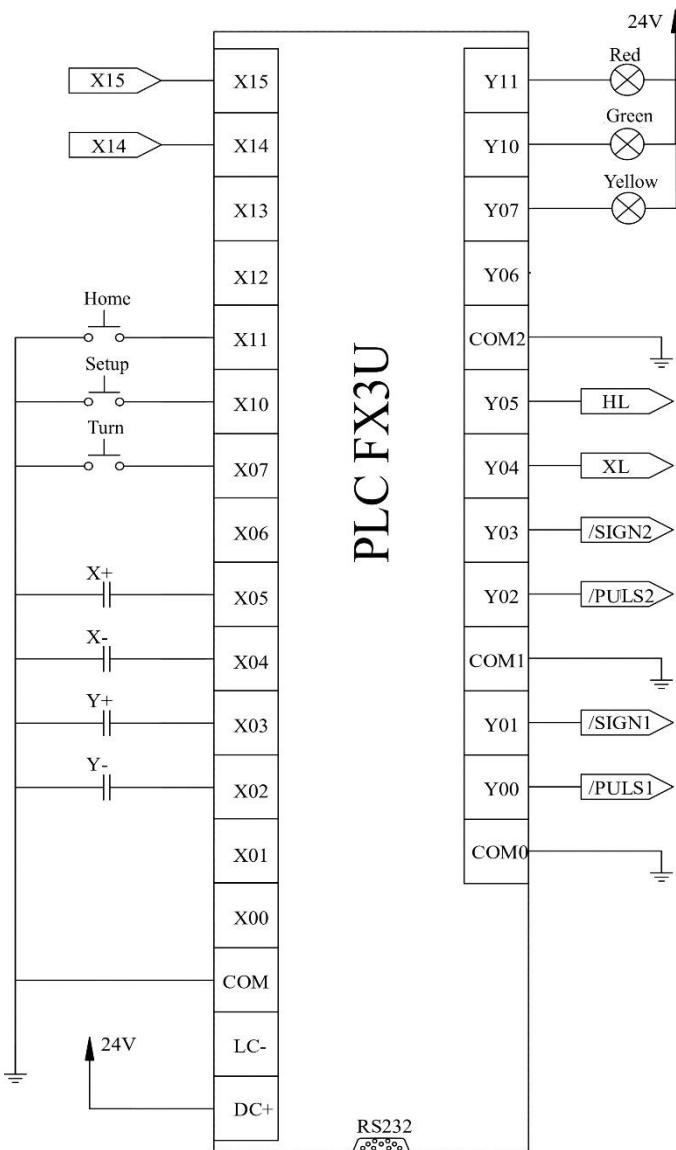
Hình 2.22: Board PLC FX3U LE3U 24MT.

Bảng 2.11: Thông số PLC FX3U LE3U 24MT.

| | |
|---------------------|--|
| Điện áp nguồn | 24V DC |
| Số ngõ vào | 14 ngõ vào, 6 ngõ vào tốc độ cao X0-X5 |
| Số ngõ ra | 10 ngõ ra, 5 ngõ bắn xung tốc độ cao Y00-Y04 |
| Giao tiếp | RS232 |
| Bộ nhớ chương trình | 8000 steps |
| Tốc độ truyền | 38400bps |

Trong đề tài này, PLC sẽ kết nối với máy tính thông qua giao thức RS232. Các ngõ ra xung tốc độ cao Y0, Y2 lần lượt được dùng để phát xung cho động cơ M_1, M_2 quay. Các chân Y1, Y3 được sử dụng để thay đổi chiều quay động cơ. Các cảm biến tiệm cận được đấu nối vào các ngõ vào từ X2 đến X5. Các chân Y4 và Y5 lần lượt được sử dụng để kích xung cho xy lanh đi ra và kích xung để giác hút bắt đầu hút. Cảm biến từ để theo dõi hành trình xy lanh thì được nối với chân X14. Ngoài ra, trong robot chơi cờ có các nút nhấn để bắt đầu chơi cờ, về gốc bàn máy và đổi lượt chơi cờ. Các nút nhấn này được nối lần lượt vào các chân X10, X11 và X7.

Bảng 2.12 tóm hợp lại các chân PLC được sử dụng với ký hiệu và chức năng của các chân đó. Sơ đồ đấu dây chi tiết các chân của PLC này với các thiết bị khác thì được thể hiện ở hình 2.23.



Hình 2.23: Sơ đồ đấu dây PLC.

Bảng 2.12: Ký hiệu và chức năng các chân PLC.

| Thiết bị | Ký hiệu | Chân PLC | Chức năng |
|-------------------------------------|---------|----------|-----------------------------------|
| Cảm biến tiệm cận LJ12A3 – 4 – Z/BX | X+ | X2 | Giới hạn hành trình chiều dương x |
| | X- | X3 | Giới hạn hành trình chiều âm x |
| | Y+ | X4 | Giới hạn hành trình chiều dương y |

| | | | |
|--------------------------------|--------|-----|--------------------------------|
| | Y- | X5 | Giới hạn hành trình chiều âm y |
| Cảm biến SMC D-A93 | X14 | X14 | Giới hạn hành trình xy lanh |
| Nút nhấn | Setup | X10 | Xếp cờ |
| | Home | X11 | Đưa đầu công tác về Home |
| | Turn | X7 | Đổi lượt chơi |
| Driver AC servo SGDV 1R6A01 | /PULS1 | Y0 | Cấp xung cho động cơ 1 |
| | /SIGN1 | Y1 | Thiết lập chiều động cơ 1 |
| | /PULS2 | Y2 | Cấp xung cho động cơ 2 |
| | /SIGN2 | Y3 | Thiết lập chiều động cơ 2 |
| Van điện từ khí nén | XL | Y4 | Điều khiển hành trình xy lanh |
| | HL | Y5 | Điều khiển hút/thả quân cờ |
| Đèn báo | Green | Y10 | Đèn báo đầu công tác về Home |
| | Red | Y11 | Đèn báo nước đi sai |
| | Yellow | Y7 | Đèn báo đến lượt người chơi |

2.2.10. Máy tính

Máy tính được sử dụng là Gigabyte U4 (hình 2.24) với cấu hình: chip Intel Core i5 - 1155G7, tốc độ CPU 2.5GHz, dung lượng RAM 16GB.



Hình 2.24: Máy tính Gigabyte U4.

2.2.11. Nguồn

Nguồn của hệ thống điện bao gồm 3 loại nguồn: 220V AC, 24V DC, 5V DC.

- Nguồn 220V AC được sử dụng cho động cơ, đèn báo khẩn cấp và đèn báo nguồn.
- Nguồn 24V DC được sử dụng cho PLC, driver, van xy lanh khí nén, các cảm biến tiệm cận, cảm biến hành trình xy lanh, relay và các đèn báo nút bấm.
- Nguồn 5V DC là nguồn cấp cho camera.

Trên thực tế, hệ thống chỉ sử dụng nguồn 220V AC, còn điện áp 24V DC được lấy ra từ bộ nguồn xung 24V DC như hình 2.25 với đầu vào là 220V AC cùng các thông số khác như trong bảng 2.14. Nguồn 5V DC dùng cho camera được lấy trực tiếp từ máy tính khi camera kết nối với máy tính qua cổng USB.



Hình 2.25: Bộ nguồn xung 24V DC.

Bảng 2.13: Thông số nguồn xung 24V DC.

| | |
|-----------------------|---------|
| Điện áp đầu vào | 220V AC |
| Điện áp đầu ra | 24V DC |
| Dòng điện cực đại | 5A |
| Sai số điện áp đầu ra | 1-3% |
| Công suất thực tế | 82% |

2.2.12. Đèn báo

Hệ thống đèn báo trong robot bao gồm 2 loại đèn báo như hình 2.26, trong đó: hai đèn báo 220V được dùng để báo có nguồn và báo tình trạng khẩn cấp, ba đèn báo 24V được dùng để báo về Home (xanh), báo lượt người chơi (vàng) và báo nước đi sai (đỏ).



a. Đèn báo 220V.



b. Đèn báo 24V.

Hình 2.26: Các loại đèn báo dùng trong hệ thống điện.

2.2.13. Công tắc và nút bấm

Nguồn 220V AC được kiểm soát thông qua một cầu dao điện tự động (CB) như hình 2.27a cho phép đóng, ngắt và bảo vệ quá dòng. Công tắc điện như hình 2.27b thì được dùng để đóng ngắt động cơ thông qua bật tắt chân Servo ON của hai driver.



a. Cầu dao điện tự động.



b. Công tắc điện.

Hình 2.27: Cầu dao điện tự động và công tắc điện.



a. Nút nhấn xoay.



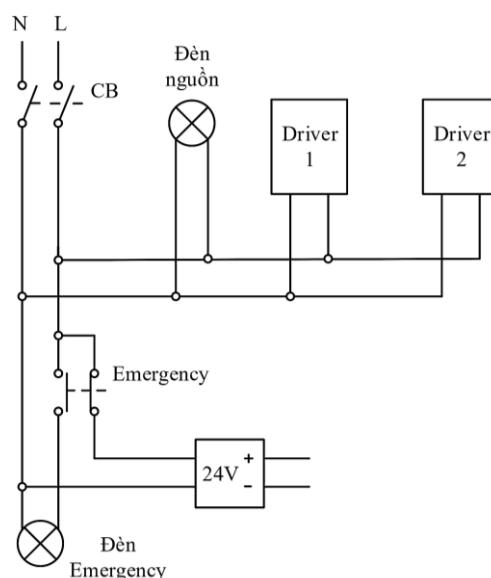
b. Nút nhấn không giữ.

Hình 2.28: Các loại nút nhấn trong hệ thống điện.

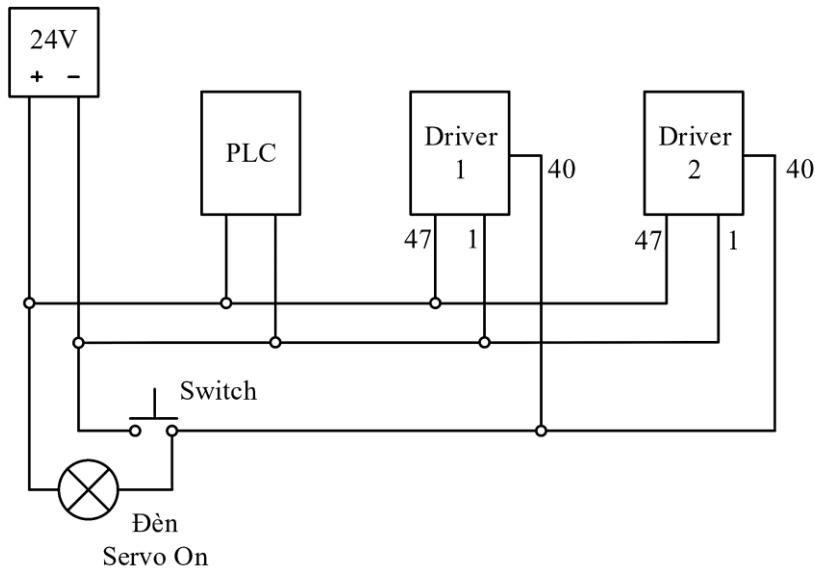
Nút nhấn xoay như hình 2.28a được sử dụng như một nút nhấn khẩn cấp để ngắt nguồn 24V DC.

Các nút nhấn Setup, Home, Turn đều là các nút nhấn không giữ như hình 2.28b nhưng mỗi nút nhấn cho mỗi mục đích sẽ có màu khác nhau.

Các công tắc, cầu dao và nút nhấn cùng các đèn báo được kết nối với nguồn theo các sơ đồ như hình 2.29 và 2.30.



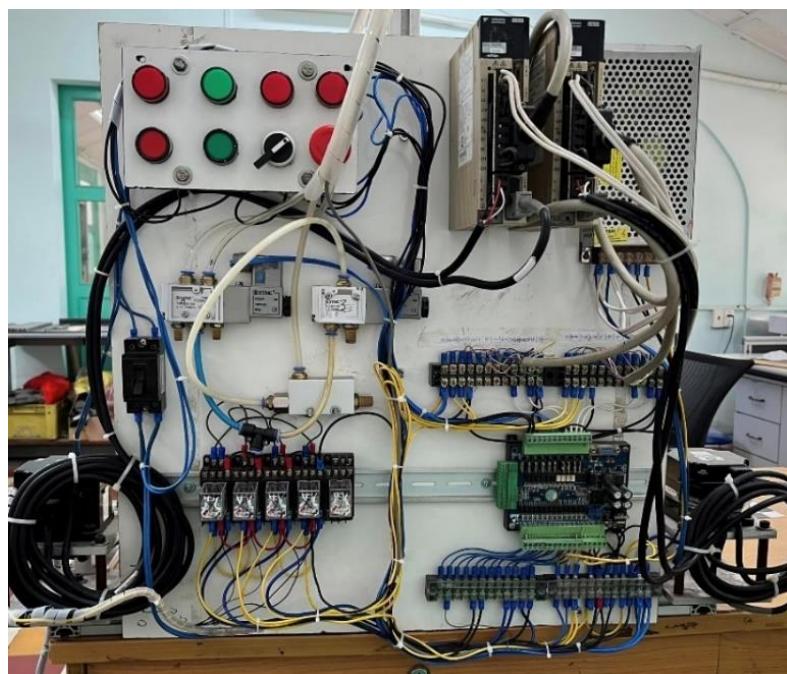
Hình 2.29: Sơ đồ đấu dây với nguồn 220V AC.



Hình 2.30: Sơ đồ đấu dây với nguồn 24V DC.

2.3. Kết quả

Sau quá trình tìm hiểu và khôi phục, robot chơi cờ như hình 2.31 có thể hoạt động bình thường. Cụ thể, khi PLC phát xung cho driver động cơ, động cơ quay và đầu công tắc di chuyển theo đúng phương trình động học đã tính toán. Khi đầu công tắc chạm cảm biến tiệm cận, chiều di chuyển tương ứng sẽ bị hạn chế. Xy lanh lên xuống và hút thả được quản cờ. Các nút nhấn, công tắc và đèn báo đều hoạt động bình thường.



a. Bảng điện của hệ thống.



b. Bàn máy H-bot.

Hình 2.31: Robot chơi cờ sau khi được khôi phục.

2.4. Kết luận:

Như vậy, hệ thống robot đã được phục hồi và thay thế các chi tiết cơ khí bị thiếu, hỏng; hệ thống điện đã được đấu nối hoàn chỉnh, các nguyên lí điều khiển được ghi nhận. Đây là cơ sở quan trọng để tiến hành gá đặt và hiệu chỉnh camera phục vụ công việc xử lý ảnh ở chương 3; phát triển giải thuật điều khiển đầu công tác ở chương 4; và cuối cùng là hoàn thành toàn bộ các bước trong quy trình điều khiển chơi cờ từ thuật toán chơi cờ đến điều khiển đầu công tác và các hệ thống đèn thông báo.

CHƯƠNG 3. XÁC ĐỊNH VỊ TRÍ VÀ TÊN CÁC QUÂN CỜ

Robot chơi cờ phải có khả năng xác định vị trí và tên của toàn bộ các quân cờ một cách nhanh chóng và ổn định trong suốt quá trình hoạt động. Để hiện thực mục tiêu đó, chương này sẽ lần lượt giải quyết 2 nhiệm vụ chính:

- Xác định tọa độ toàn bộ các quân cờ trên bàn cờ.
- Xác định tên của các quân cờ đã nhận dạng được.

Ngoài ra, robot cũng cần hoạt động tốt trong nhiều điều kiện ánh sáng khác nhau. Do đó triệt tiêu ảnh hưởng của ánh sáng thay đổi cũng là một nhiệm vụ sẽ được giải quyết.

3.1. Các khái niệm

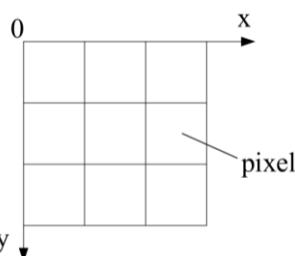
3.1.1. Xử lý ảnh

a. Ảnh kỹ thuật số và một số kỹ thuật biến đổi ảnh

- Điểm ảnh

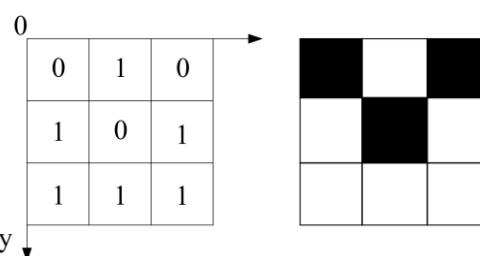
Hệ tọa độ kỹ thuật số là một hệ trực tọa độ của các đơn vị pixel, dùng cho việc xác định vị trí của điểm ảnh trong ảnh như hình 3.1. Do ảnh được lưu dưới dạng ma trận nên việc duyệt qua từng pixel trên ảnh cũng như duyệt qua các phần tử ma trận.

Điểm ảnh (pixel) là một phần tử của ảnh số tại tọa độ (x, y) với độ xám hoặc màu nhất định. Tọa độ $(0, 0)$ nằm bên trái trên.



Hình 3.1: Ma trận 2 chiều của điểm ảnh.

- Ảnh nhị phân

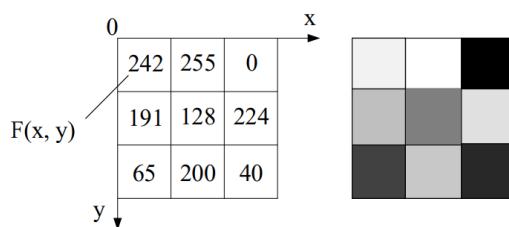


Hình 3.2: Mô hình biểu diễn ảnh nhị phân.

Ảnh nhị phân là ảnh chỉ có 2 mức đen trắng (hình 3.2). Giá trị xám mức 0 tương đương với mức trắng và mức 255 tương đương với mức đen.

- **Ảnh xám**

Ảnh xám trên hình 3.3 là ảnh có hai màu đen, trắng với mức xám hay cường độ sáng ở các điểm ảnh là khác nhau. Giá trị mức xám thay đổi từ độ sáng yếu nhất là 0 đến độ sáng mạnh nhất là 255. Khi một điểm ảnh mang giá trị mức xám 0 tức là tại đó có mức xám cao nhất nên màu hiển thị sẽ là màu đen, còn khi một điểm ảnh mang giá trị là 255 tức là tại đó có mức xám thấp nhất nên màu hiển thị sẽ là màu trắng.



Hình 3.3: Mô hình biểu diễn ảnh xám.

- **Phân ngưỡng ảnh xám**

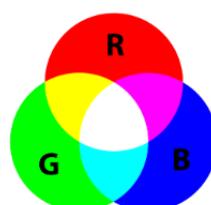
Nếu điểm ảnh có giá trị lớn hơn giá trị ngưỡng T thì nó thường được gán giá trị là mức 1 (255), ngược lại nhỏ hơn giá trị ngưỡng thì nó thường được gán giá trị là mức 0.

Trong thư viện OpenCV, hàm *threshold()* dùng để phân ngưỡng ảnh xám. Các tham số:

- *src*: ảnh đầu vào, là ảnh xám
- *threshold*: giá trị ngưỡng
- *maxval*: giá trị được gán nếu giá trị điểm ảnh lớn hơn giá trị ngưỡng
- *type*: loại phân ngưỡng

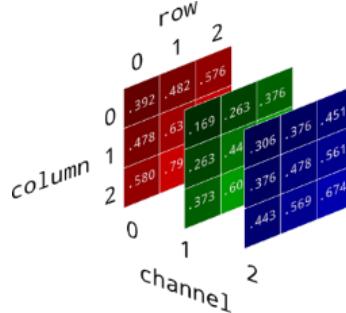
- **Ảnh màu RGB**

Ảnh màu RGB như hình 3.4 là ảnh sử dụng mô hình màu RGB [10], trong đó: đỏ (red), xanh lục (green) và xanh lam (blue) được tổ hợp với nhau để tạo ra các màu mới. Mỗi điểm màu được biểu diễn bởi bộ ba số (r, g, b) là giá trị của các màu trên.



Hình 3.4: Mô hình màu RGB.

Trên máy tính, ảnh màu được biểu diễn dưới dạng tensor 3 chiều như hình 3.5 [11], gồm 3 ma trận màu đỏ, lục, lam với kích thước là kích thước của ảnh chồng lên nhau.

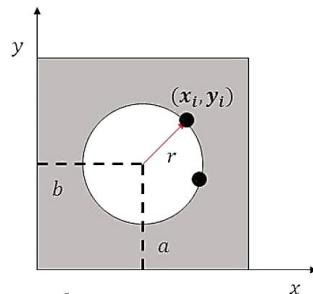


Hình 3.5: Mô hình ảnh RGB.

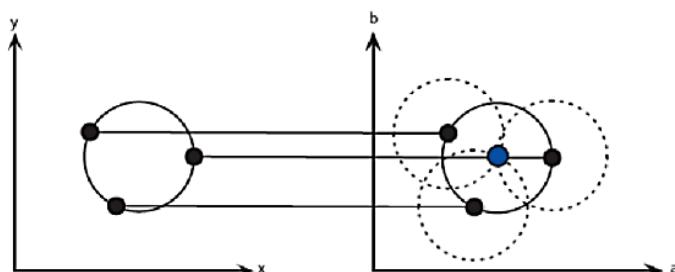
b. Giải thuật Hough circle

Giải thuật Hough là một kỹ thuật dùng phát hiện các đường thẳng, đường tròn hay đường có biên dạng bất kỳ trong ảnh xám. Khi dùng các chương trình máy tính để nhận diện một đối tượng nào đó có hình dạng bất kỳ, phương pháp hữu hiệu hiện nay là sử dụng biến đổi Hough, với điều kiện hình dạng này biểu diễn toán học được. Trong điều kiện nhận diện quân cờ, do quân cờ có dạng hình trụ tròn nên biến đổi Hough sẽ được thực hiện theo hướng nhận dạng đường tròn - Hough Circle [13].

Phương trình đường tròn [12]: $(x_i - a)^2 + (y_i - b)^2 = r^2$



Hình 3.6: Các tham số của phương trình đường tròn.



Hình 3.7: Chuyển về không gian Hough space.

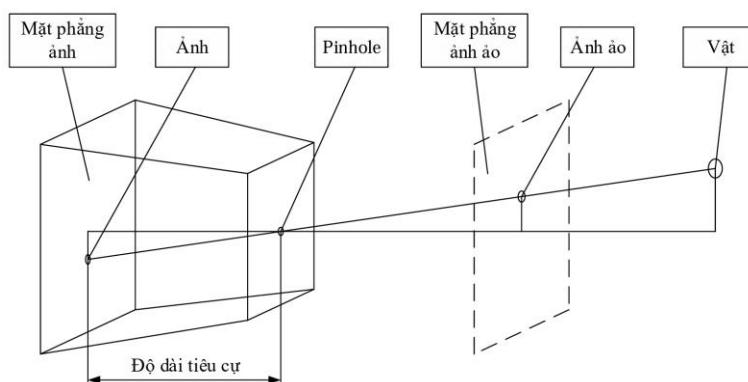
Từ không gian (x, y) như trên hình 3.6, khi chuyển về không gian (x_c, y_c) , mỗi điểm trên đường tròn trong tọa độ (x, y) sẽ tương ứng với một đường tròn trong tọa độ (x_c, y_c) . Để tìm tâm của đường tròn qua các điểm trong tọa độ (x, y) , mỗi điểm trên đường tròn được lấy làm tâm và vẽ một đường tròn bán kính r như hình 3.7. Các đường tròn này giao nhau tại một điểm, điểm đó là tâm của đường tròn đi qua tất cả các điểm đã đề cập.

Trong thư viện OpenCV, Hough circle được gọi qua hàm *HoughCircles()*. Giá trị trả về là bộ 3 số (x, y, r) , tương ứng với tọa độ và bán kính của đường tròn. Các tham số:

- *image*: ảnh đầu vào, có thể là ảnh 8-bit, ảnh nhị phân hoặc ảnh xám.
- *method*: phương pháp nhận diện.
- *minDist*: khoảng cách nhỏ nhất giữa tâm của các đường tròn.
- *minRadius, maxRadius*: bán kính nhỏ nhất, lớn nhất của đường tròn được phát hiện.

c. Biến dạng ảnh và hiệu chỉnh camera

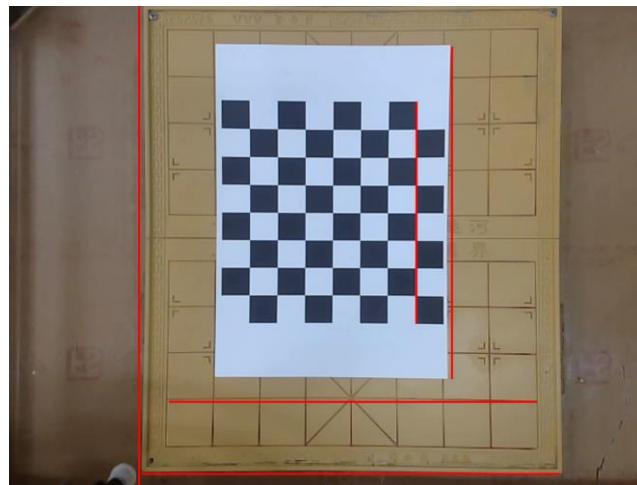
Các webcam thường là dạng camera lỗ kim, được mô tả như trong hình 3.8, chúng gây biến dạng đáng kể cho ảnh chụp được. Hai loại biến dạng chính là biến dạng xuyên tâm và biến dạng tiếp tuyến.



Hình 3.8: Mô hình camera lỗ kim.

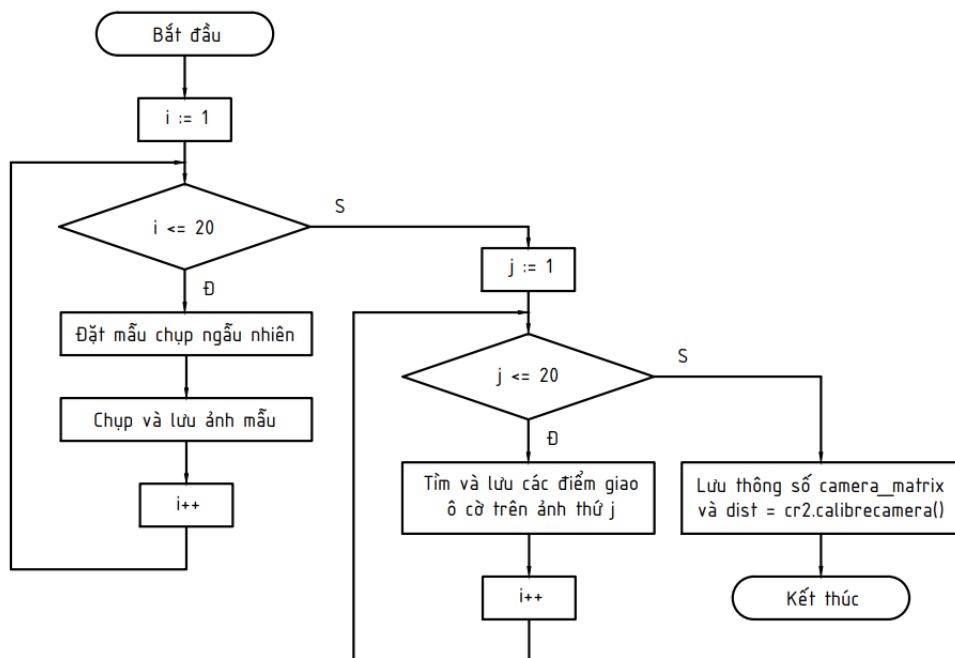
Biến dạng xuyên tâm làm cho hình ảnh bị cong, các điểm càng xa trung tâm hình ảnh thì biến dạng càng nhiều. Ví dụ như các đường chia ô, hay đường cạnh bàn cờ như trong hình 3.9 bị biến dạng, không khớp với các đường thẳng màu đỏ.

Biến dạng tiếp tuyến xảy ra do camera không được căn chỉnh hoàn toàn song song với mặt phẳng tạo ảnh. Vì vậy một số khu vực trong ảnh có khoảng cách giữa hai điểm bất kì gần hơn so với khoảng cách của chúng trong thực tế.



Hình 3.9: Ảnh bàn cờ bị biến dạng khi chụp qua camera.

Quá trình hiệu chuẩn camera (camera calibration) được thực hiện nhằm xác định các thông số nội và thông số ngoại của camera, từ đó loại bỏ ảnh hưởng của biến dạng trên ảnh, giúp cho việc tính toán các khoảng cách thực trả về kết quả chính xác.

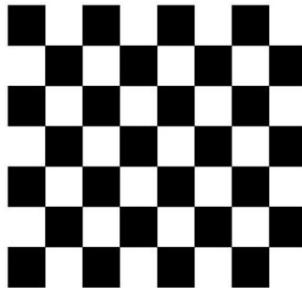


Hình 3.10: Lưu đồ quá trình hiệu chuẩn camera.

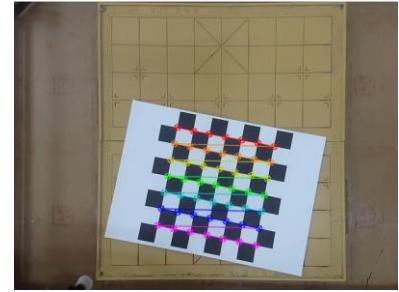
Trong thư viện OpenCV, hàm `calibrateCamera()` được sử dụng cho quá trình hiệu chuẩn. Các bước hiệu chuẩn được thể hiện như trong lưu đồ hình 3.10. Theo đó, khi sử dụng một mẫu chụp dạng caro như trên hình 3.11a, sau khi chụp ít nhất 10 ảnh của mẫu

ở các vị trí và hướng xoay khác nhau, vị trí các điểm giao nhận diện được trên mẫu này như hình 3.11b sẽ là bộ tham số đầu vào cho hàm hiệu chuẩn.

Kết quả được trả về là ma trận camera, hệ số biến dạng, các vector xoay và tịnh tiến.



(a) Mẫu chụp hiệu chỉnh camera.



(b) Vị trí các điểm giao được nhận diện.

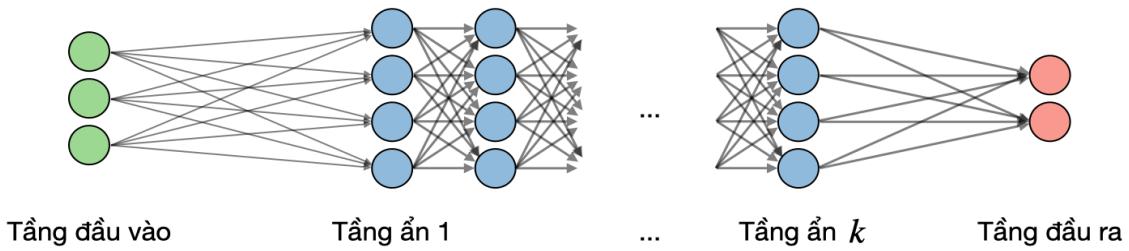
Hình 3.11: Hiệu chỉnh camera.

3.1.2. Mạng neural tích chập

a. Mạng neural

Mạng neural (neural network) là một hệ thống tính toán lấy cảm hứng từ sự hoạt động của các nơ-ron trong hệ thần kinh. Kết hợp với các kỹ thuật học sâu (deep learning), mạng neural đang trở thành một công cụ rất mạnh mẽ và mang lại hiệu quả rất tốt cho nhiều bài toán khó như nhận dạng ảnh, nhận dạng giọng nói hay xử lý ngôn ngữ tự nhiên.

- Kiến trúc của mạng neural [14] được mô tả như hình 3.11:



Hình 3.12: Kiến trúc mạng neural.

Với i là tầng thứ i của mạng, j là đơn vị ẩn thứ j của tầng, w, b, z là trọng số,

$$\text{bias và đầu ra thì: } z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$$

- Hàm kích hoạt (activation function):

Các hàm kích hoạt như trong bảng 3.1 được sử dụng ở phần cuối của các tầng ẩn để quyết định giá trị đầu ra của tầng ẩn đó.

Bảng 3.1: Các hàm kích hoạt thường được sử dụng.

| Sigmoid | Tanh | ReLU | Leaky ReLU |
|-------------------------------|--|---------------------|--|
| $g(z) = \frac{1}{1 + e^{-z}}$ | $g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ | $g(z) = \max(0, z)$ | $g(z) = \max(\epsilon z, z)$ Với $\epsilon \ll 1$ |

- Lỗi cross-entropy:

Trong các mô hình học máy, hàm lỗi được sử dụng như một công cụ để tối ưu mô hình.

Người ta sẽ huấn luyện mô hình theo hướng giảm giá trị lỗi.

Trong mạng neural, hàm lỗi cross-entropy $L(z, y)$ thường được định nghĩa như sau:

$$L(z, y) = -[y \log(z) + (1 - y) \log(1 - z)]$$

- Tốc độ học (learning rate):

Tốc độ học thường được kí hiệu α chỉ là tốc độ mà các trọng số được cập nhật. Phương thức phổ biến nhất hiện tại là Adam, phương thức thay đổi tốc độ học một cách phù hợp nhất có thể, cũng là phương thức được sử dụng trong luận văn này.

- Lan truyền ngược (backpropagation):

Phương thức này dùng để cập nhật trọng số trong mạng neural bằng cách tính toán đầu ra thực sự và đầu ra mong muốn.

Trọng số được cập nhật như sau:

$$w \leftarrow w - \alpha \frac{\partial L(z, y)}{\partial w}$$

Trong đó đạo hàm theo trọng số w được tính bằng cách sử dụng quy tắc chuỗi:

$$\frac{\partial L(z, y)}{\partial w} = \frac{\partial L(z, y)}{\partial \alpha} \times \frac{\partial \alpha}{\partial z} \times \frac{\partial z}{\partial w}$$

- Cập nhật trọng số:

Trong mạng neural, trọng số được cập nhật như sau:

- Bước 1: Lấy một batch dữ liệu huấn luyện (training data).
- Bước 2: Thực hiện lan truyền xuôi (forward propagation) để tính lỗi (loss).
- Bước 3: Lan truyền ngược lỗi để tính gradients.
- Bước 4: Sử dụng gradients để cập nhật trọng số của mạng.

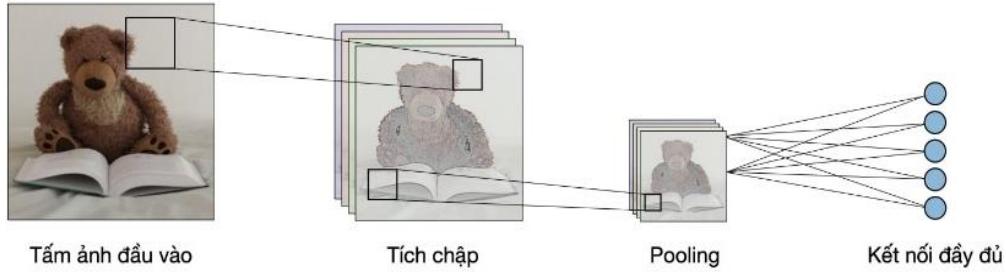
- Dropout:

Kỹ thuật này được sử dụng để tránh bị quá khớp (overfitting) tập dữ liệu huấn luyện bằng việc bỏ đi các đơn vị trong mạng neural. Các neurons hoặc bị bỏ đi bởi xác suất p hoặc được giữ lại với xác suất $1 - p$.

b. Mạng neural tích chập

Đối với các tác vụ trên hình ảnh, như là việc phân loại quân cờ trong luận văn này, mạng neural tích chập là một lựa chọn tốt. Nó sở hữu các tầng mạng hoạt động hiệu quả trên ảnh như tầng tích chập, tầng gộp được giới thiệu ở những phần tiếp theo.

- Kiến trúc của một mạng neural tích chập [14] được cấu thành như trên hình 3.12:



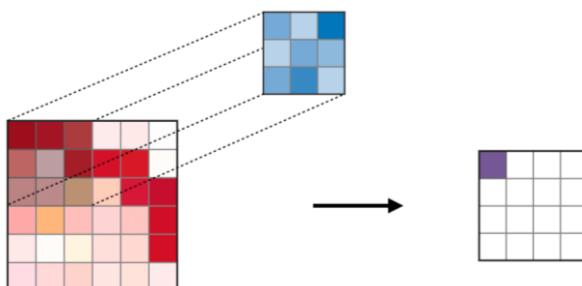
Hình 3.13: Minh họa kiến trúc một mạng neural tích chập.

Trong đó, tầng tích chập và tầng gộp có thể được hiệu chỉnh theo các siêu tham số (hyperparameters) được mô tả ở những phần tiếp theo.

- Các kiểu tầng:

- Tầng tích chập (convolution – CONV):

Tầng tích chập sử dụng các bộ lọc để thực hiện phép tích chập: dịch chuyển ma trận bộ lọc lần lượt qua tất cả các điểm ảnh trong ảnh. Ở mỗi lần dịch chuyển, thực hiện tính toán kết quả mới cho điểm ảnh đang xét bằng công thức tích chập (hình 3.14 [14]).



Hình 3.14: Áp dụng phép tích chập trên ảnh.

- Tầng gộp (pooling – POOL):

Tầng gộp thường được dùng giữa các tầng tích chập, để giảm kích thước dữ liệu nhưng vẫn giữ được các thuộc tính quan trọng. Việc giảm giữ liệu tính toán góp phần quan trọng trong việc giảm thời gian huấn luyện và kích cỡ mô hình.

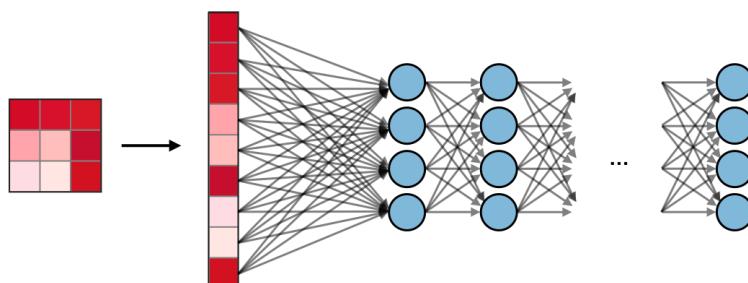
Bảng 3.2: Các kiểu pooling thường được sử dụng.

| Kiểu | Gộp cực đại | Gộp trung bình |
|-----------|--|--|
| Chức năng | Từng phép pooling chọn giá trị lớn nhất trong khu vực mà nó đang được áp dụng. | Từng phép pooling tính trung bình các giá trị trong khu vực mà nó đang được áp dụng. |
| Minh họa | | |
| Đặc điểm | Bảo toàn các đặc trưng đã phát hiện. Được sử dụng thường xuyên. | Giảm kích thước feature map. Được sử dụng trong mạng LeNet. |

Các phép gộp phổ biến được giới thiệu trong bảng 3.2 là gộp cực đại (max pooling) và gộp trung bình (average pooling), tương ứng trong đó giá trị lớn nhất và giá trị trung bình được lấy ra.

- Tầng kết nối đầy đủ (fully connected – FC):

Tầng này nhận đầu vào là các dữ liệu đã được “làm phẳng” (Flattened), mỗi đầu vào được kết nối đầy đủ đến tất cả neuron như hình 3.14 [14]. Trong mô hình mạng CNN, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng, ví dụ như độ chính xác của lớp.



Hình 3.15: Minh họa kiến trúc mạng kết nối đầy đủ.

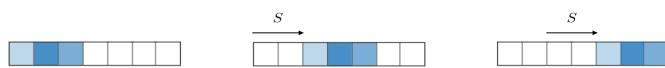
- Các siêu tham số của bộ lọc

- Sải bước (stride):

Đối với phép tích chập hoặc phép gộp, sải bước S kí hiệu số pixel mà cửa sổ sẽ di chuyển sau mỗi lần thực hiện phép tính (hình 3.15).

- 0 – đệm (zero – padding):

Quá trình tích chập sẽ làm ma trận hình ảnh bị giảm kích thước, vì vậy việc thêm vào một số hàng hay cột 0 sẽ đảm bảo giữ được kích thước ma trận đầu ra mong muốn. Bảng 3.4 giới thiệu các kiểu 0 – đệm thường được sử dụng.



Hình 3.16: Bộ lọc di chuyển S đơn vị sau mỗi lần thực hiện phép chập.

Bảng 3.3: Các kiểu 0 – đệm thường được sử dụng.

| Phương pháp | Valid | Same | Full |
|-------------|---|--|---|
| Giá trị | $P = 0$ | $P_{start} = \left\lceil \frac{S \left[\frac{I}{S} \right] - I + F - S}{2} \right\rceil$ $P_{end} = \left\lceil \frac{S \left[\frac{I}{S} \right] - I + F - S}{2} \right\rceil$ | $P_{start} \in [0, F - 1]$ $P_{end} = F - 1$ |
| Minh họa | | | |
| Mục đích | <ul style="list-style-type: none"> - Không sử dụng padding. - Bỏ phép tích chập cuối nếu số chiều không khớp. | <ul style="list-style-type: none"> - Sử dụng padding để làm cho feature map có kích thước $\left[\frac{I}{S} \right]$. - Kích thước đầu ra thuận lợi về mặt toán học. - Còn được gọi là “half – padding”. | <ul style="list-style-type: none"> - Padding tối đa sao cho các phép tích chập có thể được sử dụng tại các rìa của đầu vào. - Bộ lọc “thấy” được đầu vào từ đầu đến cuối. |

- Các hàm kích hoạt thường gặp

- Softmax:

Bước softmax được coi là một hàm logistic tổng quát, thường được sử dụng ở cuối kiến trúc mạng. Hàm lấy đầu vào là một vector chứa các giá trị $x \in R^n$ và cho ra một vector gồm các xác suất $p \in R^n$ được định nghĩa như sau:

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \text{ với } p_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

Cụ thể hơn, trong mô hình phân loại ở luận văn này, lớp softmax sẽ là lớp cuối cùng của mạng, trả về giá trị xác suất dự đoán của từng nhãn đối với hình ảnh mỗi quân cờ được đưa vào mạng.

- Rectified Linear Unit:

Tầng rectified linear unit (ReLU) là một hàm kích hoạt g được sử dụng trên tất cả các thành phần của mạng, mục đích là tăng tính phi tuyến cho mạng.

Sau khi giới thiệu kiến thức cơ bản, các mục tiếp theo sẽ trình bày về quá trình xác định tọa độ ảnh và biến đổi để nhận được tọa độ thực của quân cờ; chuẩn bị dữ liệu, xây dựng cấu trúc mô hình học sâu và huấn luyện mô hình để dự đoán tên quân cờ.

c. Huấn luyện mô hình mạng và đánh giá

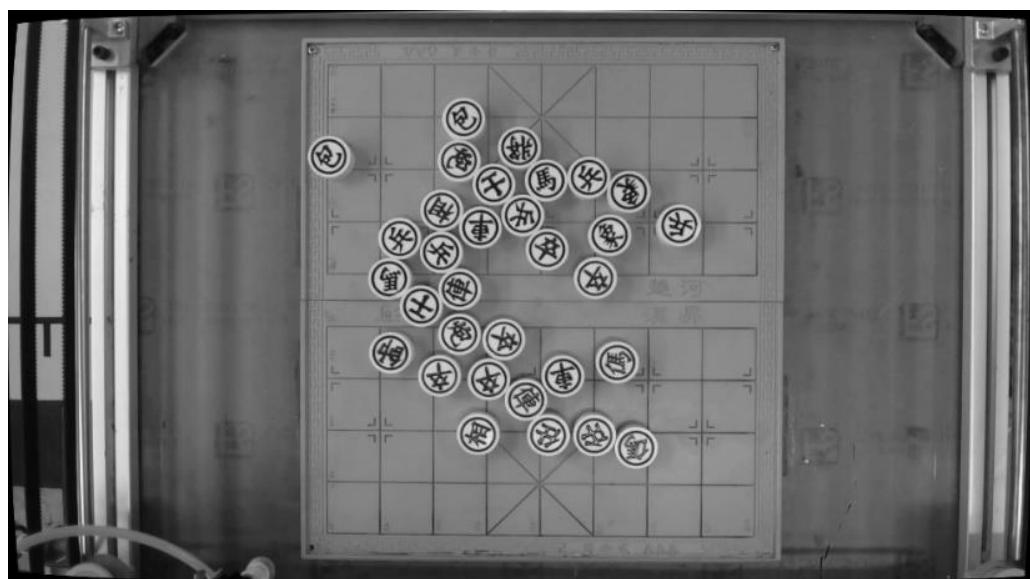
- Epoch: Số vòng lặp mà mô hình sẽ duyệt toàn bộ tập dữ liệu huấn luyện để cập nhật trọng số của nó.
- Mini-batch gradient descent: Trong quá trình huấn luyện, việc cập nhật trọng số thường không dựa trên toàn bộ tập huấn luyện cùng một lúc do độ phức tạp tính toán hoặc một điểm dữ liệu nhiều. Thay vào đó, bước cập nhật được thực hiện trên các lô nhỏ (mini-batch), trong đó số lượng điểm dữ liệu trong một lô (batch) là một siêu tham số (hyperparameter) mà chúng ta có thể điều chỉnh.
- Dừng sớm (early stopping): dừng quá trình huấn luyện một khi mất mát trên tập thẩm định (validation) đạt đến một ngưỡng nào đó hoặc bắt đầu tăng.

3.2. Chụp và hiệu chỉnh hình ảnh bàn cờ

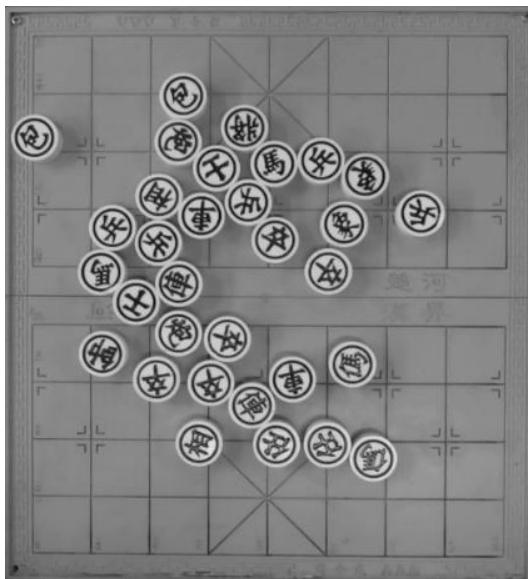
Ảnh toàn bộ khu vực bàn cờ ban đầu sẽ được chụp từ camera như hình 3.16a. Từ các thông số nội và ngoại trong quá trình hiệu chuẩn camera, ảnh được hiệu chỉnh để loại bỏ biến dạng trên khu vực bàn cờ. Sau đó được chuyển về dạng ảnh nhị phân như trong hình 3.16b. Vì camera đặt cố định và ảnh này được lưu trữ dưới dạng ma trận, chương trình nhận diện quân cờ sẽ được cài đặt sẵn tọa độ của các góc bàn cờ, từ đó cắt bỏ đi trên ảnh các phần biên thừa, chỉ giữ lại phần bàn cờ bàn cờ như hình 3.16c.



a. Ảnh khu vực bàn cờ.



b. Ảnh khu vực bàn cờ được hiệu chỉnh.



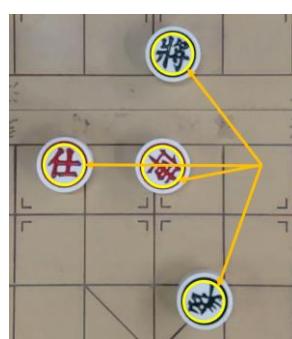
c. Ảnh bàn cờ được cắt ra.

Hình 3.17: Quá trình chụp và hiệu chỉnh hình ảnh bàn cờ.

3.3. Xác định tọa độ các quân cờ trong hệ tọa độ bàn máy

Ban đầu chương trình sẽ tìm tọa độ quân cờ trong hệ tọa độ của ảnh chụp được, sau đó biến đổi để trả về tọa độ trong hệ tọa độ bàn máy nhằm phục vụ việc di chuyển của đầu công tác.

Tọa độ quân cờ ở đây là tọa độ tâm của đường tròn mặt quân cờ. Các quân cờ tướng được sử dụng trong luận văn này có đặc điểm là có khắc một đường tròn bao quanh biểu tượng quân. Do đó, phần mềm điều khiển có thể xác định vị trí quân cờ thông qua việc xác định vị trí những đường tròn này trên ảnh chụp từ camera (từ giờ sẽ gọi là “đường tròn quân cờ”) như hình 3.18.



Hình 3.18: Phát hiện quân cờ thông qua đường tròn khắc trên nó.

Từ hình ảnh đã thu được, áp dụng giải thuật Hough circle để xác định các đường tròn trên ảnh: vị trí theo tọa độ ảnh và bán kính mỗi đường tròn đó.

3.3.1. Sử dụng giải thuật Hough circle để thu được tọa độ quân cờ

Kích thước bàn cờ là $411 \times 368\text{ mm}$, bán kính đường tròn quân cờ là 23.6 mm . Khi đó, bán kính đường tròn quân cờ cần nhận diện trên ảnh được tính như sau:

```
radius = round(0.028*image_height)
```

Ảnh dù đã qua hiệu chỉnh vẫn có thể bị biến dạng nhỏ, nên 2 thông số cận trên và cận dưới của bán kính đường tròn quân cờ là minRadius và maxRadius cũng được tăng giảm nhẹ để thu được trọng số các đường tròn (chọn sai số 3%). Khoảng cách nhỏ nhất giữa các đường tròn - minDist được chọn là 2 lần bán kính. Bảng 3.4 tổng hợp các tham số của hàm Houghcircle.

Bảng 3.4: Các tham số hàm Houghcircle()

| Tham số | Giá trị |
|-----------|---------------------------|
| radius | round(0.028*image_height) |
| minRadius | round(0.97*radius) |
| maxRadius | round(1.03*radius) |
| minDist | 2*radius |

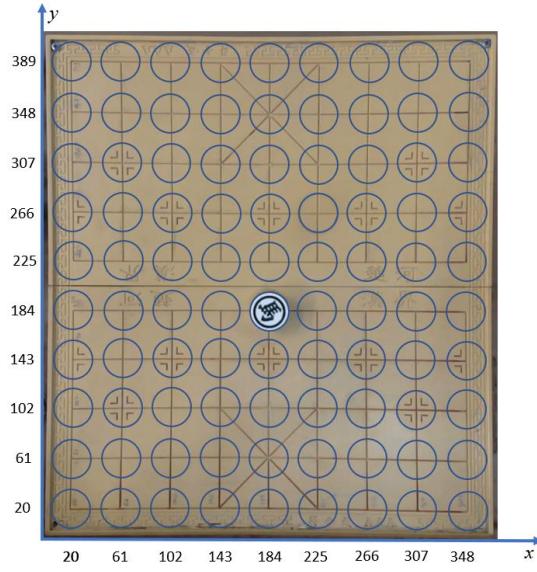
Kết quả trả về từ hàm trên là tọa độ và bán kính từng đường tròn quân cờ. Dữ liệu này sẽ được lưu lại để phục vụ chương trình nhận dạng tên của từng quân cờ.

3.3.2. Đánh giá hoạt động xác định tọa độ quân cờ

Phần này đánh giá bằng cách so sánh kết quả quá trình xác định vị trí của quân cờ từ ảnh với vị trí quân cờ trong thực tế ở nhiều trường hợp khác nhau.

Quân cờ sẽ lần lượt được đặt tại 90 điểm giao giữa các đường chia ô của bàn cờ như trên hình 3.19. Đây là 90 trường hợp được đánh giá. Bằng cách vẽ một đường tròn có tâm là mỗi điểm giao, bán kính 31 mm và đặt quân cờ chồng khít lên đường tròn đó thì tâm của đường tròn quân cờ sẽ gần như trùng với điểm giao.

Mỗi lần quân cờ được đặt tại các điểm giao, giá trị tọa độ của quân cờ được ghi lại để so sánh với giá trị tọa độ được tính toán từ ảnh. Kết quả của quá trình này được tổng hợp trên bảng 3.5. Bảng này cho thấy độ lệch giữa giá trị tính toán của phần mềm từ ảnh và giá trị thực tế.



Hình 3.19: Các vị trí đánh giá độ lệch tọa độ quân cờ.

Bảng 3.5: Bảng so sánh độ lệch giữa tọa độ quân cờ thực tế và tọa độ quân cờ tính toán được trên ảnh.

| STT | Hoành độ ảnh | Tung độ ảnh | Hoành độ thực tế | Tung độ thực tế | Độ lệch hoành độ | Độ lệch tung độ | Độ lệch tâm |
|-----|-----------------|----------------|---------------------|--------------------|---------------------|--------------------|----------------|
| 1 | 142 | 182 | 143 | 184 | 1 | 2 | 2,2 |
| 2 | 141 | 140 | 143 | 143 | 2 | 3 | 3,6 |
| 3 | 18 | 141 | 20 | 143 | 2 | 2 | 2,8 |
| 4 | 19 | 227 | 20 | 225 | 1 | 2 | 2,2 |
| 5 | 59 | 183 | 61 | 184 | 2 | 1 | 2,2 |
| 6 | 182 | 183 | 184 | 184 | 2 | 1 | 2,2 |
| 7 | 101 | 140 | 102 | 143 | 1 | 3 | 3,2 |
| 8 | 222 | 183 | 225 | 184 | 3 | 1 | 3,2 |
| 9 | 100 | 182 | 102 | 184 | 2 | 2 | 2,8 |
| 10 | 59 | 228 | 61 | 226 | 2 | 2 | 2,8 |
| 11 | 265 | 182 | 266 | 184 | 1 | 2 | 2,2 |
| 12 | 182 | 225 | 184 | 225 | 2 | 0 | 2,0 |
| 13 | 140 | 227 | 143 | 225 | 3 | 2 | 3,6 |
| 14 | 222 | 226 | 225 | 225 | 3 | 1 | 3,2 |
| 15 | 58 | 142 | 61 | 143 | 3 | 1 | 3,2 |
| 16 | 307 | 183 | 307 | 184 | 0 | 1 | 1,0 |
| 17 | 99 | 224 | 102 | 225 | 3 | 1 | 3,2 |
| 18 | 183 | 268 | 184 | 266 | 1 | 2 | 2,2 |

| | | | | | | | |
|----|-----|-----|-----|-----|---|---|-----|
| 19 | 306 | 226 | 307 | 225 | 1 | 1 | 1,4 |
| 20 | 346 | 183 | 348 | 184 | 2 | 1 | 2,2 |
| 21 | 347 | 225 | 348 | 225 | 1 | 0 | 1,0 |
| 22 | 264 | 227 | 266 | 225 | 2 | 2 | 2,8 |
| 23 | 225 | 268 | 225 | 266 | 0 | 2 | 2,0 |
| 24 | 266 | 267 | 266 | 266 | 0 | 1 | 1,0 |
| 25 | 225 | 309 | 225 | 307 | 0 | 2 | 2,0 |
| 26 | 19 | 182 | 21 | 184 | 2 | 2 | 2,8 |
| 27 | 266 | 309 | 266 | 307 | 0 | 2 | 2,0 |
| 28 | 308 | 268 | 307 | 266 | 1 | 2 | 2,2 |
| 29 | 345 | 268 | 348 | 266 | 3 | 2 | 3,6 |
| 30 | 351 | 306 | 348 | 307 | 3 | 1 | 3,2 |
| 31 | 185 | 309 | 184 | 307 | 1 | 2 | 2,2 |
| 32 | 307 | 309 | 307 | 307 | 0 | 2 | 2,0 |
| 33 | 18 | 99 | 20 | 102 | 2 | 3 | 3,6 |
| 34 | 19 | 21 | 20 | 20 | 1 | 1 | 1,4 |
| 35 | 58 | 100 | 61 | 102 | 3 | 2 | 3,6 |
| 36 | 347 | 19 | 348 | 20 | 1 | 1 | 1,4 |
| 37 | 58 | 59 | 61 | 61 | 3 | 2 | 3,6 |
| 38 | 101 | 59 | 102 | 61 | 1 | 2 | 2,2 |
| 39 | 145 | 20 | 143 | 20 | 2 | 0 | 2,0 |
| 40 | 307 | 19 | 307 | 20 | 0 | 1 | 1,0 |
| 41 | 223 | 18 | 225 | 20 | 2 | 2 | 2,8 |
| 42 | 142 | 60 | 143 | 61 | 1 | 1 | 1,4 |
| 43 | 265 | 19 | 266 | 20 | 1 | 1 | 1,4 |
| 44 | 224 | 58 | 225 | 61 | 1 | 3 | 3,2 |
| 45 | 141 | 100 | 143 | 102 | 2 | 2 | 2,8 |
| 46 | 183 | 20 | 184 | 20 | 1 | 0 | 1,0 |
| 47 | 267 | 58 | 266 | 61 | 1 | 3 | 3,2 |
| 48 | 183 | 99 | 184 | 102 | 1 | 3 | 3,2 |
| 49 | 60 | 20 | 61 | 20 | 1 | 0 | 1,0 |
| 50 | 101 | 101 | 102 | 102 | 1 | 1 | 1,4 |
| 51 | 347 | 59 | 348 | 61 | 1 | 2 | 2,2 |
| 52 | 348 | 101 | 348 | 102 | 0 | 1 | 1,0 |
| 53 | 223 | 100 | 225 | 102 | 2 | 2 | 2,8 |
| 54 | 183 | 141 | 184 | 143 | 1 | 2 | 2,2 |
| 55 | 184 | 58 | 184 | 61 | 0 | 3 | 3,0 |

| | | | | | | | |
|------------|-----|-----|-----|-----|-----|-----|-----|
| 56 | 267 | 101 | 266 | 102 | 1 | 1 | 1,4 |
| 57 | 265 | 142 | 266 | 143 | 1 | 1 | 1,4 |
| 58 | 308 | 101 | 307 | 102 | 1 | 1 | 1,4 |
| 59 | 349 | 144 | 348 | 143 | 1 | 1 | 1,4 |
| 60 | 224 | 141 | 225 | 143 | 1 | 2 | 2,2 |
| 61 | 306 | 58 | 307 | 61 | 1 | 3 | 3,2 |
| 62 | 19 | 61 | 20 | 61 | 1 | 0 | 1,0 |
| 63 | 101 | 18 | 102 | 20 | 1 | 2 | 2,2 |
| 64 | 309 | 141 | 307 | 143 | 2 | 2 | 2,8 |
| 65 | 18 | 268 | 20 | 266 | 2 | 2 | 2,8 |
| 66 | 18 | 310 | 20 | 307 | 2 | 3 | 3,6 |
| 67 | 58 | 390 | 61 | 389 | 3 | 1 | 3,2 |
| 68 | 19 | 392 | 20 | 389 | 1 | 3 | 3,2 |
| 69 | 58 | 266 | 61 | 266 | 3 | 0 | 3,0 |
| 70 | 99 | 309 | 102 | 307 | 3 | 2 | 3,6 |
| 71 | 101 | 267 | 102 | 266 | 1 | 1 | 1,4 |
| 72 | 18 | 351 | 20 | 348 | 2 | 3 | 3,6 |
| 73 | 59 | 310 | 61 | 307 | 2 | 3 | 3,6 |
| 74 | 60 | 350 | 61 | 348 | 1 | 2 | 2,2 |
| 75 | 143 | 310 | 143 | 307 | 0 | 3 | 3,0 |
| 76 | 224 | 351 | 225 | 348 | 1 | 3 | 3,2 |
| 77 | 183 | 391 | 184 | 389 | 1 | 2 | 2,2 |
| 78 | 267 | 390 | 266 | 389 | 1 | 1 | 1,4 |
| 79 | 266 | 351 | 266 | 348 | 0 | 3 | 3,0 |
| 80 | 143 | 350 | 143 | 348 | 0 | 2 | 2,0 |
| 81 | 184 | 351 | 184 | 348 | 0 | 3 | 3,0 |
| 82 | 349 | 351 | 348 | 348 | 1 | 3 | 3,2 |
| 83 | 308 | 392 | 307 | 389 | 1 | 3 | 3,2 |
| 84 | 348 | 392 | 348 | 389 | 0 | 3 | 3,0 |
| 85 | 102 | 393 | 102 | 389 | 0 | 4 | 4,0 |
| 86 | 142 | 269 | 143 | 266 | 1 | 3 | 3,2 |
| 87 | 103 | 351 | 102 | 348 | 1 | 3 | 3,2 |
| 88 | 141 | 390 | 143 | 389 | 2 | 1 | 2,2 |
| 89 | 307 | 351 | 307 | 348 | 0 | 3 | 3,0 |
| 90 | 224 | 390 | 225 | 389 | 1 | 1 | 1,4 |
| Trung bình | | | | | 1,3 | 1,8 | 2,5 |

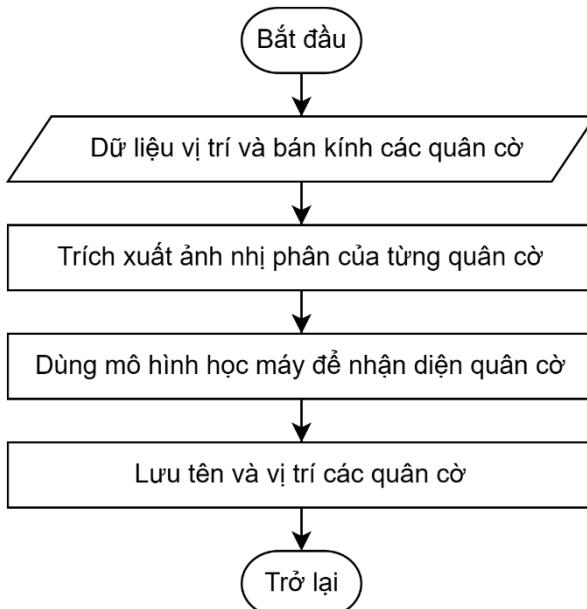
Theo bảng 3.5, độ lệch trung bình của tọa độ quân cờ là 2,5 mm. Giá trị này là nhỏ so với đường kính quân cờ là 31 mm. Điều này cho phép đầu công tác di chuyển tới đúng vị trí của các quân cờ trên bàn cờ và thực hiện các thao tác đi quân, ăn quân trong quá trình xếp cờ và chơi cờ.

3.4. Xác định tên của các quân cờ

Bàn cờ tướng gồm có 7 loại quân cờ: xe, mã, tượng, sĩ, tướng, pháo và tốt. Quân cờ trên bàn cờ cũng có thể ở hướng xoay bất kì nên thuật toán cần có khả năng nhận diện quân cờ trong những trạng thái đó.

Yêu cầu xác định tên từng quân cờ trong luận văn này có nét tương đồng nhất định với bài toán nhận diện chữ viết tay MNIST [15] nổi tiếng, vì vậy việc huấn luyện một mô hình mạng neural học sâu để giải quyết vấn đề là lựa chọn khả thi và đáng tin cậy.

Đối với hình ảnh quân cờ cần nhận diện, ở các hướng xoay và điều kiện ánh sáng khác nhau, các hình ảnh này sẽ có màu sắc và độ đậm nhạt sáng tối khác nhau. Do đó việc đưa ảnh về dạng ảnh nhị phân sẽ triệt tiêu được ảnh hưởng của môi trường một cách hiệu quả. Hơn nữa, dữ liệu ảnh quân cờ ở dạng nhị phân cũng giảm số chiều của ảnh, giảm khối lượng tích toán và các tham số mô hình, giúp chương trình nhận diện hoạt động nhanh hơn.



Hình 3.20: Lưu đồ quá trình xác định tên từng quân cờ đã nhận dạng.

Ý tưởng xác định tên từng quân cờ được thể hiện trong lưu đồ trên hình 3.20. Từ dữ liệu vị trí và bán kính quân cờ đã xác định được, tiến hành trích xuất hình ảnh của riêng từng quân cờ, rồi nhận diện quân cờ đó thông qua mô hình mạng neuron đã được huấn luyện từ trước. Như vậy, phần này có 2 nhiệm vụ như sau:

- Trích xuất ảnh nhị phân từng quân cờ.
- Huấn luyện mô hình học máy để nhận diện quân cờ.



Hình 3.21: *Ảnh quân cờ sau khi được cắt ra từ bàn cờ và biến đổi.*

Đối với nhiệm vụ trích xuất ảnh nhị phân từng quân cờ, từ vị trí tâm và bán kính mỗi quân cờ cắt ra một ảnh hình vuông chứa toàn bộ quân cờ đó. Sau đó sử dụng hàm `threshold()` để biến đổi ảnh quân cờ thành ảnh nhị phân như hình 3.21.

Trong điều kiện phòng thí nghiệm nơi tiến hành luận văn, từ thực nghiệm chọn được tham số ngưỡng `thresh = 150`. Trong các trường hợp ánh sáng thay đổi nhiều có thể điều chỉnh tham số này phù hợp từng hoàn cảnh. Từ thực nghiệm, lựa chọn tham số `thresh` với một số môi trường như trong bảng 3.6.

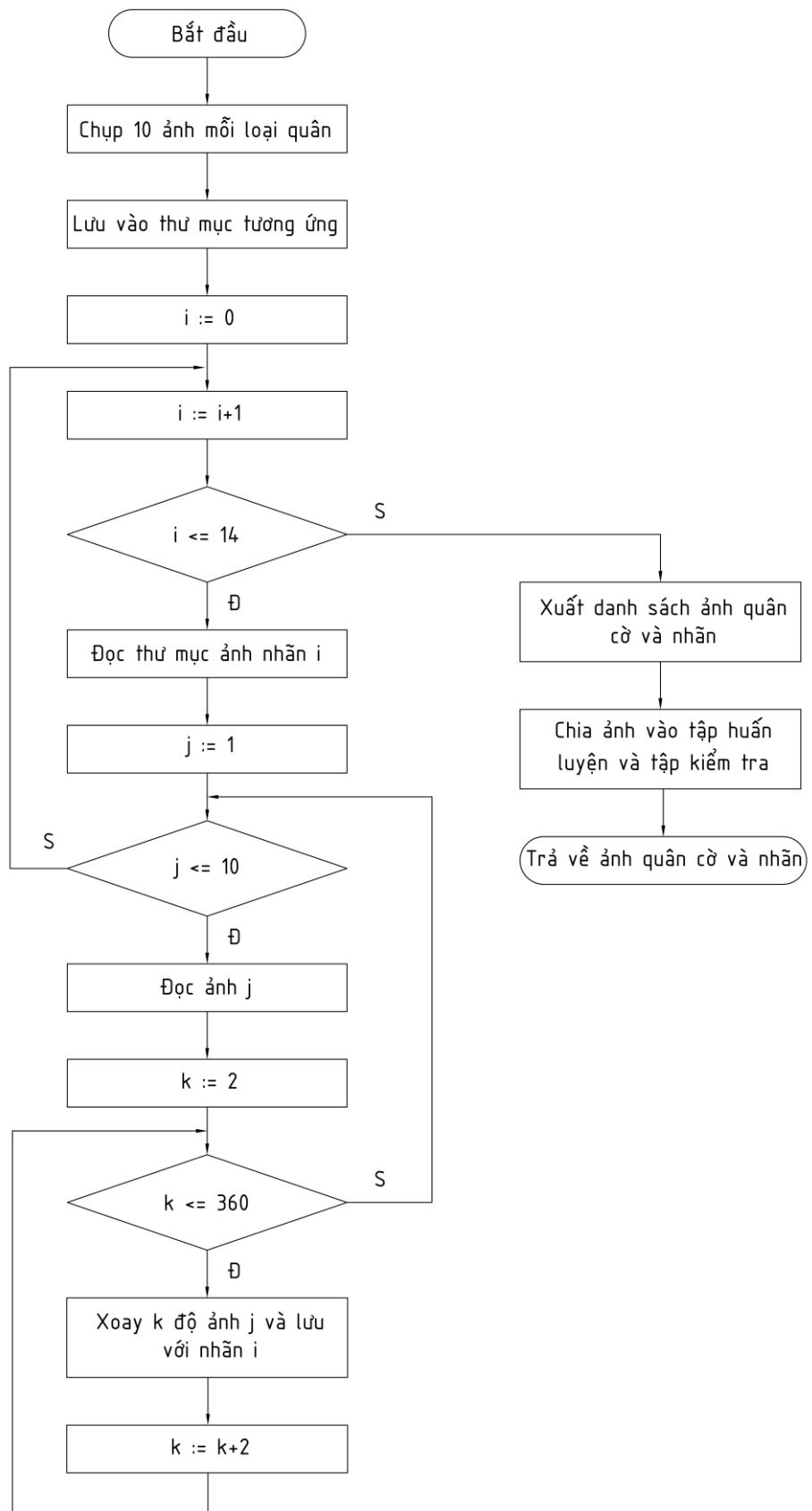
Bảng 3.6: *lựa chọn tham số thresh trong một số môi trường.*

| Môi trường | <i>Trong nhà</i> | <i>Ngoài trời, trong bóng râm</i> | <i>Ngoài trời, trực tiếp dưới nắng</i> |
|-----------------------------|--------------------------------|---|--|
| Ví dụ | Phòng thí nghiệm P202 – C1. | Ghế đá ngoài trời tòa B11, trong mái hiên. | Ngoài trời trước xưởng C1. |
| Giá trị <code>thresh</code> | 150 - 170 | 150 | 180 |

3.4.1. Chuẩn bị dữ liệu để huấn luyện mô hình

Quá trình chuẩn bị dữ liệu để huấn luyện mô hình được mô tả trong lưu đồ hình 3.22, bao gồm 2 bước chính như sau:

- Lấy dữ liệu hình ảnh quân cờ.
- Sinh thêm dữ liệu.



Hình 3.22: Lưu đồ quá trình lấy dữ liệu để huấn luyện mô hình.

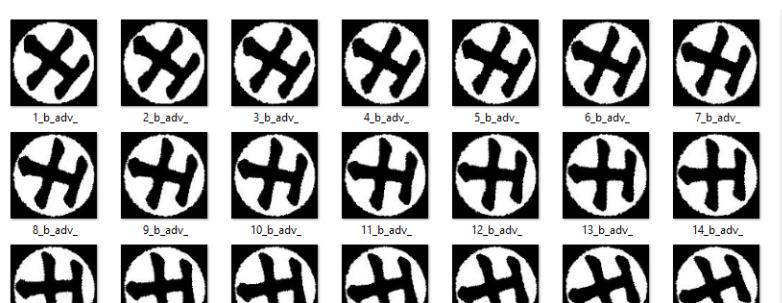
a. Lấy dữ liệu để huấn luyện mô hình

Dữ liệu đầu vào để huấn luyện mô hình là các ảnh nhị phân của từng quân cờ. Sau đó, phân loại tập ảnh thu được ở trên vào các thư mục theo tên từng quân để phục vụ việc đánh nhãn dữ liệu.

b. Tạo thêm dữ liệu ảnh

Thực tế trên bàn cờ, quân cờ có thể được đặt xoay theo mọi hướng không xác định. Quá trình lấy dữ liệu đã đề cập ở trên cũng không thể nào cung cấp đầy đủ các trường hợp trạng thái của mỗi quân cờ, vì vậy việc sinh thêm dữ liệu phục vụ cho hoạt động huấn luyện mô hình là cần thiết.

Ở đây, mỗi ảnh đã thu được sẽ tiếp tục sinh ra 179 ảnh nữa qua từng lần xoay ảnh đó 2° như ví dụ trên hình 3.23.



Hình 3.23: Dữ liệu quân cờ được sinh ra qua quá trình xoay.

3.4.2. Xác định và xây dựng mô hình

a. Giới thiệu ý tưởng

Trong luận văn này, mô hình học sâu được sử dụng mạng neural tích chập (convolutional neural network – CNN) để nhận dạng các quân cờ. Đây là một trong những mô hình phổ biến nhất để nhận dạng và phân loại hình ảnh.

Về kỹ thuật, mô hình CNN dùng để huấn luyện và kiểm tra, mỗi hình ảnh đầu vào sẽ được biến đổi qua một loạt các lớp tích chập với các bộ lọc (kernels), tổng hợp lại qua lớp kết nối đầy đủ (full connected layer) và áp dụng hàm softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1.

Sau khi huấn luyện mạng CNN, ta có thể dùng nó để dự đoán tên của mỗi quân cờ trích xuất được trên ảnh (gọi là nhãn). Đối với hình ảnh mỗi quân cờ, mô hình trả về tập giá

trị là xác suất dự đoán nhãn của nó. Cuối cùng, ta chọn nhãn có xác suất cao nhất làm nhãn quân cờ đó.

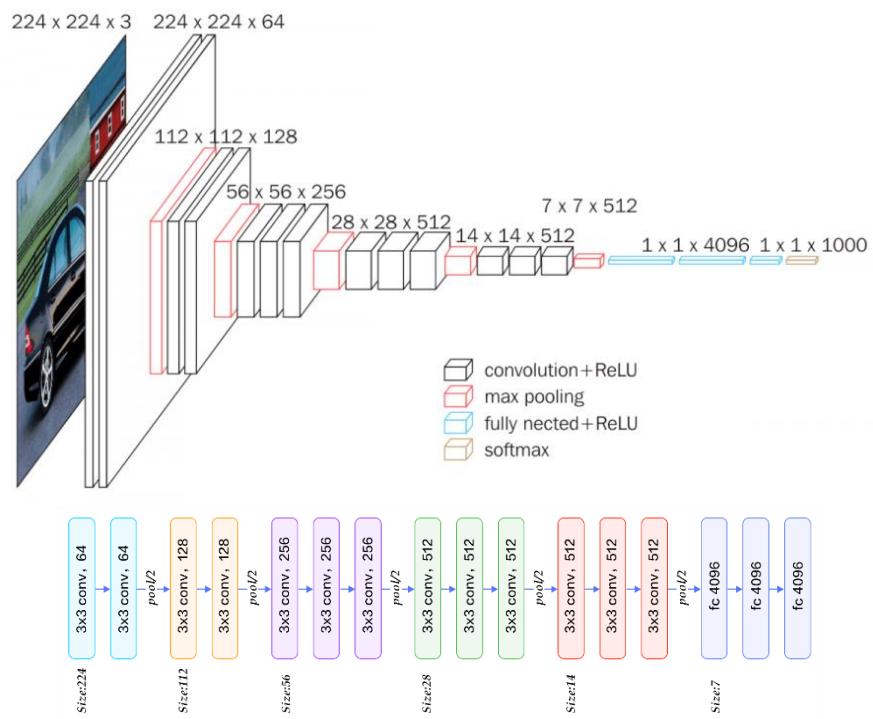
b. Lựa chọn cấu trúc mô hình mạng CNN

Cấu trúc CNN được sử dụng tương tự mạng VGG gốc (VGG – 11) [16].

Đối với mạng tích chập cổ điển, khối cơ bản của mạng là một chuỗi các tầng sau đây: (i) một tầng tích chập (với phần đệm để duy trì độ phân giải), (ii) một tầng phi tuyến như ReLU, (iii) một tầng gộp như gộp cực đại.

Đối với mạng VGG, một khối VGG gồm một chuỗi các tầng tích chập, tiếp nối bởi một tầng gộp cực đại để giảm chiều không gian. Trong bài báo gốc của VGG [6], các tác giả sử dụng tích chập với các hạt nhân 3×3 và tầng gộp cực đại 2×2 với sải bước bằng 2. (giảm một nửa độ phân giải sau mỗi khối).

Mạng VGG gốc như trên hình 3.24 có 5 khối tích chập, trong đó hai khối đầu tiên bao gồm một tầng tích chập ở mỗi khối, ba khối còn lại chứa hai tầng tích chập ở mỗi khối. Khối đầu tiên có 64 kênh đầu ra, mỗi khối tiếp theo nhân đôi số kênh đầu ra cho tới khi đạt giá trị 512. Vì mạng này sử dụng 8 tầng tích chập và 3 tầng kết nối đầy đủ nên nó thường được gọi là VGG-11.



Hình 3.24: Minh họa cấu trúc mạng VGG.

Trong việc lập trình mô hình mạng, luận văn sử dụng các API Tensorflow Keras. [17]

- Khởi tạo mô hình: `model = Sequential()`.
- Các tầng sử dụng: `Conv2D(conv)`, `MaxPooling2D(maxpooling)`, `Dense(fc)`.

Các tham số:

- *filter*: số bộ lọc sử dụng.
- *kernel_size*: kích thước cửa sổ lọc.
- *padding*: kiểu đệm.
- *input_shape*: kích thước đầu vào.
- *activation*: hàm kích hoạt.
- Compiler: hàm `compile()`
 - Hàm mất mát: `loss = "categorical_crossentropy"`
 - Hàm tối ưu: `optimizer = "Adam"`
- Huấn luyện mô hình: `model.fit()`.

3.4.3. Huấn luyện mô hình và đánh giá

Sau khi đã chuẩn bị dữ liệu và xác định được cấu trúc mô hình mạng sử dụng, phần này trình bày về quá trình huấn luyện mô hình, cũng như đưa ra một số đánh giá sơ bộ.

a. Huấn luyện mô hình

Như đã trình bày ở mục 3.3.2, luận văn sử dụng API Keras, để huấn luyện mô hình, dùng hàm `fit()` với các tham số như sau:

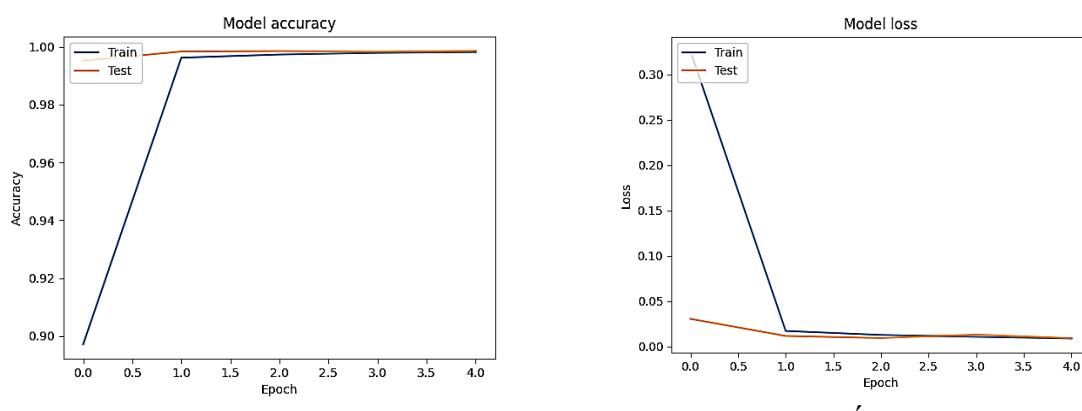
- *x_train* và *y_train*: *x_train* là tập huấn luyện và *y_train* là nhãn tương ứng. Các txập này được lấy từ dữ liệu đã chuẩn bị ở mục 3.2, trong trường hợp này là 80% dữ liệu, 20% còn lại phục vụ tập kiểm tra (*x_test*).
- *epoch* = 8. Số epoch cũng không cần quá chính xác, bởi ở đây mô hình sẽ sử dụng kĩ thuật dừng sớm.
- *callbacks*: sử dụng dừng sớm. Hàm này được gọi với các tham số như sau:
 - o *monitor*: giá trị quan sát. Chọn độ chính xác kiểm chứng (validation accuracy – “`val_accuracy`”).

- `min_delta`: giá trị thay đổi nhỏ nhất của giá trị quan sát qua các epochs, nếu thay đổi nhỏ hơn giá trị này trong suốt một số epoch thì dừng huấn luyện. Chọn `min_delta = 0.0001`.
 - `patience`: số epoch mà quá trình huấn luyện không được cải thiện. Chọn `patience = 3`.
 - `mode`: chế độ, chọn “`max`” – mô hình sẽ ngừng huấn luyện khi giá trị quan sát ngừng tăng lên.
- `validation_split`: tỉ lệ dữ liệu được chọn cho tập kiểm định. Chọn `0.2`, tức là 20%.

b. Dánh giá mô hình

Mô hình dừng huấn luyện sau 4 epochs, mỗi epoch 1571 vòng lặp, đạt độ chính xác kiểm định (validation accuracy) là 100%.

Có thể thấy dữ liệu huấn luyện có chất lượng khá tốt nên độ chính xác mô hình tăng mạnh trên tập huấn luyện, tập kiểm tra đạt độ chính xác > 99% ngay từ epoch đầu tiên như trên hình 3.25a. Hàm mất mát trên hình 3.25b cũng giảm nhanh và ổn định ở giá trị mong muốn.



a. Độ chính xác.

b. Mất mát.

Hình 3.25: Độ chính xác và giá trị mất mát của mô hình trên tập huấn luyện và tập kiểm tra qua các epochs.

3.4.4. Đánh giá hoạt động của chương trình trong một số môi trường khác nhau

Phần này đánh giá hoạt động nhận diện quân cờ trong các điều kiện ánh sáng khác nhau. Khi cường độ sáng thay đổi, thông số ngưỡng trong quá trình chuyển ảnh các quân cờ

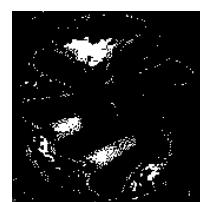
thành ảnh nhị phân được điều chỉnh cho phù hợp để ghi lại đầy đủ các nét trong biểu tượng quân cờ.

Ví dụ: hình 3.26a được chụp từ camera, sau khi biến đổi thành ảnh xám và áp dụng các ngưỡng threshold khác nhau thu được các ảnh nhị phân 3.26b, 3.26c, 3.26d và 3.26e. Trong trường hợp này, hình 3.26c và 3.26d là những hình ảnh phù hợp cho quá trình nhận diện.

Ba trường hợp môi trường ánh sáng khác nhau được chọn thử nghiệm cùng thông số ngưỡng phù hợp và kết quả quá trình nhận diện quân cờ trong những trường hợp đó thể hiện trong bảng 3.7.



a. Ảnh chụp từ camera.



b. threshold = 130. c. threshold = 180. d. threshold = 230. e. threshold = 250.

Hình 3.26: Hình ảnh quân cờ qua biến đổi ở các ngưỡng khác nhau.

Bảng 3.7: Kết quả nhận diện quân cờ trong các điều kiện ánh sáng khác nhau.

| Môi trường | Trong nhà | | Ngoài trời, trong bóng râm | | Ngoài trời, dưới ánh nắng trực tiếp | |
|------------|------------------------|---------------|-------------------------------|---------------|--|---------------------|
| | Phòng 202 – C1 | Trước tòa B11 | | | Trước xưởng C1 | |
| Ngưỡng | 130 | | 160 | | 200 | |
| Trường hợp | 8 quân cờ bất kì | 32 quân cờ | 14 quân cờ khác nhau | 32 quân cờ | 14 quân cờ khác nhau | 9 quân cờ bất kì |

| | | | | | | |
|-------------------------|---------|---------|---------|---------|---------|---------|
| Nhận diện đúng | 100% | 100% | 100% | 100% | 100% | 100% |
| Thời gian trung bình | 40,8 ms | 40,6 ms | 41,4 ms | 41,2 ms | 40,9 ms | 40,3 ms |

Quá trình nhận diện quân cờ trong các trường hợp được kiểm tra đều cho kết quả chính xác 100%, tốc độ nhận diện nhanh, trung bình khoảng 41 ms với mỗi quân cờ. Việc điều chỉnh thông số để tương thích với các môi trường khác nhau đơn giản. Đây là những cải thiện đáng kể so với quá trình nhận diện trong các đề tài đã thực hiện trước đó tại trường Đại học Bách khoa TPHCM.

3.5. Kết luận

Các nhiệm vụ đề ra ở đầu chương đã được giải quyết.

- Quá trình xác định vị trí quân cờ trả về đầy đủ tọa độ các quân với sai lệch không đáng kể.
- Quá trình nhận diện quân cờ đã có tốc độ xử lý nhanh và cho kết quả hoàn toàn chính xác trong nhiều trường hợp được kiểm tra.

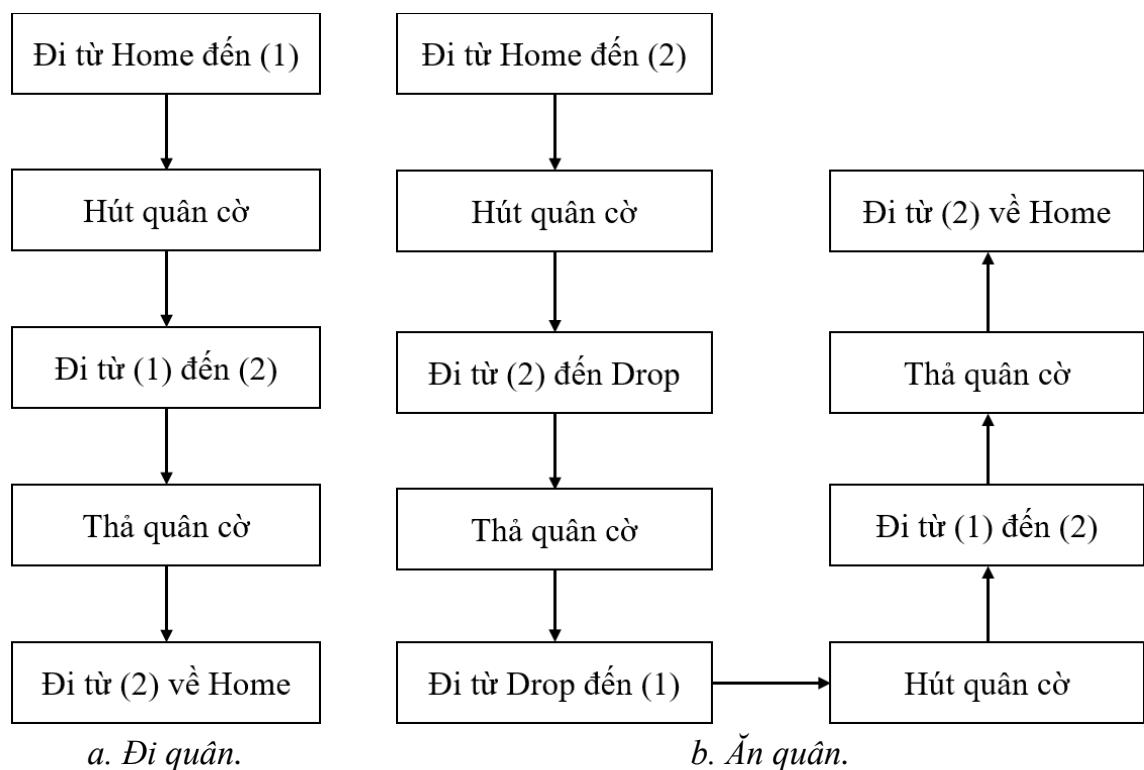
Việc xác định vị trí và nhận diện quân cờ ổn định và hiệu quả sẽ hỗ trợ giải quyết các vấn đề trong chương tiếp theo là lập trình tự động sắp xếp quân cờ và tự động chơi cờ cho robot.

CHƯƠNG 4. ĐIỀU KHIỂN ĐẦU CÔNG TÁC

Đầu công tác của robot chơi cờ tướng trong đề tài này sẽ được điều khiển theo cấu trúc điều khiển phân cấp. Trong đó, máy tính (master) sẽ dựa theo tín hiệu xử lý ảnh và thuật toán đánh cờ để điều khiển PLC (slave) thực hiện các tác vụ đối với đầu công tác. Vì vậy, trong chương này giải thuật điều khiển đầu công tác và giao tiếp PLC – máy tính sẽ lần lượt được giới thiệu.

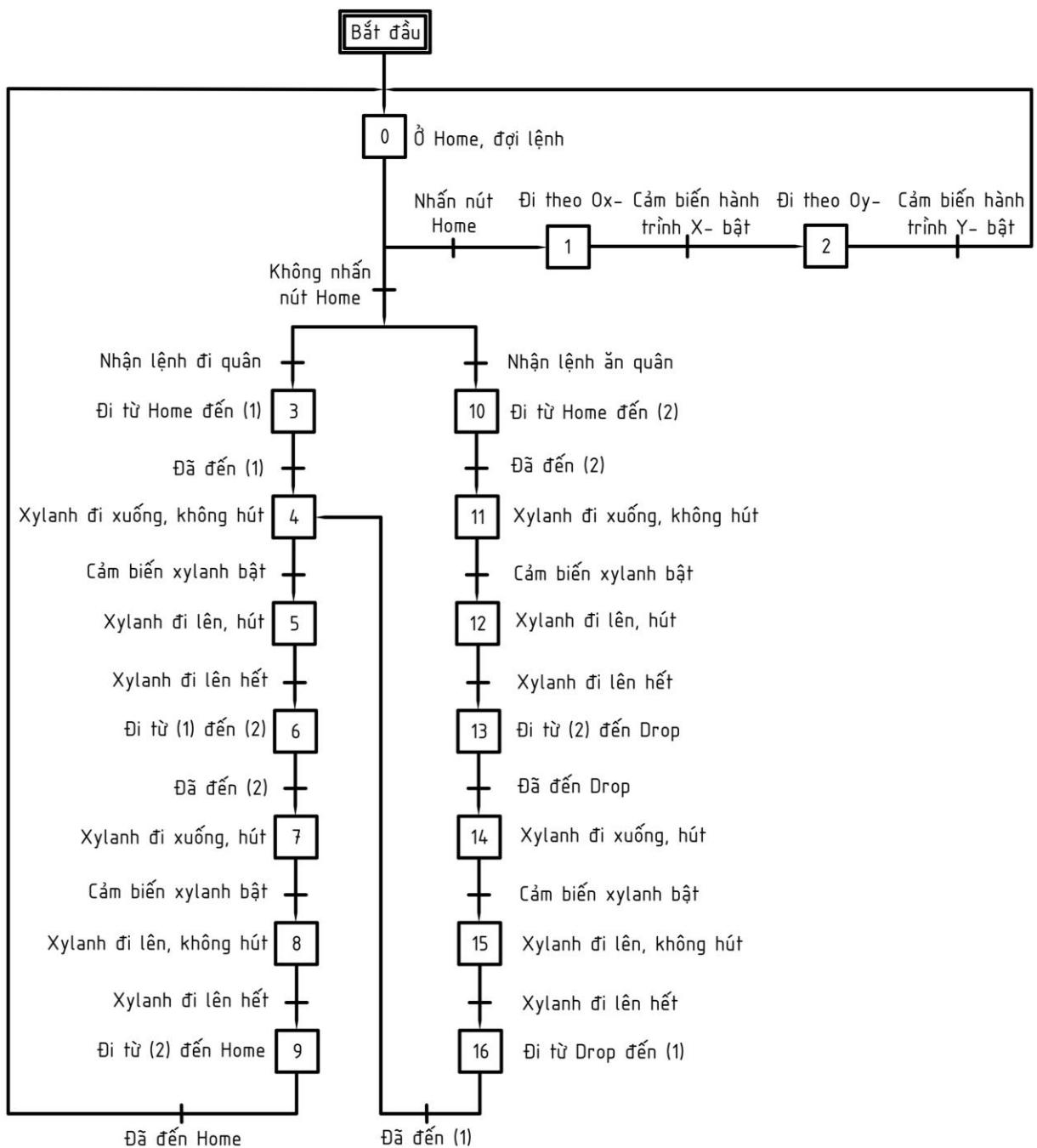
4.1. Giải thuật điều khiển đầu công tác

Nhằm phân tích các trạng thái của đầu công tác trong quá trình vận hành của robot, điều cần xác định trước tiên là các hành động của robot. Cụ thể, đó là hành động đi quân và ăn quân đối thủ. Hai hành động này có thể được phân tích thành một loạt các bước nhỏ mà đầu công tác thực hiện như hình 4.1. Trong đó, Home là vị trí ban đầu của đầu công tác, (1) là vị trí hiện tại của quân cờ cần di chuyển hoặc ăn quân, (2) là vị trí mà quân cờ sẽ đến hoặc vị trí quân đối phương sẽ bị ăn và Drop là điểm thả quân cờ đã ăn. Ngoài ra, robot còn hành động về Home trong những trường hợp thiết lập trạng thái ban đầu cho đầu công tác hay trường hợp dừng khẩn cấp.



Hình 4.1: Các bước để đầu công tác thực hiện hành động đi quân và ăn quân.

Lưu đồ giải thuật điều khiển đầu công tác như hình 4.2 được trình bày nhằm làm rõ hơn các trạng thái của đầu công tác trong từng quá trình cũng như điều kiện chuyển giữa các trạng thái. Lưu đồ thể hiện rằng ở trạng thái đầu tiên sau khi bắt đầu (trạng thái 0), đầu công tác ở trạng thái đứng yên ở Home và chờ lệnh từ PLC. Nếu nút Home được nhấn, robot sẽ tiến hành quá trình về Home. Ngược lại, robot sẽ tiến hành ăn quân hoặc đi quân tùy theo lệnh được gọi trước khi trở lại trạng thái 0.



Hình 4.2: Lưu đồ giải thuật điều khiển đầu công tác.

4.1.1. Các trạng thái của quá trình về Home

Quá trình về Home bắt đầu bằng việc đầu công tác chạy dọc theo phương x bàn máy theo chiều âm đến khi gặp cảm biến hành trình X-. Tại thời điểm này, đầu công tác lại chuyển sang trạng thái chạy dọc theo phương y bàn máy theo chiều âm. Đầu công tác chỉ dừng lại khi gặp cảm biến hành trình Y-. Khi đó, chuyển động theo chiều âm của cả hai phương x và y đều bị khóa, đầu công tác hoàn toàn đứng yên ở Home và quá trình về Home kết thúc.

4.1.2. Các trạng thái của quá trình đi quân

Di chuyển điểm đến điểm là một bước quan trọng trong quá trình đi quân và cả ăn quân. Xung được cấp cho động cơ để làm đầu công tác di chuyển từ điểm đến điểm được tính toán dựa vào phương trình động học nghịch.

Gọi p_1, p_2 là xung cần cấp cho động cơ M_1, M_2 . Biết rằng, bán kính pulley chủ động là $R = 14mm$, một vòng quay của động cơ cần 4500 xung. Như vậy, mối quan hệ giữa xung cần cấp cho mỗi động cơ và góc quay động cơ là:

$$\begin{cases} \alpha = \frac{2\pi}{4500} p_1 \\ \beta = \frac{2\pi}{4500} p_2 \end{cases} \quad (4.1)$$

Thay (4.1) vào phương trình động học ngược, mối quan hệ giữa xung cần cấp cho động cơ và độ dời theo hai phương x và y được thể hiện như sau:

$$\begin{cases} -\Delta_x + \Delta_y = \alpha R = \frac{2\pi R}{4500} p_1 \text{ (mm)} \\ -\Delta_x - \Delta_y = \beta R = \frac{2\pi R}{4500} p_2 \text{ (mm)} \end{cases} \quad (4.2)$$

Quá trình đi quân bao gồm một chuỗi trạng thái bắt đầu từ việc đầu công tác đi từ Home đến vị trí (1). Sau đó, xylanh sẽ đi xuống đến khi gặp cảm biến từ xylanh thì bắt đầu hút và chuyển hướng đi lên. Sau khi xylanh đi lên, quân cờ lúc này đã được hút và giữ, đầu công tác di chuyển từ vị trí (1) đến vị trí (2). Khi đầu công tác đến vị trí (2), xylanh đi xuống và thả quân cờ ngay khi cảm biến từ xylanh được kích hoạt. Ngay sau đó, xylanh lại đi lên trong trạng thái không hút. Cuối cùng, khi xylanh đi lên hết hành trình của nó,

đầu công tác đi từ vị trí (2) về Home và quá trình kết thúc khi đầu công tác đã ở Home và chờ lệnh mới.

4.1.3. Các trạng thái của quá trình ăn quân

Trong quá trình này, việc gấp thả các quân cờ sẽ được tiến hành tương tự như quá trình đi quân. Đầu tiên, đầu công tác sẽ đi từ Home đệm (2) và tiến hành gấp quân cờ. Sau đó, đầu công tác đến vị trí Drop để thả quân cờ. Khi quân cờ bị ăn được thả xong thì đầu công tác lại đi từ vị trí Drop đến vị trí (1). Lúc này, các bước đầu công tác thực hiện lại giống với quá trình đi quân từ vị trí (1) đến (2).

4.2. Giao tiếp giữa PLC và máy tính

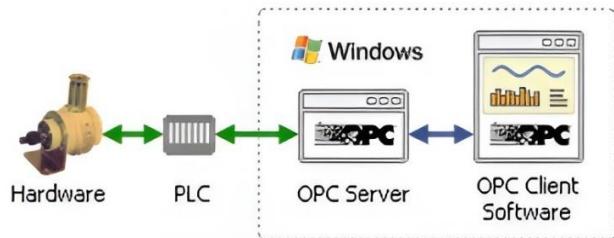
4.2.1. Chuẩn giao tiếp OPC

OPC là một chuẩn giao tiếp dữ liệu giữa các phần mềm, theo cơ chế client-server , được sử dụng rộng rãi trong công nghiệp để đảm bảo tính linh hoạt và tương thích giữa các thành phần.

Từ viết tắt OPC được hiểu là “OLE (Object Linking and Embedding) for Process Control”. Vì OLE dựa trên tiêu chuẩn là Windows COM (Component Object Model), nên OPC về cơ bản là COM. OPC thực sự bao gồm một số tiêu chuẩn, đầu tiên và quan trọng nhất trong số đó là OPC Data Access (OPC DA), sau đó phổ biến hiện tại là OPC Unified Architecture (OPC UA). Giá trị của OPC là nó là một tiêu chuẩn mở, có nghĩa là chi phí sẽ thấp hơn cho các nhà sản xuất và nhiều tùy chọn hơn cho người dùng. Các nhà sản xuất phần cứng chỉ cần phải cung cấp một máy chủ OPC (OPC Server) duy nhất cho các thiết bị của họ để giao tiếp với bất kỳ một máy khách OPC (OPC Client) nào.

Như vậy, việc cần làm để thiết lập giao thức OPC là tạo ra một OPC Server.

- OPC Server là một chương trình phần mềm chuyển đổi giữa giao thức truyền thông phần cứng được PLC sử dụng thành ra giao thức OPC.
- OPC Client là bất kỳ chương trình nào mà cần kết nối với phần cứng, chẳng hạn như là HMI hay các phần mềm SCADA. OPC Client sử dụng OPC Server để lấy được dữ liệu từ hay gửi các lệnh tới phần cứng như được minh họa trong hình 4.3.

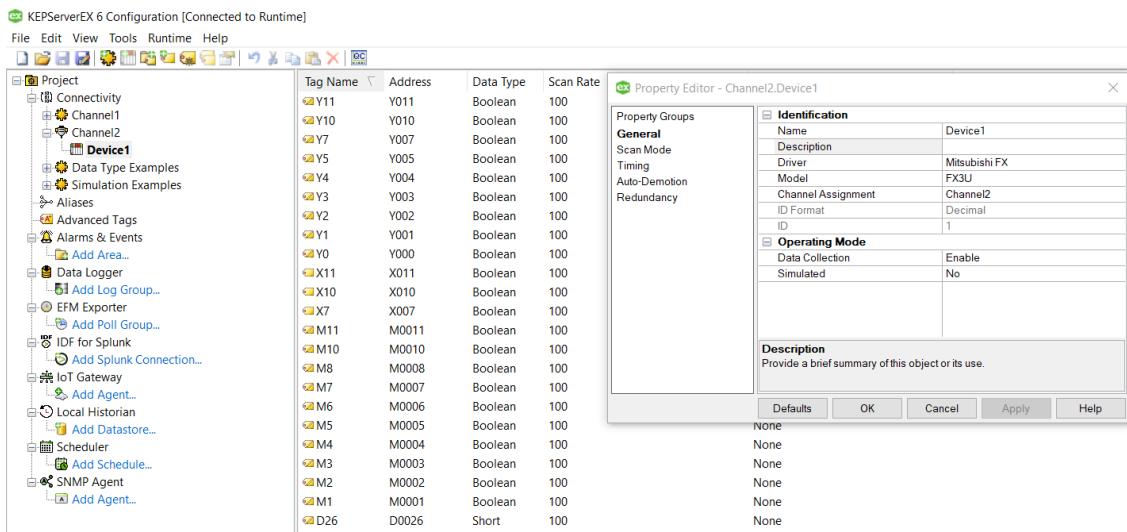


Hình 4.3: Mô hình OPC kết nối máy tính với các thiết bị công nghiệp. [18]

4.2.2. Phần mềm KEPServerEX

KEPServerEX là phần mềm tạo OPC Server, cho phép người dùng kết nối, theo dõi và kiểm soát các thiết bị tự động hóa và ứng dụng phần mềm trong doanh nghiệp.

Các OPC Server tạo ra bởi phần mềm này có thể hỗ trợ các thiết bị của các hãng: OMRON, KEYENCE, KOYO, Toshiba, JTEKT, Panasonic, FANUC, Fuji Electric, Mitsubishi Electric, Yaskawa Electric, Yokogawa Electric, Beckhoff, Rockwell, Siemens,... Trong đó, PLC FX3U được sử dụng trong đề tài cũng được hỗ trợ.



Hình 4.4: Giao diện KEPServerEX khi tạo OPC Server cho PLC Mitsubishi FX3U.

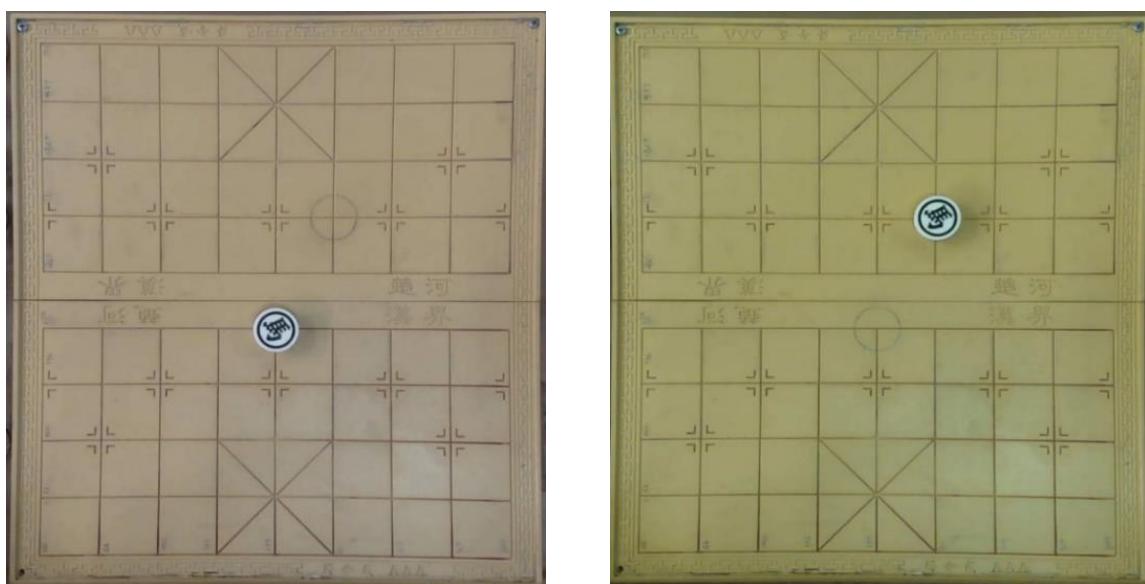
Sau khi kết nối cổng RS23 của PLC với cổng COM của máy tính, các bước tiếp theo sẽ là tạo một OPC Server cho thiết bị và khai báo các cờ, timer, thanh ghi được sử dụng trong chương trình PLC như hình 4.4.

Sau khi OPC Server được tạo, phần mềm chơi cờ sẽ đóng vai trò như một OPC Client truy xuất đến OPC Server để đọc/ghi các cờ, timer hay thanh ghi của PLC, từ đó gán tiếp điều khiển đầu công tác từ máy tính.

4.3. Kết quả

Kết quả đạt được trong chương này là việc điều khiển thành công đầu công tác đi quân hoặc ăn quân từ tín hiệu máy tính. Cụ thể, hai thử nghiệm tiếp sau đây sẽ minh họa cho tính chính xác của giải thuật điều khiển đầu công tác.

Trong thử nghiệm 1, quân cờ được đặt vào một đường tròn tâm có tọa độ điểm xuất phát và bán kính bằng bán kính quân cờ như hình 4.5a. Mục tiêu của thử nghiệm là đánh giá sai lệch của quân cờ được di chuyển bởi robot tới điểm đến như hình 4.5b.



a. Trước khi đi quân.

b. Sau khi đi quân.

Hình 4.5: Minh họa cho việc tiến hành thực nghiệm 1.

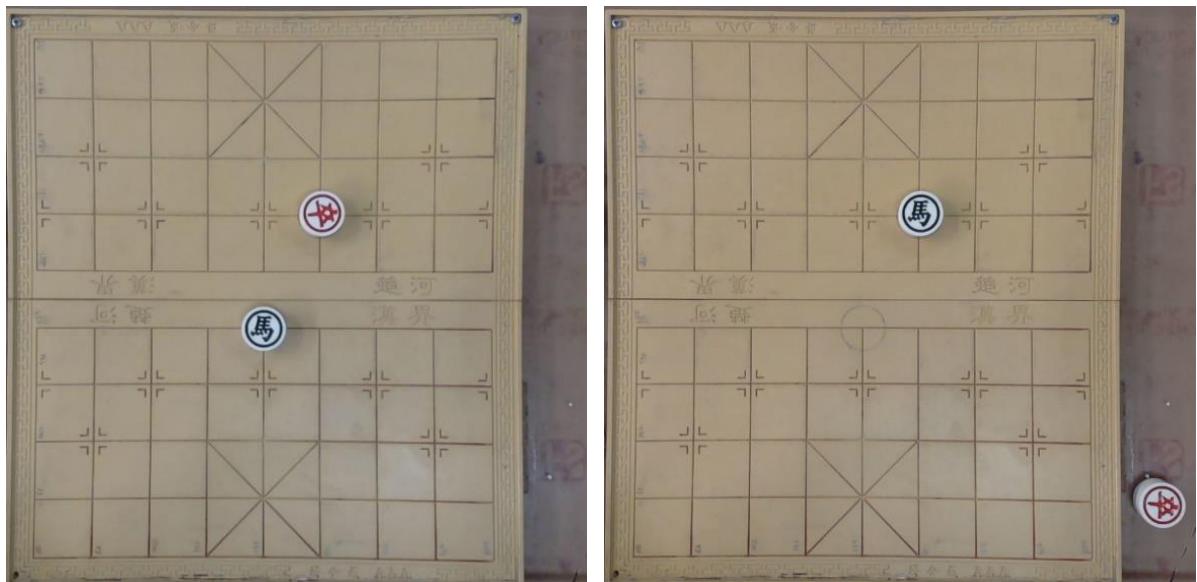
Thực nghiệm 1 được tiến hành 10 lần với các vị trí xuất phát và vị trí đến khác nhau. Kết quả về sai lệch theo phương x và y được đo bằng thước có độ chia nhỏ nhất 1mm và được tổng hợp trong bảng 4.1.

Bảng 4.1: Kết quả thực nghiệm 1.

| Lần | $ \Delta_x $ (mm) | $ \Delta_y $ (mm) |
|-----|-------------------|-------------------|
| 1 | 0 | 1 |
| 2 | 1 | 1 |
| 3 | 0 | 1 |
| 4 | 1 | 2 |
| 5 | 1 | 2 |
| 6 | 1 | 1 |
| 7 | 0 | 1 |

| | | |
|----|----------------------|----------------------|
| 8 | 0 | 0 |
| 9 | 0 | 2 |
| 10 | 1 | 0 |
| | $ \Delta_x = 0,5mm$ | $ \Delta_y = 1,1mm$ |

Thực nghiệm 2 được thực hiện gần như tương tự thực nghiệm 1 nhưng bao gồm thêm hành động gấp quân cờ bị ăn và đưa ra vị trí Drop bên ngoài bàn cờ như hình 4.6. Kết quả thực nghiệm 2 sau 10 lần tiến hành với các vị trí khác nhau được đo bằng thước có độ chia nhỏ nhất 1mm và được thể hiện trong bảng 4.2.



Hình 4.6: Minh họa cho việc tiến hành thực nghiệm 2.

Bảng 4.2: Kết quả thực nghiệm 2.

| Lần | $ \Delta_x (mm)$ | $ \Delta_y (mm)$ |
|-----|-------------------|-------------------|
| 1 | 1 | 1 |
| 2 | 1 | 1 |
| 3 | 2 | 2 |
| 4 | 2 | 1 |
| 5 | 1 | 1 |
| 6 | 2 | 1 |
| 7 | 0 | 1 |
| 8 | 1 | 1 |
| 9 | 0 | 2 |
| 10 | 1 | 1 |

| | | |
|--|---------------------------------|---------------------------------|
| | $ \overline{\Delta_x} = 1,1mm$ | $ \overline{\Delta_y} = 1,2mm$ |
|--|---------------------------------|---------------------------------|

4.4. Kết luận:

Như vậy, chương 4 đã hoàn thành quy trình điều khiển đầu công tác, máy tính có thể di chuyển và hút thả quân cờ theo mong muốn với sai số của đầu công tác và vị trí đích không quá 1.5 mm. Kết hợp với sai số xác định vị trí quân cờ ở chương 3 là 2.5 mm, tâm của đầu công tác có thể bị lệch so với tâm quân cờ từ 1 – 4 mm, khá nhỏ so với đường kính quân cờ là 31 mm. Đây là một kết quả quan trọng, đảm bảo cơ sở vững chắc để các hoạt động như xếp cờ hay chơi cờ (chương 5) hoạt động ổn định.

CHƯƠNG 5. PHÂN TÍCH VÀ XÂY DỰNG GIẢI THUẬT CHƠI CỜ CHO ROBOT

Chương này tập trung tìm hiểu, phân tích và xây dựng các giải thuật cho phần mềm điều khiển để đưa nước cờ mỗi khi đến lượt của robot. Ngoài ra, các tính năng như xếp cờ, chơi từ thế cờ có sẵn hay lựa chọn các mức độ chơi dễ đến khó của phần mềm chơi cờ cũng được phát triển.

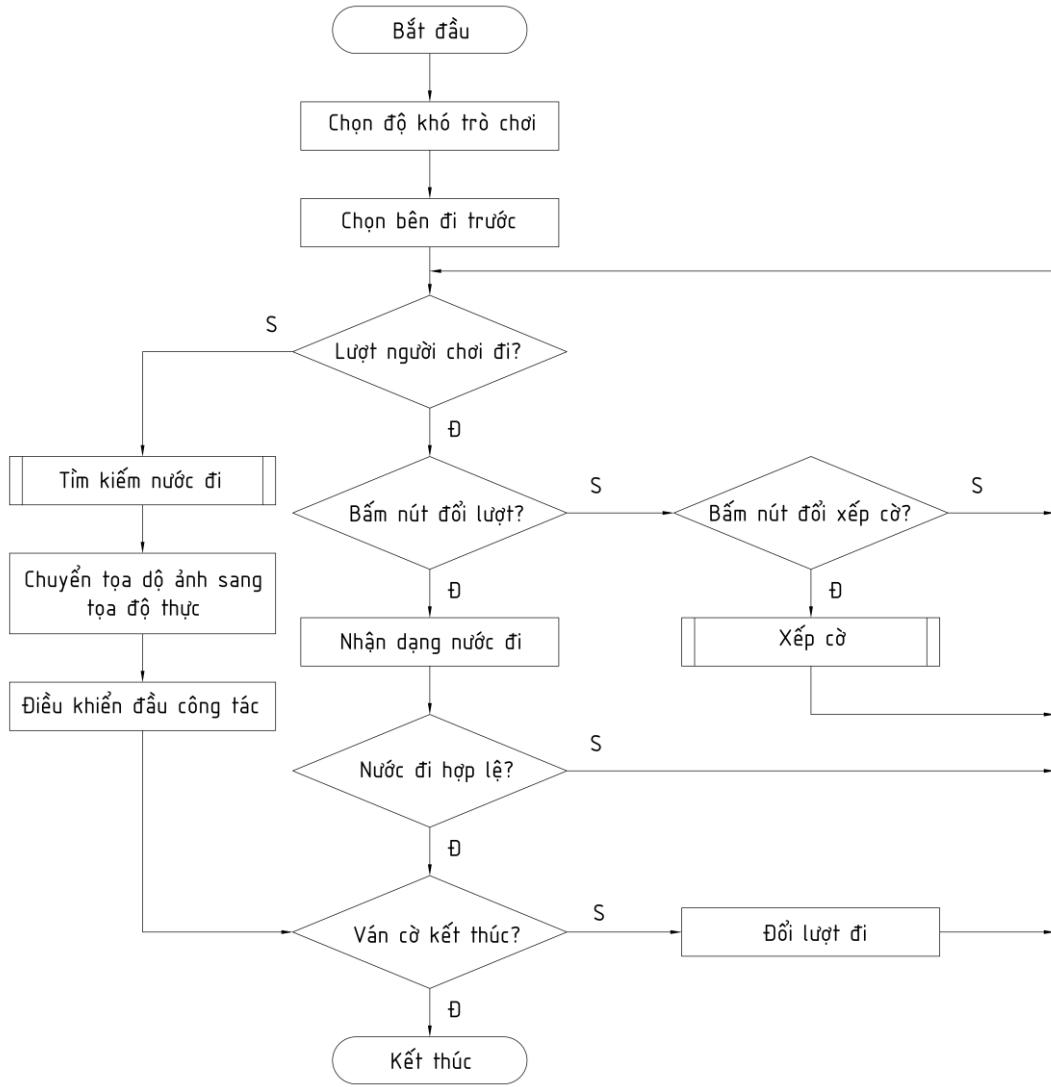
Lưu đồ hình 5.1 thể hiện quá trình tự động chơi cờ của robot.

- Khi người chơi thực hiện xong lượt của mình và bấm nút đổi lượt, đèn báo đổi lượt tắt, camera ghi lại hình ảnh bàn cờ. Phần mềm điều khiển sẽ nhận diện các quân cờ trên bàn cờ cùng vị trí của chúng để kiểm tra tính hợp lệ của trạng thái bàn cờ. Nếu trạng thái là hợp lệ, phần mềm sẽ phát hiện nước đi của người chơi, và tiếp tục kiểm tra tính hợp lệ của nước đi đó. Sau khi đã xác nhận nước đi không có lỗi, phần mềm sẽ tính toán nước đi phản hồi, rồi điều khiển đầu công tác thực hiện nước đi đó. Quá trình chơi cờ kết thúc khi người chơi hay máy bị chiêu hết, hoặc người chơi bỏ cuộc.
- Khi người chơi bấm nút xếp cờ, phần mềm điều khiển sẽ ghi nhận lại hình ảnh bàn cờ, rồi xác định tọa độ đồng thời nhận diện các quân cờ. Cuối cùng, đầu công tác lần lượt di chuyển đến các quân cờ và xếp chúng vào đúng vị trí. Chức năng này cũng phù hợp với chế độ chơi cờ thê, chỉ cần cài đặt trước thế cờ, robot sẽ tự động sắp xếp cho người chơi.

Việc tự động sắp xếp quân cờ là một chức năng cần thiết của robot, giúp người chơi có một khoảng nghỉ thoải mái trước khi bắt đầu một ván cờ mới.

Như vậy, có 3 mục tiêu chính cần giải quyết ở chương này:

- Xây dựng giải thuật sắp xếp quân cờ.
- Xây dựng giải thuật phát hiện nước đi của người chơi và kiểm tra tính đúng đắn của nước đi đó.
- Xây dựng giải thuật chơi cờ để đưa ra nước đi tại mỗi lượt chơi của robot.



Hình 5.1: Lưu đồ hoạt động chơi cờ của robot.

5.1. Sắp xếp quân cờ

5.1.1. Giải thuật sắp xếp quân cờ

Từ đây, quy ước rằng quân xanh là của người chơi, quân cờ đỏ là của robot. Trên hình ảnh bàn cờ sau quá trình xếp cờ, quân xanh sẽ nằm phía trên, quân đỏ nằm phía dưới.

Ban đầu các quân cờ sẽ được đặt một cách ngẫu nhiên trên bàn cờ. Vấn đề có thể nảy sinh với hoạt động xếp cờ là khi các quân cờ nằm che lấp vị trí của quân cờ khác. Bởi vậy, cần có một thứ tự ưu tiên cho các quân cờ để luôn tồn tại khoảng trống tại vị trí cần xếp mỗi lần xét đến một quân cờ nào đó. Ví dụ: trên hình 5.2, quân pháo xanh đã nằm che mất vị trí xếp ban đầu của tượng đỏ, mặt khác quân pháo đỏ lại che mất vị trí của quân pháo đen.

Nhằm giảm tải việc tính toán thứ tự sắp xếp từng quân cờ, trong phạm vi đế tài này, quy định rằng các quân cờ trước khi xếp không được nằm trong 2 vùng không gian giới hạn bởi đường ngang chia ô thứ nhất ở mỗi phía và mép bàn cờ như trên hình 5.3.



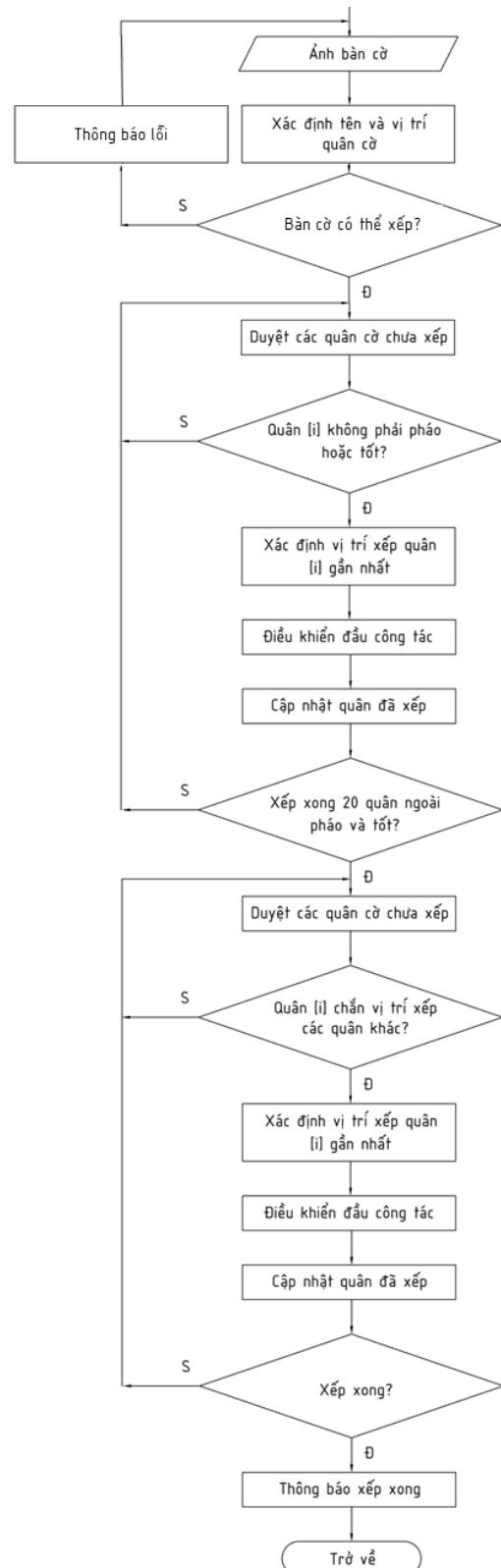
Hình 5.2: *Quân cờ nằm che lấp vị trí cần xếp vào của quân cờ khác.*



Hình 5.3: *Giới hạn không gian đặt quân cờ trước khi xếp.*

Quá trình xếp cờ được thể hiện trong lưu đồ hình 5.4. Theo đó, sau khi bấm nút xếp cờ, đèn báo đổi lượt tắt, camera chụp lại hình ảnh bàn cờ. Phần mềm điều khiển sẽ xác định vị trí và tên các quân cờ, từ đó kiểm tra các quân cờ có đảm bảo các điều kiện cho trước: đầy đủ 32 quân cờ đặt trên bàn, các quân cờ được lật ngửa phần hình lên. Từ đó, phần mềm lập danh sách thứ tự xếp cờ. Các quân Xe, Mã, tượng, Sĩ, Tướng sẽ được ưu tiên trong danh sách vì vị trí cần xếp vào của chúng đang trống. Trừ quân Tướng chỉ có 1 vị trí để xếp vào, mỗi quân cờ sẽ có nhiều nhất 2 vị trí để xếp. Khi ấy, vị trí được chọn sẽ

là vị trí còn trống gần quân cờ cần xếp nhất. Điều này sẽ tiết kiệm thời gian di chuyển cho đầu công tác, từ đó tối ưu quá trình xếp cờ.



Hình 5.4: Lưu đồ quá trình xếp cờ.

Đối với các quân cờ còn lại, quân nào đang nằm che vị trí xếp của các quân khác sẽ được ưu tiên xếp trước. Quân cờ được xác định là che vị trí xếp của quân khác khi khoảng cách giữa tâm quân cờ và vị trí xếp nhỏ hơn một khoảng nhất định. Với đường kính quân cờ là 31 mm, khoảng cách lớn nhất giữa tâm quân cờ và vị trí xếp được chọn là 25 mm.

Mỗi vị trí xếp sẽ được ghi nhận số lượng quân cờ đang che nó. Khi một quân cờ được đưa vào danh sách xếp, số lượng quân cờ đang che những vị trí xếp xung quanh nó được giảm đi 1 đơn vị. Khi tất cả các quân cờ đang che một vị trí xếp đều đã được đưa vào danh sách di chuyển thì vị trí đó trở thành vị trí trống và có thể xếp cờ vào.

Sau khi có danh sách thứ tự xếp cờ, đầu công tác di chuyển từ vị trí Home đến quân cờ đầu tiên, hút nó lên, rồi đặt vào vị trí cần xếp vào tương ứng. Đầu công tác tiếp tục di chuyển tới quân cờ tiếp theo và lặp lại quá trình xếp cờ cho đến khi các quân cờ đều ở đúng vị trí. Cuối cùng đầu công tác di chuyển về vị trí Home, đèn báo đổi lượt bật sáng và người chơi có thể bắt đầu chơi cờ.

5.1.2. Đánh giá chức năng sắp xếp quân cờ

Phần này đánh giá chức năng sắp xếp quân cờ thông qua thời gian xếp cờ và tỉ lệ số quân được xếp đúng trong 20 trường hợp bàn cờ bất kì. Thời gian xếp cờ tính từ lúc bấm nút xếp cờ đến khi đầu công tác di chuyển trở lại vị trí Home của nó.

Bảng 5.1: Tỉ lệ xếp đúng và thời gian xếp trong các trường hợp bàn cờ khác nhau.

| Trường hợp | Tỉ lệ xếp đúng | Thời gian xếp |
|------------|----------------|---------------|
| 1 | 100% | 147 s |
| 2 | 100% | 136 s |
| 3 | 100% | 149 s |
| 4 | 100% | 145 s |
| 5 | 100% | 142 s |
| 6 | 100% | 147 s |
| 7 | 100% | 139 s |
| 8 | 100% | 146 s |

| | | |
|------------|------|---------|
| 9 | 100% | 140 s |
| 10 | 100% | 135 s |
| 11 | 100% | 138 s |
| 12 | 100% | 138 s |
| 13 | 100% | 137 s |
| 14 | 100% | 146 s |
| 15 | 100% | 148 s |
| 16 | 100% | 145 s |
| 17 | 100% | 149 s |
| 18 | 100% | 149 s |
| 19 | 100% | 149 s |
| 20 | 100% | 144 s |
| Trung bình | 100% | 143,4 s |

Chức năng xếp cờ trong các trường hợp được thử nghiệm hoạt động hoàn toàn chính xác. Thời gian xếp cờ ổn định với chênh lệch không đáng kể, trung bình 143,4 s.

5.2. Xác định nước đi của người chơi

Phần này xác định nước đi của người chơi và kiểm tra tính đúng đắn của nước đi đó.

- Nếu nước đi hợp lệ, robot sẽ tính toán nước đi phản hồi.
- Nếu nước đi không hợp lệ, robot sẽ thông báo để người chơi biết và đợi người chơi thực hiện lại nước đi rồi bấm nút đổi lượt một lần nữa.

Quá trình xác định nước đi của người chơi trải qua các bước như sau:

- Từ hình ảnh bàn cờ đã lưu, phần mềm xác định tên và vị trí các quân cờ trong hệ tọa độ bàn cờ. Nếu các quân cờ đều nằm trên các điểm giao của các đường thẳng chia ô bàn cờ thì một trạng thái bàn cờ sẽ được mô phỏng lại. Trạng thái này là hợp lệ và được lưu trữ để tính toán tiếp nếu như tất cả quân cờ đều nằm trong những khu vực hoặc vị trí được quy định.
- Bằng việc so sánh trạng thái bàn cờ sau mỗi lượt chơi, phần mềm sẽ phát hiện ra nước đi của người chơi. Nếu chỉ có 1 nước đi được xác định và nước đi này đúng luật, nước đi sẽ được lưu lại để phần mềm tính toán nước đi phản hồi cho robot.

Các mục tiếp theo sẽ trình bày cụ thể phương án phát hiện, kiểm tra và lưu trữ trạng thái bàn cờ, các điều kiện và phương thức kiểm nghiệm tính đúng đắn của nước đi.

5.2.1. Mô tả trạng thái bàn cờ

Từ ảnh chụp, trạng thái bàn cờ được mô tả dưới 2 dạng sau: ma trận và chuỗi FEN.

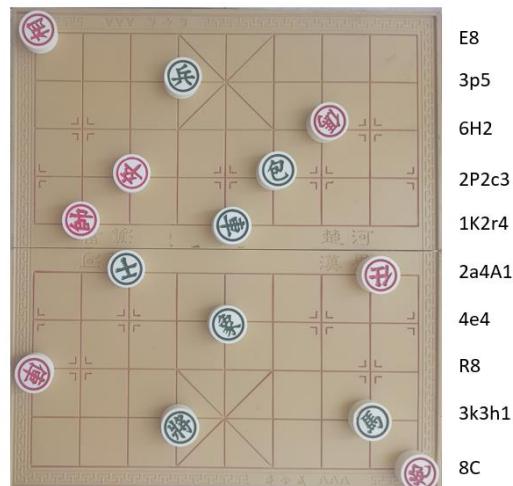
a. Chuỗi FEN

FEN (Forsyth-Edwards Notation) là một tiêu chuẩn để mô tả vị trí quân cờ trong cờ Tướng sử dụng bộ kí tự ASCII. Ưu điểm của nó là đơn giản, ngắn gọn, dễ diễn đạt và dễ lập trình cho máy.

Khi Mã hóa bàn cờ thành chuỗi FEN, các quân cờ sẽ được viết bằng các chữ cái, quân đỏ (hay trắng) được viết bằng chữ cái in hoa, quân xanh (hay đen) được viết bằng chữ cái thường như trên bảng 5.2. Các điểm trống liên tiếp nhau trên bàn cờ được viết bằng số lượng của chúng (1 - 9). Ký tự gạch chéo (/) dùng để ngăn cách giữa các hàng. Ở cuối chuỗi là “w” nếu đến lượt quân đỏ đi, hoặc “b” nếu đến lượt quân xanh đi. Hình 5.5 mô tả quá trình mã hóa bàn cờ thành chuỗi FEN.

Bảng 5.2: Quy ước Mã hóa các quân cờ.

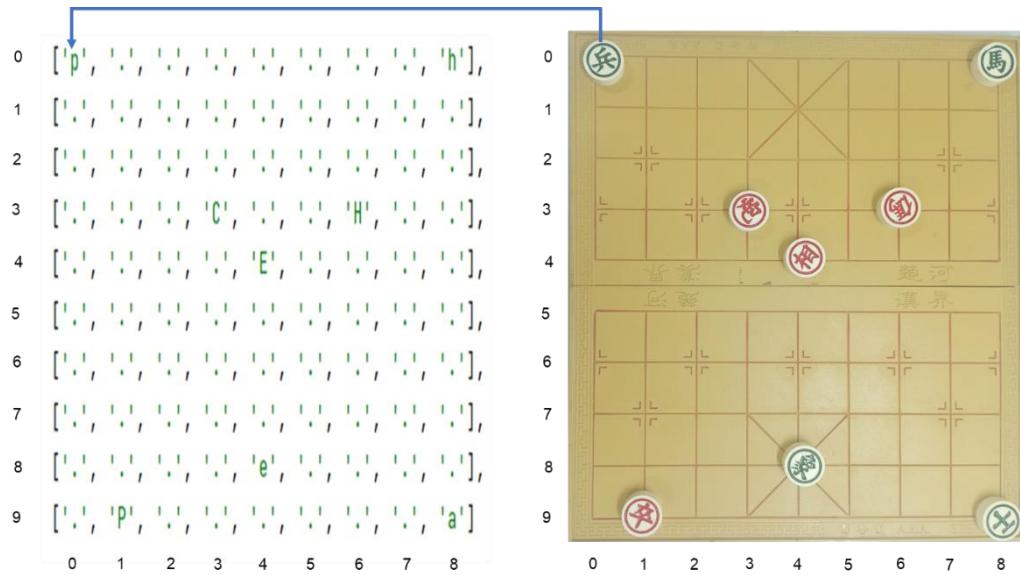
| | | | | | | | |
|---------|------------|---------|------------|---------|---------|-----------|----------|
| Quân | Tướng đỏ | Sĩ đỏ | Tượng đỏ | Mã đỏ | Xe đỏ | Pháo đỏ | Tốt đỏ |
| Kí hiệu | K | A | E | H | R | C | P |
| Quân | Tướng xanh | Sĩ xanh | Tượng xanh | Mã xanh | Xe xanh | Pháo xanh | Tốt xanh |
| Kí hiệu | k | a | e | h | r | c | p |



Hình 5.5: Mã hóa bàn cờ thành chuỗi FEN.

b. Ma trận

Một ma trận cờ 10×9 được dùng để mô tả trạng thái bàn cờ như trên hình 5.6. Phần tử $[i, j]$ của ma trận sẽ có giá trị ‘k’, ‘h’...(giống cách kí hiệu của chuỗi FEN) tùy theo quân cờ ở vị trí $[i, j]$ trên ảnh bàn cờ, ngược lại vị trí trống sẽ có giá trị trên ma trận là ‘.’.



Hình 5.6: Mô tả trạng thái bàn cờ dưới dạng ma trận.

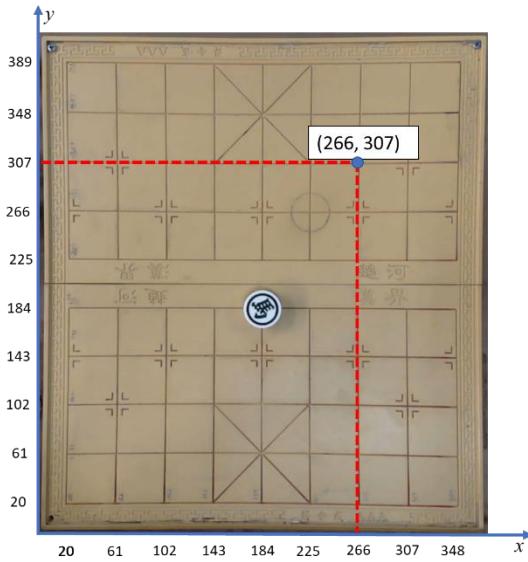
Việc lưu trữ trạng thái bàn cờ dưới dạng ma trận sẽ giúp quá trình phát hiện trạng thái bàn cờ sai hay người chơi đi sai dễ dàng hơn. Trạng thái bàn cờ dưới dạng ma trận có thể chuyển đổi đơn giản sang dạng FEN hoặc ngược lại thuận tiện cho việc lưu trữ và truy xuất các ván chơi, nước đi.

5.2.2. Kiểm tra tính hợp lệ của bàn cờ

Quá trình kiểm tra tính hợp lệ của bàn cờ gồm 2 bước chính:

- Bước 1: Kiểm tra các quân cờ có nằm trên các giao điểm của các đường chia ô bàn cờ.

Trong hệ tọa độ bàn cờ, tọa độ các điểm giao được xác định như trên hình 5.7. Khi chơi trong thực tế, quân cờ không cần phải nằm chính xác trên các điểm giao, mà có thể nằm lệch một chút. Một khác quá trình xác định tọa độ quân cờ từ ảnh chụp cũng có sai số nhỏ. Trong phạm vi đề tài này, quy định độ lệch lớn nhất giữa tọa độ quân cờ và các điểm giao là 10 mm.



Hình 5.7: Hệ tọa độ bàn cờ.

Khi duyệt lần lượt toàn bộ các quân cờ đã được phát hiện, nếu độ lệch giữa một quân cờ và một điểm giao gần nó không vượt quá độ lệch giới hạn thì quân cờ đó được tính là nằm trên đường giao. Nếu có ít nhất một quân cờ không nằm trên đường giao thì trạng thái bàn cờ không hợp lệ.

- Bước 2: Kiểm tra quân cờ có nằm trong những khu vực hoặc vị trí được quy định. Các điều kiện kiểm tra bao gồm:

- Các quân Tướng, Sĩ, Tượng phải nằm đúng phía ở mỗi bên của bàn cờ.

Theo ma trận bàn cờ, quân xanh nằm ở các hàng có thứ tự nhỏ hơn 5, quân đỏ nằm ở các hàng có thứ tự lớn hơn hoặc bằng 5.

- Quân Tướng:

+ Phải nằm trong khu vực di chuyển được của nó theo luật cờ Tướng, đó là các vị trí trong hình vuông $(0, 3) - (2, 5)$ đối với quân xanh, và $(7, 3) - (9, 5)$ đối với quân đỏ (hình 5.8).

+ 2 quân Tướng không được “lộ mặt” với nhau (2 quân Tướng nằm trên cùng 1 đường thẳng mà không có quân nào chắn ở giữa).

- Quân Sĩ: Phải nằm ở các vị trí $(0, 3), (0, 5), (1, 4), (2, 3)$ hoặc $(2, 5)$ đối với Sĩ xanh; $(9, 3), (9, 5), (8, 4), (7, 3)$ hoặc $(7, 5)$ đối với Sĩ đỏ.

- Quân Tượng: Phải nằm ở các vị trí $(0, 2), (0, 6), (2, 0), (2, 4), (2, 8), (4, 2)$ hoặc $(4, 6)$ đối với Sĩ xanh; $(9, 2), (9, 6), (7, 0), (7, 4), (7, 8), (5, 2)$ hoặc $(5, 6)$ đối với Sĩ đỏ.

- Quân Tốt:

+ Không thể đi lùi, nên quân Tốt không thể nằm ở các hàng có thứ tự nhỏ hơn 3 đối với quân xanh, và lớn hơn 6 đối với quân đỏ.

+ Khi chưa sang sông quân Tốt không thể đi ngang, nên Tốt xanh không thể nằm ở cột 1, 3, 5, 7 nếu hàng của nó nhỏ hơn 5; tương tự Tốt đỏ cũng không thể nằm ở các cột kề trên nếu hàng của nó lớn hơn 6.

Trên ví dụ hình 5.9, các quân cờ được đánh dấu \times nằm không đúng vị trí quy định và sẽ được phần mềm điều khiển của robot phát hiện.



Hình 5.8: Các quân cờ nằm không đúng vị trí theo luật (đánh dấu \times).

5.2.3. Phát hiện và kiểm tra nước đi trên bàn cờ

a. Phát hiện nước đi

Sau khi đã chắc chắn bàn cờ hợp lệ, việc so sánh ma trận trạng thái bàn cờ hiện tại (ma trận N) với ma trận trạng thái bàn cờ trước khi người chơi đi quân (ma trận T) như ví dụ bảng 5.3 sẽ trả về thông tin nước đi của người chơi.

Bảng 5.3: Ví dụ so sánh các ma trận trạng thái để phát hiện nước đi quân cờ.

| Vị trí | Ma trận T | Ma trận N | Lý giải |
|--------|-----------|-----------|--|
| (0, 3) | a | . | Quân sĩ xanh (a) di chuyển khỏi vị trí (0, 3) |
| (1, 4) | C | a | Quân sĩ xanh (a) ăn quân pháo đỏ ở vị trí (1, 4) |

Kết luận: Quân sĩ xanh di chuyển từ vị trí (0, 3) đến vị trí (1, 4) để ăn quân pháo đỏ

Như vậy, nếu chỉ có 2 sự khác biệt giữa 2 ma trận N và T thì đã có 1 nước đi được thực hiện, và phần mềm điều khiển có thể phát hiện ra nó. Ngược lại, nếu có nhiều hơn 2 sự khác biệt giữa 2 ma trận này thì đã có nhiều hơn 1 nước đi; nếu không có sự khác biệt nào thì người chơi chưa thực hiện nước đi. Trong cả hai trường hợp này, phần mềm sẽ báo lỗi để người chơi điều chỉnh.

b. Kiểm tra nước đi hợp lệ

Qua việc ma trận trạng thái bàn cờ hiện tại cũng đánh giá được tính hợp lệ của nước đi cờ. Nước đi hợp lệ thỏa mãn các tiêu chí sau:

- Nước đi đúng luật di chuyển: quân Xe, Tốt, Tướng đi và ăn theo đường thẳng; quân Mã đi và ăn theo đường hình chữ L, không bị cản; quân Tốt đi thẳng một ô, khi qua sông có thể đi ngang theo chiều rộng bàn cờ; quân Pháo đi theo đường thẳng, ăn quân bằng việc nhảy qua một quân khác; quân Sĩ và Tượng đi theo đường chéo...
- Nước đi không khiến quân Tướng bị chiếu.
- Nước đi đúng bên, tức là người chơi không thể di chuyển quân cờ của robot và ngược lại

5.2.4. Đánh giá chức năng xác định nước đi của người chơi

Phần này đánh giá chức năng xác định nước đi của người chơi trong 20 trường hợp bàn cờ bất kỳ.

Bảng 5.4: Đánh giá chức năng xác định nước đi trong các trường hợp bất kỳ.

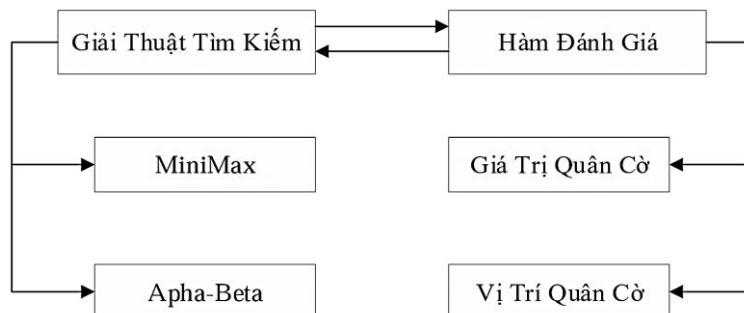
| Trường hợp | Thông báo của phần mềm | Xác nhận |
|------------|--|----------|
| 1 | Bàn cờ không hợp lệ: Quân cờ không nằm trên đường giao | Đúng |
| 2 | Bàn cờ không hợp lệ: Quân cờ không nằm trên đường giao | Đúng |
| 3 | Bàn cờ không hợp lệ: Quân cờ không nằm đúng vị trí | Đúng |
| 4 | Pháo xanh (4, 4) → (6, 4) | Đúng |
| 5 | Mã xanh (2, 1) → (4, 2) | Đúng |
| 6 | Nước đi không hợp lệ: Tướng xanh bị chiếu | Đúng |
| 7 | Nước đi không hợp lệ: Sĩ xanh (0, 3) → (2, 2) | Đúng |

| | | |
|----|--|------|
| 8 | Bàn cờ không hợp lệ: Quân cờ không nằm đúng vị trí | Đúng |
| 9 | Bàn cờ không hợp lệ: Nhiều hơn một nước đi | Đúng |
| 10 | Bàn cờ không hợp lệ: Quân cờ không nằm đúng vị trí | Đúng |
| 11 | Mã xanh (2, 1) → (4, 2) | Đúng |
| 12 | Nước đi không hợp lệ: Xe xanh (0, 3) → (2, 4) | Đúng |
| 13 | Tốt xanh (5, 2) → (5, 3) | Đúng |
| 14 | Nước đi không hợp lệ: Tướng xanh bị chiếu | Đúng |
| 15 | Bàn cờ không hợp lệ: Quân cờ không nằm trên đường giao | Đúng |
| 16 | Bàn cờ không hợp lệ: Lộ mặt Tướng | Đúng |
| 17 | Bàn cờ không hợp lệ: Quân cờ không nằm đúng vị trí | Đúng |
| 18 | Bàn cờ không hợp lệ: Nhiều hơn một nước đi | Đúng |
| 19 | Bàn cờ không hợp lệ: Quân cờ không nằm đúng vị trí | Đúng |
| 20 | Nước đi không hợp lệ: Tướng xanh bị chiếu | Đúng |

Quá trình đánh giá trong 20 trường hợp trên đạt tỉ lệ chính xác 100%. Việc xác định bàn cờ và nước đi của người chơi hợp lệ sẽ đảm bảo trò chơi vận hành đúng, tiết kiệm thời gian tính toán nước đi cho phần mềm điều khiển.

5.3. Giải thuật chơi cờ của robot

Mục này phân tích và phát triển giải thuật chơi cờ cho robot để robot thực hiện nước đi hồi đáp mỗi khi người chơi bấm nút đổi lượt.



Hình 5.9: Sơ đồ tổng quan giải thuật chơi cờ.

Một giải thuật chơi cờ được hình thành dựa trên hai yếu tố: “Giải thuật tìm kiếm” và “Hàm đánh giá”. “Giải thuật tìm kiếm” xây dựng một không gian trạng thái (cây trò chơi) chứa tất cả những nước đi có thể xảy ra ở một độ sâu tìm kiếm nhất định. “Hàm

đánh giá” đưa ra các quy tắc, tiêu chuẩn để lựa chọn nước đi tốt nhất trong không gian trạng thái đó. Sơ đồ tổng quan giải thuật chơi cờ được thể hiện trong hình 5.9.

5.3.1. Giải thuật tìm kiếm

a. Giải thuật Minimax

Minimax là một thuật toán đệ quy được sử dụng phổ biến trong các trò chơi đối kháng 2 người như: cờ Tướng, cờ vua, cờ vây, cờ caro,... với mục đích tìm kiếm nước đi kế tiếp tốt nhất. Hai đối thủ trong trò chơi được gọi là Max và Min luân phiên thực hiện lượt chơi. Chiến lược đi của Max phụ thuộc vào nước đi của Min và ngược lại. Max hướng tới tối ưu hóa lợi thế của mình, trong khi Min hướng tới cực tiểu hóa ưu thế của đối phương. Mỗi trạng thái Max, Min được biểu diễn bởi một nút trên cây trò chơi. Nút lá là nút chứa trạng thái kết thúc của trò chơi. Giá trị tại một nút cha được xác định dựa vào trạng thái hiện tại của nút cha (Max hay Min) và giá trị của các nút con. Nếu nút thuộc lớp Max thì giá trị của nút là giá trị lớn nhất của các nút con. Nếu nút thuộc lớp Min thì giá trị của nút là giá trị nhỏ nhất của các nút con.

Các bước của giải thuật Minimax:

- Xây dựng cây trò chơi cho đến khi đạt đến giới hạn tìm kiếm.
- Xác định giá trị tĩnh tại các nút lá.
- Tính giá trị tĩnh cho các nút cha bên trên.

Nếu nút đang xét thuộc lớp Max, giá trị của nút là giá trị lớp nhất của các nút con. Ghi nhận kết quả lớn nhất.

Nếu nút đang xét thuộc lớp Min, giá trị của nút là giá trị nhỏ nhất của các nút con. Ghi nhận kết quả nhỏ nhất.

Ví dụ trong hình 5.10 biểu diễn một cây trò chơi sử dụng giải thuật Minimax.

- Vị trí gốc của cây trò chơi [a] đại diện cho vị trí hiện tại của người chơi, các nút con [b], [c]... [m], [n] là tổng hợp các tình huống tiếp theo có thể được tạo ra từ vị trí gốc. Một nút trên cây trò chơi có thể vừa là con của nút cha bên trên vừa là cha của các nút con bên dưới.
- Nút gốc [a] thuộc lớp Max, nút [b], [c], [d] thuộc lớp Min, nút [e], [f], [g], [h], [i], [k], [l], [m], [n] thuộc lớp Max. Giá trị tĩnh của các nút lá được xác định và

biểu diễn bên trong hình tròn. Nút [e] thuộc lớp Max và là cha của các nút lá có giá trị tĩnh lần lượt là: [8], [7], [2]. Từ đây tính được giá trị tại nút [e]:

$$e = \max(8, 7, 2) = 8.$$

Tương tự:

$$f = \max(9, 5, 4) = 9,$$

...,

$$n = \max(7, 3, 4) = 7.$$

- Nút [b] thuộc lớp Min và là cha của các nút con [e], [f], [g].

$$b = \min(e, f, g) = \min(8, 9, 4) = 4.$$

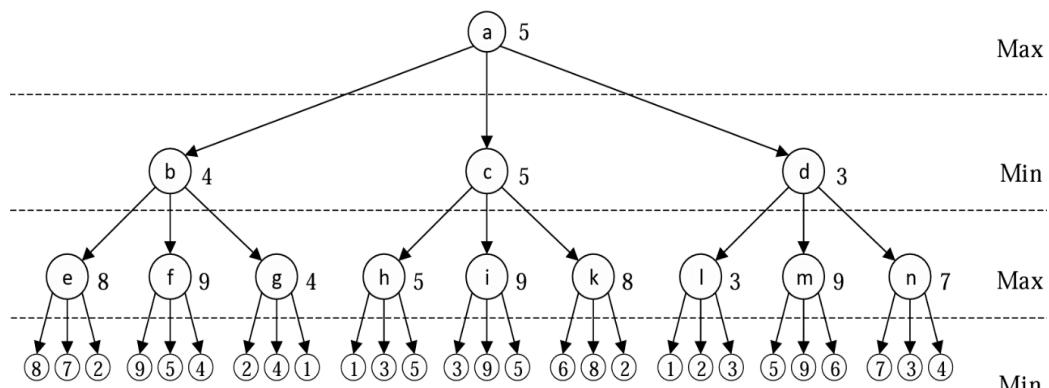
Tương tự:

$$c = \min(h, i, k) = 5, d = \min(l, m, n) = 3.$$

Nút [a] thuộc lớp Max và là cha của các nút con [b], [c], [d].

$$a = \max(b, c, d) = 5.$$

Như vậy, nhánh được lựa chọn trên cây trò chơi hình 5.11 là: a → c → h → 5.



Hình 5.10: Sơ đồ minh họa giải thuật Minimax.

Giải thuật Minimax đã tạo ra một cây trò chơi trong đó chúa tất cả những nước đi có thể xảy ra kể từ trạng thái gốc [a] cho đến khi ván đấu kết thúc. Đây cũng chính là nhiệm vụ của “Giải thuật tìm kiếm”. Còn việc lựa chọn nước đi nào trong cây trò chơi trên đó là nhiệm vụ của “Hàm đánh giá”.

Minimax là một thuật toán tìm kiếm theo độ sâu. Cây trò chơi liên tục được mở rộng cho đến khi đạt đến trạng thái cuối cùng (thắng, hòa, thua). Với một trò chơi phức tạp

như cờ Tướng, việc xây dựng không gian trạng thái từ một nút gốc bất kỳ cho đến khi kết thúc ván đấu là điều không tưởng bởi sự bùng nổ khổng lồ của tổ hợp. Vì thế, trong giải thuật tìm kiếm có khái niệm “độ sâu tìm kiếm (ply)” chính là số nước đi tối đa của người chơi và robot mà máy tính có thể tiên tri được trong một lượt đi. “Độ sâu tìm kiếm” do người lập trình qui định. Nếu tìm kiếm ở độ sâu càng lớn thì khả năng chọn được nước đi tốt càng cao. Tuy nhiên, thời gian tính toán cũng tăng lên. Thời gian thực hiện một nước đi của robot ở các độ sâu khác nhau được thể hiện trong bảng 5.5 [19].

Bảng 5.5: Số nước đi và thời gian tính toán trung bình giải thuật Minimax.

| Độ sâu | Số nước đi trung bình | Thời gian tính toán trung bình |
|--------|-----------------------|--------------------------------|
| 2 | 1600 | < 1 s |
| 3 | 6400 | 1 s |
| 4 | $2,56 \times 10^6$ | 30 s |
| 5 | $1,02 \times 10^8$ | 1000 s |

Bản chất của giải thuật Minimax là một thuật toán vét cạn. Giải thuật này xem xét tất cả các nút trên cây trò chơi kể cả những nút không cần thiết. Đây là nguyên nhân làm cho sự bùng nổ tổ hợp diễn ra nhanh chóng. Xét trong cùng một khoảng thời gian, để tăng độ sâu tìm kiếm đòi hỏi phải giảm số lượng nút xét trên cây trò chơi. Giải thuật Alpha-Beta cho phép làm giảm số nút xét mà không ảnh hưởng đến kết quả cuối cùng.

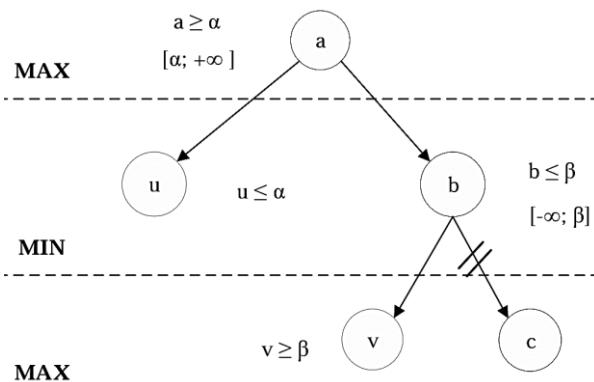
b. Giải thuật Alpha – Beta

Giải thuật Alpha – Beta là một giải thuật cải tiến của giải thuật Minimax nhằm cắt tỉa những nhánh không cần thiết trên cây trò chơi, làm giảm số lượng trạng thái phát sinh, nhờ đó tăng được độ sâu tìm kiếm.

Thuật toán Alpha – Beta tạo ra hai giá trị apha (α) và beta (β) trong quá trình tìm kiếm, alpha gắn với nút Max và có khuynh hướng không giảm. Ngược lại, beta gắn với nút Min và có khuynh hướng không tăng. Một nút trên cây trò chơi có thể vừa là cha của nút con bên dưới, vừa là con của nút cha bên trên. Giải thuật Alpha Beta sử dụng mối liên hệ giữ nút ông, nút cha và nút con trên cùng một nhánh. Trước khi xem xét một nút

con bất kỳ phải kiểm tra miền giá trị $[\alpha, \beta]$ của nút cha và nút ông của nó, nếu miền giá trị bằng rỗng thì ta bỏ qua nút con đó.

Hình 5.11 mô tả ý tưởng thực hiện của giải thuật Alpha – Beta. Giả sử hiện tại đang xét nút c, c có nút cha là b, nút ông là a. Nút a nhận giá trị truyền về từ nút u, nút b nhận giá trị truyền về từ nút v. Nếu giá trị tại nút u lớn hơn giá trị tại nút v (nghĩa là $\alpha > \beta$), thì loại bỏ tất cả các nút con còn lại bên dưới nút b.



Hình 5.11: Sơ đồ mô tả giải thuật cắt tỉa Alpha – Beta.

Trong giải thuật Alpha-Beta, thứ tự xét các nhánh rất quan trọng. Ban đầu alpha, beta có giá trị trong khoảng $(\alpha, \beta) = (-\infty, +\infty)$, trong quá trình tính toán, khoảng giá trị (α, β) sẽ được thu hẹp dần, những giá trị nằm ngoài khoảng (α, β) sẽ bị cắt bỏ. Do đó, nếu chọn nhánh xét ban đầu tốt, khoảng (α, β) nhỏ, số lượng nhánh được cắt tỉa sẽ lớn và ngược lại. Cùng giải thuật Alpha-Beta, nhưng với những trò chơi khác nhau thì nguyên tắc xét duyệt nhánh cũng khác nhau. Trong giải thuật chơi cờ Tướng này, việc xét nước đi nào trước, nước đi nào sau tuân thủ theo hai nguyên tắc sau:

- Nguyên tắc 1: Một nước đi bắt quân sẽ được xét trước. Nếu có nhiều nước đi bắt quân thì nước đi nào bắt được quân có giá trị lớn hơn được xét trước. Giá trị các quân theo thứ tự giảm dần: Vua, Xe, Pháo, Mã, Tốt, Tượng, Sĩ. Nếu có nhiều nước đi cùng bắt một quân hoặc bắt các quân có cùng giá trị thì xét nước đi của quân mạnh hơn trước.
- Nguyên tắc 2: Các nước đi không bắt quân sẽ được xét theo thứ tự: Xe, Pháo, Mã, Tốt, Tượng, Sĩ, Vua. Khi một trong các quân Xe, Pháo, Mã, Tốt có nhiều nước đi thì nước đi nào điểm đến có giá trị vị trí cao hơn sẽ được xét trước. Khi

một trong các quân Tượng, Sĩ, Vua có nhiều nước đi thì xét theo thứ tự vị trí đến. Từ hàng 0 đến hàng 9, cột 0 đến cột 8. Vị trí đến càng gần gốc tọa độ sẽ được xét trước.

Hệ số phân nhánh trung bình của cờ Tướng là 40. Số lượng nút trong cây được tính theo công thức b^n . Trong đó b là hệ số phân nhánh và n là số tầng. Nếu sử dụng kỹ thuật cắt tỉa Alpha – Beta một cách tối ưu thì tổng số nút của cây được tính theo công thức [5]: $b^{\lceil \frac{n}{2} \rceil} + b^{\lfloor \frac{n}{2} \rfloor} - 1$. Bảng 5.6 so sánh số nút phải xét giữa hai thuật toán Minimax và cắt tỉa Alpha – Beta.

Bảng 5.6: Số nút phải xét khi dùng thuật toán Minimax và Alpha – Beta.

| Số tầng | Minimax | Alpha – Beta | Tỉ lệ số nút $\frac{\text{minimax}}{\text{alphabeta}}$ |
|---------|---------------|--------------|--|
| 0 | 1 | 1 | 1,0 |
| 1 | 40 | 40 | 1,0 |
| 2 | 1600 | 79 | 20,2 |
| 3 | 64000 | 1639 | 39,0 |
| 4 | 2569999 | 3199 | 800,3 |
| 5 | 102400000 | 65569 | 1561,7 |
| 6 | 4096000000 | 127999 | 32000,0 |
| 7 | 163840000000 | 2623999 | 620608,4 |
| 8 | 6553600000000 | 5119999 | 1280000,3 |

Như vậy, giải thuật cắt tỉa Alpha – Beta hoàn toàn không chống lại được sự bùng nổ tổ hợp mà chỉ giúp làm chậm quá trình bùng nổ tổ hợp, thuật toán tìm kiếm trở nên gọn nhẹ hơn nhưng hiệu quả vẫn tương đương với giải thuật Minimax.

5.3.2. Hàm lượng giá

Hàm lượng giá sẽ nhận vào một trạng thái và trả ra một số. Số càng lớn thì nghĩa là bên được lượng giá càng có lợi. Hàm lượng giá trò chơi Cờ Tướng được tính toán như sau:

$$Eval(s) = \sum_{i=1}^7 s_i Q_i \quad (5.1)$$

Trong đó:

- s_i : số lượng quân cờ i cùng màu đang hiện diện trên bàn cờ.
- Q_i : giá trị của quân cờ i .

Để tính ra giá trị lượng giá của một trạng thái, dùng điểm lượng giá của bên máy $Eval(m)$ trừ cho điểm lượng giá của bên người chơi $Eval(n)$ nếu đến lượt máy đi hoặc ngược lại. Nếu có thể sinh cây tới lúc biết thắng thua thì trạng thái sẽ có ba giá trị là +1 (máy thắng), 0 (hòa), -1 (máy thua). Tuy nhiên, điều đó chỉ áp dụng được khi cây tìm kiếm có kích thước nhỏ như trò chơi caro. Còn với cờ Tướng và các trò chơi có kích thước cây tìm kiếm lớn thì khả năng thắng thua được lượng giá bằng những đặc điểm của trạng thái đó. Trong cờ Tướng, có rất nhiều yếu tố ảnh hưởng đến chất lượng hàm lượng giá như: sức mạnh của mỗi quân cờ, vị trí mỗi quân cờ trên bàn cờ, kỹ thuật khai cục, tàn cục, tính linh hoạt của một quân cờ trong một thế cờ, sức mạnh kết hợp giữa các quân cờ... Trong phạm vi luận văn này, hàm đánh giá được phát triển dựa trên hai yếu tố: sức mạnh của mỗi quân cờ và vị trí mỗi quân cờ trên bàn cờ.

a. Sức mạnh mỗi quân cờ

Đây là phương pháp cơ bản và quan trọng nhất trong hàm đánh giá. Mỗi quân cờ có vai trò, khả năng khác nhau. Để chơi cờ Tướng, máy tính phải hiểu được cần bảo vệ quân nào, nếu phải hi sinh quân cờ nên lựa chọn quân cờ nào để giảm tổn thất tối đa, hay khi có cơ hội bắt quân nên bắt quân cờ nào của đối phương để tối đa lợi thế. Bằng cách định nghĩa sức mạnh của mỗi quân cờ bằng một số nguyên, quân cờ càng quan trọng thì giá trị số nguyên càng lớn, những quân cờ cùng loại thì có cùng giá trị số nguyên. Từ đó, máy tính phân biệt được sức mạnh giữa các quân cờ trên bàn cờ. Dựa theo những công trình nghiên cứu về cờ Tướng và quá trình thực nghiệm, giá trị từng quân cờ được định nghĩa trong bảng 5.7 [19].

Bảng 5.7: Giá trị mỗi quân cờ.

| Tướng | Sĩ | Tượng | Mã | Xe | Pháo | Tốt |
|-------|-----|-------|-----|-----|------|----------|
| 6000 | 120 | 120 | 270 | 600 | 285 | 60 (100) |

Trong số bảy quân cờ, Tướng là quan trọng nhất, trong mọi trường hợp việc mất Tướng sẽ dẫn tới thua cuộc, vì vậy giá trị của Tướng phải vượt xa các quân còn lại. Trong bốn

quân tấn công xe, pháo, Mã, tốt, tổng giá trị hai quân pháo, Mã xấp xỉ bằng giá trị quân Xe. Quân Tốt có giá trị ban đầu là 60, nhưng khi đã sang sông thì giá trị của nó là 100. Quân pháo có giá trị lớn hơn quân Mã. Quân Tượng có giá trị bằng quân Sĩ.

Sức mạnh của một quân cờ được thể hiện ở khả năng tấn công, phòng thủ, phạm vi hoạt động của nó. Một quân cờ vừa có khả năng tấn công mạnh, vừa có khả năng phòng thủ tốt đồng thời phạm vi hoạt động rộng thì giá trị của nó càng cao. Ví dụ: dựa vào bảng lượng giá sức mạnh quân cờ này, trong cùng một nước đi, xét ở độ sâu tìm kiếm $Ply = 1$, khi máy tính hoặc có thể bắt được quân Xe hoặc có thể bắt được quân Mã của đối phương thì máy tính sẽ chọn bắt quân Xe vì giá trị của quân Xe cao hơn quân Mã.

Để biết được đâu là nước đi tốt nhất trong tập hợp những nước đi khả thi, máy tính sẽ tiến hành đi thử từng nước đi. Sau mỗi nước đi thử máy tính sẽ tính toán các giá trị $Eval(m)$, $Eval(n)$, $value$ sau đó ghi lại giá trị $value$. Ở lượt của máy, $value$ càng lớn thì nước đi đó càng có lợi cho máy. Bằng cách lựa chọn giá trị $value$ lớn nhất trong các giá trị $value$ trả về, máy tính có được nước đi tốt nhất. Quá trình đi thử được xử lý bên trong máy tính, không hiển thị nước đi thử ra màn hình. Trong một nước đi thử, sau khi có được giá trị $value$, máy tính sẽ khôi phục lại trạng thái bàn cờ gốc trước khi thực hiện nước đi thử tiếp theo. Tuy nhiên, phương pháp này có một nhược điểm lớn, bởi vì nước đi tốt nhất là nước đi có giá trị $value$ lớn nhất. Do đó trong trường hợp nhiều nước đi cùng trả về một giá trị $value$, không có căn cứ hiệu quả để lựa chọn nước đi tốt trong các nước đi ấy.

b. Vị trí quân cờ

Một ván đấu cờ Tướng được chia làm 3 gia đoạn: khai cuộc, trung cuộc, tàn cuộc. Thông thường, trong một ván cờ Tướng mỗi bên sẽ có 40 - 50 nước đi. Trong đó chế độ khai cuộc được định nghĩa là 8 - 12 nước đi đầu tiên. Mục tiêu của giai đoạn khai cuộc không phải là tìm mọi cách để chiếu bí Tướng đối phương mà là khai triển quân một cách hợp lý, chính xác. Trong giai đoạn khai cục các quân phải cố giành được những vị trí thuận lợi nhất trên bàn cờ tạo nền tảng vững chắc trước khi chuyển sang giai đoạn trung cuộc. Giá trị của các quân cờ ở bảng 5.7 mang tính chất tuyệt đối, nghĩa là so sánh sức mạnh giữa chúng trên một bàn cờ trống, mà không đặt trong một thế cờ cụ thể nào. Tuy nhiên

trong một ván cờ thực tế, ở từng giai đoạn cụ thể, với những vị trí đứng khác nhau, giá trị của từng quân cờ sẽ thay đổi. Nó có thể tăng cao hơn, nếu quân cờ đứng ở vị trí tốt, hoặc cũng có thể giảm đi, nếu quân cờ đứng ở vị trí xấu. Đây được coi là “thế” của quân cờ. Vì vậy, trong một thế cờ cụ thể ngoài xem xét giá trị vốn có của quân cờ còn phải đánh giá “thế” của quân cờ đó trước khi quyết định thực hiện nước đi.



Hình 5.12: Vị trí ban đầu của các quân cờ.

Việc xác định giá trị quân cờ theo vị trí là rất quan trọng. Tuy nhiên, để xây dựng một thuật toán tổng quát nhằm xác định “thế” của mỗi quân cờ trong từng thế cờ cụ thể là quá lớn. Trong phạm vi luận văn sẽ xem xét “thế” của quân cờ trong giai đoạn khai cục, giá trị vị trí này được xây dựng dựa theo bàn cờ lúc khởi tạo như hình 5.12. Dựa theo một số công trình nghiên cứu về cờ Tướng và kinh nghiệm chơi cờ thực tế, giá trị “thế” của các quân Xe, Pháo, Mã, Tốt được thể hiện ở hình 5.13. [19]

Cùng một quân cờ, một vị trí được đánh giá cao hơn các vị trí khác khi tại đó quân cờ có thể trực tiếp hoặc gián tiếp uy hiếp Tướng của đối phương, chốt giữ những vị trí quan trọng, có khả năng di chuyển rộng kết hợp với các quân cờ khác tạo thành thế tấn công, phòng thủ. Giá trị vị trí của quân cờ hình 5.13 được xây dựng dựa trên thế cờ lúc khởi tạo do đó cực kỳ hiệu quả cho những nước đi đầu tiên, càng về sau độ chính xác càng giảm. Bản chất hàm đánh giá theo giá trị quân cờ là tìm mọi cách tối đa hóa lợi thế, bắt được quân càng giá trị càng tốt mà không quan tâm đến “thế” của quân cờ. Khi áp đặt

điều này vào giai đoạn khai cục dẫn đến máy tính không khai triển được quân cờ hợp lý, từ đó đánh mất lợi thế ban đầu.

| | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|--|----|---|----|-----|-----|-----|----|---|----|
| -2 | 10 | 6 | 14 | 12 | 14 | 6 | 10 | -2 | | 0 | 0 | 2 | 6 | 6 | 6 | 2 | 0 | 0 |
| 8 | 4 | 8 | 16 | 8 | 16 | 8 | 4 | 8 | | 0 | 2 | 4 | 6 | 6 | 6 | 4 | 2 | 0 |
| 4 | 8 | 6 | 14 | 12 | 14 | 6 | 8 | 4 | | 4 | 0 | 8 | 6 | 10 | 6 | 8 | 0 | 4 |
| 6 | 10 | 8 | 14 | 14 | 14 | 8 | 10 | 6 | | 0 | 0 | 0 | 2 | 4 | 2 | 0 | 0 | 0 |
| 12 | 16 | 14 | 20 | 20 | 20 | 14 | 16 | 12 | | -2 | 0 | 4 | 2 | 6 | 2 | 4 | 0 | -2 |
| 12 | 14 | 12 | 18 | 18 | 18 | 12 | 14 | 12 | | 0 | 0 | 0 | 2 | 8 | 2 | 0 | 0 | 0 |
| 12 | 18 | 16 | 22 | 22 | 22 | 16 | 18 | 12 | | 0 | 0 | -2 | 4 | 10 | 4 | -2 | 0 | 0 |
| 12 | 12 | 12 | 18 | 18 | 18 | 12 | 12 | 12 | | 2 | 2 | 0 | -10 | -8 | -10 | 0 | 2 | 2 |
| 16 | 20 | 18 | 24 | 26 | 24 | 18 | 20 | 16 | | 2 | 2 | 0 | -4 | -14 | -4 | 0 | 2 | 2 |
| 14 | 14 | 12 | 18 | 16 | 18 | 12 | 14 | 14 | | 6 | 4 | 0 | -10 | -12 | -10 | 0 | 4 | 6 |

a. Giá trị vị trí của quân Xe xanh.

b. Giá trị vị trí của quân Pháo xanh.

| | | | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|--|----|----|----|----|-----|----|----|----|----|
| 0 | -4 | 0 | 0 | 0 | 0 | 0 | -4 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 2 | 4 | 4 | -2 | 4 | 4 | 2 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 2 | 8 | 8 | 4 | 8 | 8 | 2 | 4 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 6 | 8 | 6 | 10 | 6 | 8 | 6 | 2 | | 0 | 0 | -2 | 0 | 4 | 0 | -2 | 0 | 0 |
| 4 | 12 | 16 | 14 | 12 | 14 | 16 | 12 | 4 | | 2 | 0 | 8 | 0 | 8 | 0 | 8 | 0 | 2 |
| 6 | 16 | 18 | 14 | 16 | 14 | 18 | 16 | 6 | | 6 | 12 | 18 | 18 | 20 | 18 | 18 | 12 | 6 |
| 8 | 24 | 18 | 24 | 20 | 24 | 18 | 24 | 8 | | 10 | 20 | 30 | 34 | 40 | 34 | 30 | 20 | 10 |
| 12 | 14 | 16 | 20 | 18 | 20 | 16 | 14 | 12 | | 14 | 26 | 42 | 60 | 80 | 60 | 42 | 26 | 14 |
| 4 | 10 | 28 | 16 | 8 | 16 | 28 | 10 | 4 | | 18 | 36 | 56 | 80 | 120 | 80 | 56 | 36 | 18 |
| 4 | 8 | 16 | 12 | 4 | 12 | 16 | 8 | 4 | | 0 | 3 | 6 | 9 | 12 | 9 | 6 | 3 | 0 |

c. Giá trị vị trí của quân Mã xanh.

d. Giá trị vị trí của quân Tốt xanh.

Hình 5.13: Giá trị theo vị trí của các quân cờ.

Với những ý tưởng cơ bản đã đề cập, các kĩ thuật trong những mục tiếp theo sẽ hướng tới việc tối ưu tìm kiếm trên cây và tính toán hàm lượng giá tốt hơn, giúp tiết kiệm thời gian và tài nguyên.

5.3.3. Cắt nhánh khi mất quân tướng

Quân tướng là quân quan trọng nhất bàn cờ, bởi vậy cần xây dựng giải pháp để phần mềm chơi cờ xem quân tướng là quân đặc biệt, không bao giờ được đẻ mất. Theo đó, ở mỗi nước đi, ưu tiên thứ ăn Tướng đối phương. Nếu có nước đi hợp lệ, không cần phải xét tới các nhánh con khác nữa. Tương tự, phần mềm chơi cờ cũng sẽ tránh đẻ đối phương có thể tấn công Tướng của mình trong những lượt tiếp theo. Như vậy, nếu có nhiều đường thắng thì phần mềm sẽ chọn đường thắng nhanh nhất, nếu có nhiều đường

thua thì sẽ chọn đường thua lâu nhất. Điều này giống với cách những người chơi tư duy ngoài thực tế.

5.3.4. Hiện tượng horizon effect và giải thuật quiescence search

Horizon effect là hiện tượng gặp phải tìm kiếm bằng giải thuật Alpha – Beta với số tầng cố định.

Xét một ví dụ như sau: Giả sử bên đi trước là đỏ. Nếu phần mềm chơi cờ duyệt một tầng thì nó sẽ dùng xe đỏ để ăn pháo đen. Tuy nhiên nếu 2 tầng thì phần mềm sẽ thấy xe đen ăn lại pháo đỏ. Nếu duyệt 3 tầng thì AI thấy mã đỏ có thể ăn xe đen lại. Quiescence search sẽ giải quyết được vấn đề này (horizon effect) bằng cách gia tăng số tầng Alpha-Beta đến khi nào bàn cờ trở nên "ổn định". Giải thuật quiescence search sẽ được gọi ở nút là nếu có chiếu hoặc bắt quân ở nút lá. Queiescence search sinh ra tất cả nước bắt quân và chiếu đến khi nào không còn nước bắt quân và chiếu nào nữa.

Quiescence search sẽ tăng thời gian tìm kiếm. Trong đa số các trường hợp thì thời gian tìm kiếm sẽ tăng 20% - 30% [19]. Tuy nhiên, đổi lại thì chương trình sẽ thông minh hơn và ít khi bị bắt quân một cách dễ dàng.

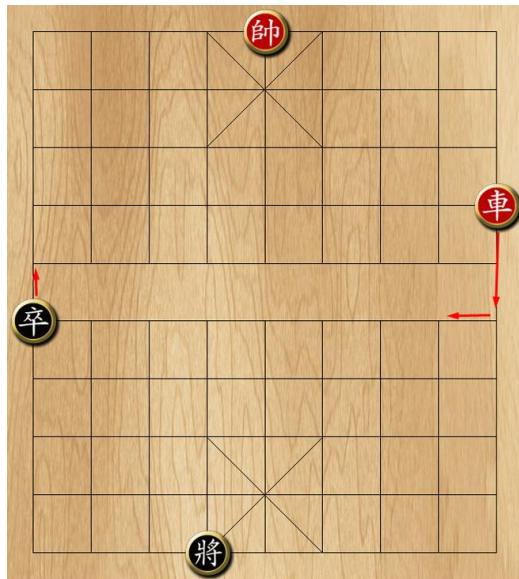
5.3.5. Cải tiến Alpha – Beta bằng transposition table

Chương trình chơi cờ tìm kiếm nước đi bằng cách xây dựng một cây tìm kiếm lớn. Đây thực chất là một đồ thị có chu trình, bởi với mỗi trạng thái bàn cờ, sau một số hoán vị nước đi sẽ trả về lại trạng thái đó. Ví dụ như trên hình 5.14, bên đỏ đi trước {xe đỏ tiến 2, tốt đen tiến 1, xe đỏ bình 1} sinh ra một bàn cờ giống hệt với cách đi {xe đỏ bình 1, tốt đen tiến 1, xe đỏ tiến 2}. Việc hoán vị một chuỗi nước đi đều dẫn tới cùng một trạng thái gọi là transposition.

Ý tưởng của bảng transposition table là khi duyệt qua một nút cây sẽ lưu thông tin của nút đó, đến khi duyệt một nút tương tự thì có thể sử dụng lại, từ đó tiết kiệm thời gian tính toán. Transposition table thường được hiện thực bằng mảng một chiều. Mỗi phần tử của mảng chứa thông tin như sau:

- Khóa: Khóa này là độc nhất, được sinh ra bằng cách mã hóa bàn cờ theo giải thuật Zobrist hoặc BCH.
- Nước đi: Lưu lại nước đi tốt nhất mỗi lần duyệt Alpha – Beta trên bàn cờ.

- Điểm: mỗi bàn cờ trong quá trình duyệt AlphaBeta đều có điểm lượng giá (điểm này có thể là điểm thực hoặc điểm được truyền ngược lên từ những nút con của nó).
- Cờ: Có ba loại cờ là:
 - Chính xác: điểm của trạng thái thỏa mãn điều kiện: $\alpha < \text{điểm} < \beta$.
 - Cận dưới: điểm trạng thái thỏa mãn điều kiện: $\text{điểm} \leq \alpha$.
 - Cận trên: điểm trạng thái thỏa mãn điều kiện: $\text{điểm} \geq \beta$.
- Độ sâu: khoảng cách từ trạng thái đang xét tới trạng thái gốc.



Hình 5.14: Bàn cờ bị lặp trạng thái.

Trước khi duyệt một trạng thái, cần truy xuất trong bảng transposition table có trạng thái đó chưa? Xét các trường hợp như sau:

- Nếu độ sâu trong bảng transposition table nhỏ hơn hoặc bằng trạng thái đang duyệt và có cờ là cờ chính xác, lập tức trả về điểm trong transposition table.
- Nếu độ sâu trong bảng transposition table nhỏ hơn hoặc bằng độ sâu trạng thái đang duyệt, có cờ là cận dưới và $\text{điểm} > \alpha$ của trạng thái đang duyệt thì điều chỉnh $\alpha = \text{điểm}$.
- Nếu độ sâu trong bảng transposition table nhỏ hơn hoặc bằng độ sâu trạng thái đang duyệt, có cờ là cận dưới và $\text{điểm} < \beta$ của trạng thái đang duyệt thì điều chỉnh $\beta = \text{điểm}$.

- Nếu độ sâu trong bảng transposition table lớn hơn độ sâu trạng thái đang duyệt thì chỉ dùng được nước đi trong bảng để áp dụng tính toán.

5.3.6. Kiểm soát thời gian duyệt bằng giải thuật iterative depth first search

Với việc tiếp cận cây bằng số tầng biết trước thì không thể kiểm soát được thời gian. Ví dụ đầu ván chơi, việc sinh 4 tầng có thể tốn hơn 1 phút, nhưng đến cuối trận, việc này chỉ mất vài chục giây. Do đó hướng tiếp cận bổ sung thêm giới hạn thời gian tìm kiếm đối với mỗi chế độ là giải pháp hiệu quả.

Từ đây, có thể quy định các mức độ chơi từ dễ đến khó của trò chơi theo thời gian tính toán như bảng 5.8 sau:

Bảng 5.8: Thời gian giới hạn của giải thuật chơi cờ với các mức độ khó khác nhau.

| Mức độ | Dễ | Trung bình | Khó |
|-----------|-----|------------|------|
| Thời gian | 1 s | 2 s | 15 s |

5.3.7. Giải thuật Zobrist Hashing

Khi sử dụng một số giải pháp như transposition table hay cơ sở dữ liệu sẽ dẫn đến nhu cầu lưu trữ bàn cờ lại để truy xuất nhanh chóng và tiết kiệm bộ nhớ. Phương pháp Zobrist Hashing cho phép tạo ra một giá hash duy nhất từ một bàn cờ để tiện trong việc lưu trữ và truy xuất. Giải thuật trải qua 2 bước:

- Sinh ra một dãy các số đại diện cho loại quân cờ và vị trí của nó trong bàn cờ.
- Tính XOR mỗi quân trên bàn cờ với giá trị hash, giá trị hash ban đầu bằng 0.

Như vậy kỹ thuật này không cần phải tính lại giá trị Zobrist hash từ đầu sau mỗi lần di chuyển quân cờ mà sẽ cập nhật lại Zobrist Hash bằng cách dùng toán tử XOR. Khi muốn xóa một quân cờ khỏi giá trị Zobrist hash thì chỉ cần dùng hash XOR với số đại diện cho quân cờ tại vị trí đó. Bởi vì khi XOR một giá trị hai lần với cùng một số thì sẽ được giá trị ban đầu. Ví dụ $1110 \text{ XOR } 1001 = 0111$, $0111 \text{ XOR } 1001 = 1110$.

5.3.8. Cơ sở dữ liệu

Cơ sở dữ liệu – Book trong các phần mềm chơi cờ tướng là tập hợp dữ liệu của các thế cờ và nước đi tương ứng do người làm Book thu thập từ các cuốn sách về cờ tướng, các giải đấu hay các phần mềm cờ tướng. Trong luận văn này, các Book khai cuộc và tàn

cuộc đã được cài đặt. Khi khai cuộc, phần mềm chơi cờ cần sinh ra một lượng tầng và nút rất lớn nên thời gian tính toán sẽ rất lâu, sử dụng Book sẽ cải thiện vấn đề này.

Cơ sở dữ liệu sẽ được mã hóa sử dụng giải thuật Zobrist hashing, giúp tối ưu bộ nhớ lưu trữ và thời gian truy xuất.

5.3.9. Phân chia các cấp độ chơi cờ

Như đã đề cập, trong luận văn này, việc phân chia các cấp độ chơi từ dễ đến khó của phần mềm chơi cờ có thể dựa theo thời gian duyệt tìm kiếm nước đi như trên bảng 5.8. Thời gian tìm kiếm lâu hơn đồng nghĩa với độ sâu tìm kiếm lớn hơn, số nước đi tối đa mà phần mềm có khả năng tính toán được nhiều hơn. Tổng kết lại, phân chia các cấp độ chơi cờ thể hiện trong bảng 5.9. Cả ba giải thuật đều sử dụng chung hàm đánh giá, giải thuật tìm kiếm alpha-beta và các kĩ thuật cắt tỉa, tìm kiếm tăng cường đã đề cập, chỉ khác nhau ở độ sâu tìm kiếm mặc định và thời gian tìm kiếm.

Bảng 5.9: Phân chia giải thuật chơi cờ ở 3 cấp độ khác nhau.

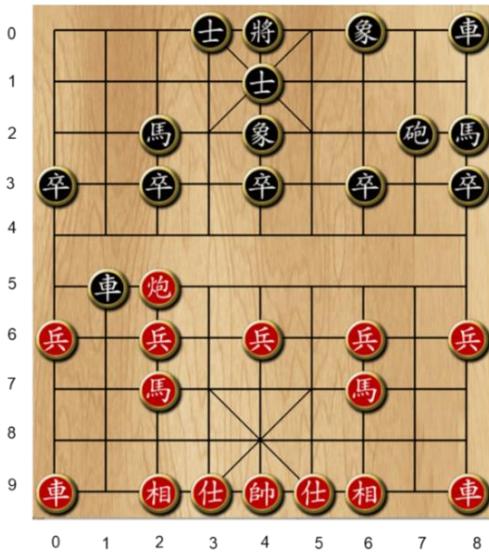
| Cấp độ chơi | Độ sâu tìm kiếm mặc định | Thời gian tìm kiếm giới hạn |
|-------------|--------------------------|-----------------------------|
| Dễ | 3 | 1 s |
| Trung bình | 4 | 2 s |
| Khó | 5 | 15 s |

Để so sánh khả năng chơi cờ của các chế độ, xét hai thế cờ thể cờ ví dụ như trên hình 5.15 và 5.16. Cờ xanh (đen) của người chơi, cờ đỏ của phần mềm chơi cờ. Nước đi tiếp theo là của phần mềm. Phần mềm lần lượt thực hiện nước đi cờ ở các chế độ dễ, trung bình, khó.

Xét thế cờ hình 5.15, kết quả thực hiện nước đi của các chế độ chơi được thể hiện trong bảng 5.10.

Bảng 5.10: Nước đi phản hồi thế cờ hình 5.15 của phần mềm theo các chế độ.

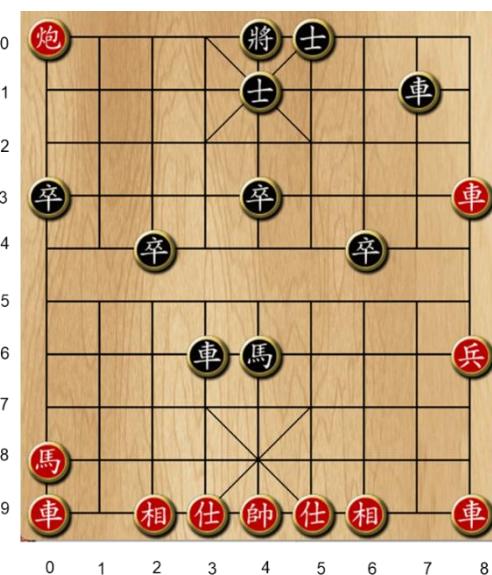
| Chế độ | Dễ | Trung bình | Khó |
|---------|----------------------|--------------------|--------------------|
| Nước đi | Pháo (5, 2) – (2, 2) | Xe (9, 0) – (9, 1) | Xe (9, 8) – (9, 7) |



Hình 5.15: Thé cờ (1) so sánh nước đi giữa các chế độ chơi.

- Nước đi ở chế độ dễ không hề tốt, việc dùng Pháo (5, 2) bắt Mã đối phương (2, 2) không sẽ mang lại lợi thế bởi vì nó sẽ ngay lập tức sẽ bị Pháo (2, 7) bắt lại.
- Nước đi ở chế độ trung bình thể hiện ý định đổi quân Xe, nếu đổi quân thành công nước đi này có thể giúp quân Pháo (5, 1) thoát khỏi mối đe dọa từ quân Xe (5, 2).
- Nước đi ở chế độ khó thông minh hơn, việc ra Xe (9, 8) – (9, 7) sẽ uy hiếp trực tiếp Pháo (2, 7) của đối phương. Pháo (2, 7) đang giữ Mã (2, 2), tiêu diệt được quân này sẽ mang lại nhiều lợi thế cho máy.

Xét thé cờ ví dụ hình 5.16.



Hình 5.16: Thé cờ (2) so sánh nước đi giữa các chế độ chơi.

Ở đây, lực lượng hai bên khá cân bằng, đều có khả năng mở đợt tấn công sâu vào cung tướng của đối phương và giành chiến thắng chung cuộc. Một đòn tấn công mạnh của quân Xanh là kết hợp sức mạnh của quân Xe (1, 7) và quân Mã (6, 4). Nếu quân xe (1, 7) di chuyển xuống vị trí (8, 7), đồng thời Mã di chuyển (6, 4) sang (8, 3) rồi tiếp tục di chuyển về (7, 5) thì quân Tướng của robot sẽ rơi vào trạng thái bị chiếu bí.

Một đòn tấn công khác là kết hợp ba quân Xe (6, 3), Mã (6, 4) và Tướng (0, 4). Đầu tiên Mã xanh di chuyển (6, 4) sang (7, 6) để bắt chết vị trí (8, 4) không cho Tướng đỏ tiến lên, tiếp đó Tướng xanh (0, 4) di chuyển sang (0, 3), rồi Xe (6, 3) bắt quân Sĩ (9, 3), đồng thời chiếu bí Tướng đỏ. Ở thế cờ này, nếu không thận trọng thì trong vòng ba nước đi của người chơi, robot sẽ bị chiếu bí.

Các lựa chọn nước đi của robot được thể hiện trong bảng 5.11.

Bảng 5.11: Nước đi phản hồi thế cờ hình 5.16 của phần mềm theo các chế độ.

| Chế độ | Dễ | Trung bình | Khó |
|---------|--------------------|--------------------|--------------------|
| Nước đi | Sĩ (9, 3) – (8, 4) | Xe (3, 8) – (3, 4) | Xe (9, 8) – (9, 7) |

- Ở chế độ dễ, robot lựa chọn nước đi Sĩ (9, 3) – (8, 4). Đây là nước đi nhằm tránh hướng tấn công của Xe (6, 3). Đây là nước đi không cần thiết, vì quân Xe (6, 3) của đối phương không thể trực tiếp bắt Sĩ (9, 3), bởi sẽ bị Tướng (9, 4) bắt lại. Tuy nhiên, do thời gian hạn chế nên máy chưa thể tính toán sâu, không thể phát hiện ra điều này.

- Chế độ trung bình chọn nước đi Xe (3, 8) – (3, 4), bắt Tốt (3, 4) của người chơi, đồng thời gián tiếp đe dọa Tướng xanh (0, 4). Đây chưa phải nước đi tốt nhất, nhưng là một nước đi rất có lợi cho robot.

- Chế độ khó lựa chọn nước đi Xe (9, 0) – (9, 1), mục đích là tiếp tục di chuyển Xe từ (9, 1) đến (0, 1) để chiếu Tướng xanh. Nước đi tốt nhất của người chơi trong trường hợp này là đưa Xe (6, 3) – (0, 3) để che chắn. Pháo đỏ (0, 1) sẽ bắt Xe xanh. Nếu người chơi dùng Sĩ (1, 4) bắt lại Pháo, Xe đỏ (3, 8) sẽ bắt Tốt (3, 4), chiếu Tướng xanh rồi uy hiếp Mã (6, 4).

Như vậy, trong 3 chế độ chơi, chế độ khó luôn đưa ra những nước đi thông minh hơn hẳn, buộc người chơi phải phân tích và suy luận nhiều hơn.

5.4. Kết luận

Chương này đã hoàn thành việc xây dựng giải thuật sắp xếp quân cờ một cách nhanh chóng và ổn định, robot đã có khả năng nhận diện và đánh giá nước đi của người chơi, từ đó đưa ra các nước đi phản hồi phù hợp theo từng mức độ khó được cài đặt. Hoạt động và chức năng của robot đã hoàn thiện.

CHƯƠNG 6. TỔNG KẾT

Chương này tổng kết và đánh giá những kết quả đã đạt được trong luận văn, ghi nhận những hạn chế còn tồn tại và đưa ra những hướng phát triển của đề tài.

6.1. Kết quả

Những kết quả đạt được từ luận văn như sau:

- Tìm hiểu tổng quan về các đề tài robot chơi cờ tướng hiện có trên thế giới và đặc biệt là các robot chơi cờ đã được thực hiện trong nước, bao gồm ưu, nhược điểm của từng loại.
- Tìm hiểu và khôi phục thành công phần cơ khí và hệ thống điện của robot.
- Tìm hiểu và áp dụng các kỹ thuật xử lý ảnh vào việc xây dựng chương trình để thành công nhận diện tên và vị trí từng quân cờ trên bàn với thời gian trung bình 41ms / quân cờ, cải thiện đáng kể so với những đề tài trước đó; thành công xây dựng phương pháp hiệu chỉnh chương trình xử lý ảnh để áp dụng trong các môi trường ánh sáng khác nhau.
- Xây dựng giải thuật điều khiển đầu công tác bằng PLC, thành công thực nghiệm di chuyển đầu công tác và gấp thả quân cờ, sai số không quá 1.5 mm.
- Tìm hiểu và xây dựng giao thức truyền thông giữa PLC và máy tính, thành công điều khiển đầu công tác cũng như đọc/ghi các giá trị với PLC từ máy tính.
- Xây dựng giải thuật xếp cờ hiệu quả và tiến hành kiểm nghiệm kết quả xếp cờ thực tế đạt kết quả tốt, thời gian xếp cờ trung bình 143,4 s.
- Hoàn thành phát triển giải thuật chơi cờ với ba mức độ từ dễ đến khó, thời gian tính toán cho mỗi chế độ lần lượt được cài đặt là 2 s, 5 s và 10 s.
- Thực hiện chơi hoàn chỉnh một ván cờ, thực nghiệm nhiều lần với nhiều người chơi khác nhau và ở độ khó khác nhau.
- Xây dựng được giải thuật nhận biết nước đi sai bao gồm: nước đi sai luật, đặt quân sai vị trí trên bàn cờ, ăn nhiều quân cùng lúc và không thoát khỏi chiếu tướng.
- Tìm hiểu và hoàn thành xây dựng một giao thức kết nối các thành phần bao gồm: chương trình xử lý ảnh, chương trình điều khiển PLC và chương trình chơi cờ.

6.2. Hạn chế

Bên cạnh những kết quả đạt được, luận văn vẫn còn tồn tại một số hạn chế như sau:

- Quá trình gấp thả quân cờ gây nhiều tiếng ồn.
- Giải thuật chơi cờ phụ thuộc nhiều vào tốc độ xử lý của phần cứng máy tính, trong khi cấu hình máy tính còn hạn chế.
- Kích thước của robot còn lớn so với một số sản phẩm trên thị trường.

6.3. Hướng phát triển

Hệ thống robot chơi cờ tướng vẫn có thể được cải tiến theo một số hướng sau đây:

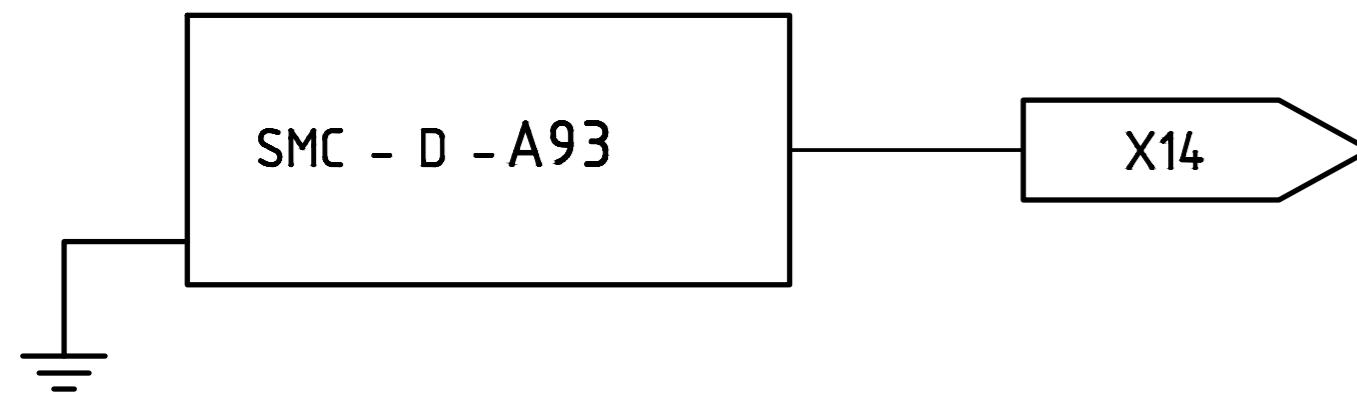
- Tìm hiểu một phương pháp tiêu âm khí nén hiệu quả hoặc thay cơ cấu chấp hành khí nén bằng một loại cơ cấu khác.
- Ứng dụng điện toán đám mây và máy tính nhúng để lưu trữ và vận hành phần mềm chơi cờ nhằm tăng tốc độ xử lý.
- Nghiên cứu và ứng dụng giải thuật chơi cờ được ứng dụng phổ biến hiện tại như cây Monte – Carlo kết hợp học tăng cường để tối ưu thời gian tìm kiếm nước đi.
- Ứng dụng kỹ thuật đa luồng, đa tiến trình chuyên đổi lập trình trên các ngôn ngữ như C/C++ để tăng cường hiệu suất, tối ưu hiệu quả sử dụng phần cứng.

TÀI LIỆU THAM KHẢO

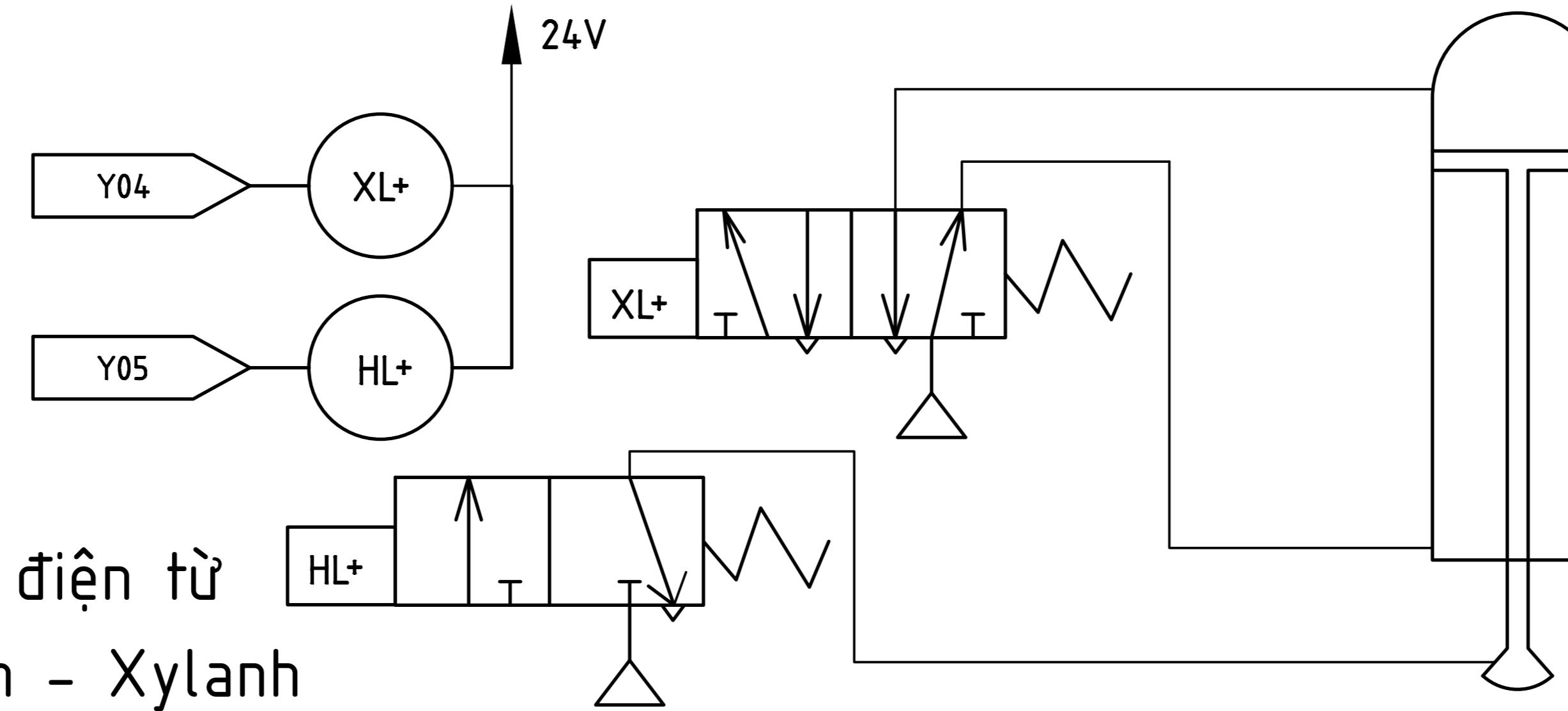
- [1] Hoàng Ngọc, “SEA Games 31: Khởi tranh môn cờ tướng”.
<https://baotintuc.vn/the-thao/sea-games-31-khoi-tranh-mon-co-tuong-20220514123811110.htm>
- [2] Q.Minh, “Giải Cờ tướng trung cao tuổi toàn quốc năm 2020: Sân chơi bổ ích cho Người cao tuổi”.
<https://tdtt.gov.vn/article/giai-co-tuong-trung-cao-tuoi-toan-quoc-nam-2020-san-choi-bo-ich-cho-nguoicao-tuoi-1>
- [3] TTXVN, “Sử dụng nhiều máy tính, điện thoại ảnh hưởng xấu tới sức khỏe thanh thiếu niên”.
<https://vtv.vn/cong-nghe/su-dung-nhieu-may-tinh-dien-thoai-anh-huong-xau-toi-suc-khoe-thanh-thieu-nien-20210811070016854.htm>
- [4] “SenseTime Creates the First AI Chinese Chess Robot for Family Entertainment with Endorsement from Olympic Gold Medalist Guo Jingjing”.
<https://www.sensetime.com/en/news-detail/51164633?categoryId=1072>
- [5] “A Robotic Chinese Chess(Xiangqi) Master”.
<https://www.hkmu.edu.hk/st/computing/fyp/a-robotic-chinese-chessxiangqi-master/>
- [6] “First Prize of Arm Technology Enterprise Proposition: Chinese chess robot game system based on EAIDK-610”.
<http://aijishu.com/a/1060000000361674>
- [7] Vũ Văn Khắc, Lê Bạch Quý Cường (2013), “Robot chơi cờ tướng”, Luận văn đại học, Đại học Bách Khoa TP. HCM.
- [8] Lê Phương Trà, Nguyễn Thành Việt (2018), “Phân tích và thiết kế robot chơi cờ tướng”, Luận văn đại học, Đại học Bách Khoa TP.HCM.

- [9] Nguyễn Văn An, Hoàng Minh cường (2018), “Phân tích và phát triển giải thuật robot chơi cờ tướng”, Luận văn đại học, Đại học Bách Khoa TP.HCM.
- [10] “RGB color model”.
https://en.wikipedia.org/wiki/RGB_color_model
- [11] How to Convert an RGB Image to Grayscale.
https://e2eml.school/convert_rgb_to_grayscale
- [12] Guillermo Sapiro, Mubarak Shah, Derek Hoiem, “Digital Image Analysis Lecture 25: Hough Transform”.
- [13] Virendra Kumar Yadav, Saumya Batham, A. K. Acharya, Rahul Paul, “Approach to accurate circle detection: Circular Hough Transform and Local Maxima concept”, 2014 International Conference on Electronics and Communication Systems (ICECS)
- [14] Shervine Amidi, “Deep learning cheatsheet”.
<https://stanford.edu/~shervine/lvi/teaching/cs-229/cheatsheet-deep-learning>
- [15] “MNIST database”.
https://en.wikipedia.org/wiki/MNIST_database
- [16] Asif Tandel, “VGG-Net Architecture”.
<https://www.linkedin.com/pulse/vgg-net-architecture-asif-tandel>
- [17] “Module: tf.keras”.
https://www.tensorflow.org/api_docs/python/tf/keras
- [18] Trịnh Duy Thanh, “OPC server là gì? – Cái nhìn tổng quan mới nhất về OPC”.
<https://bkhost.vn/blog/opc-server-la-gi/>
- [19] Shi-Jim Yen, “Computer Chinese Chess”, ICGA Journal. No 10, March 2004.

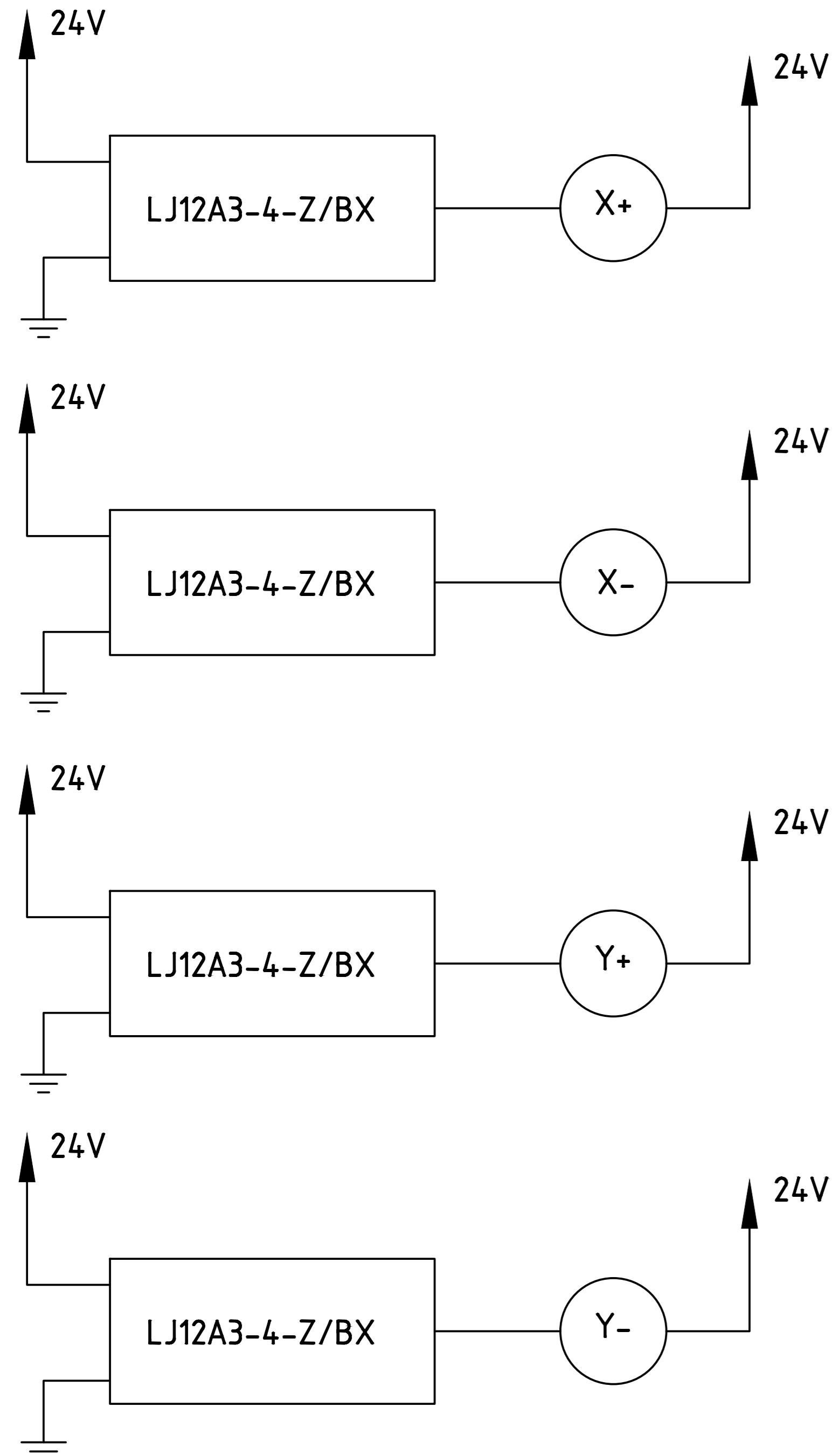
Cảm biến từ xylanh



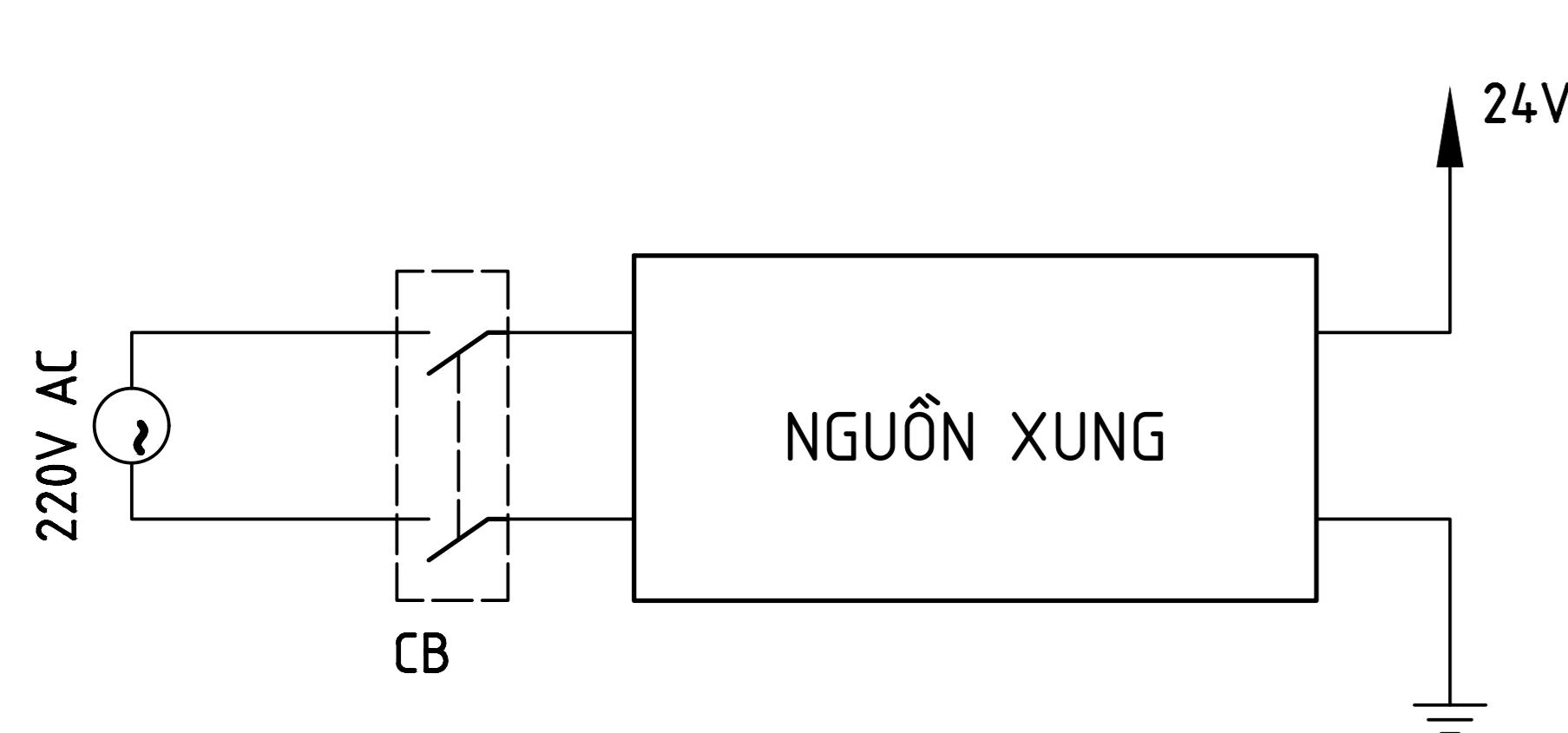
Van điện từ
khí nén - Xylanh



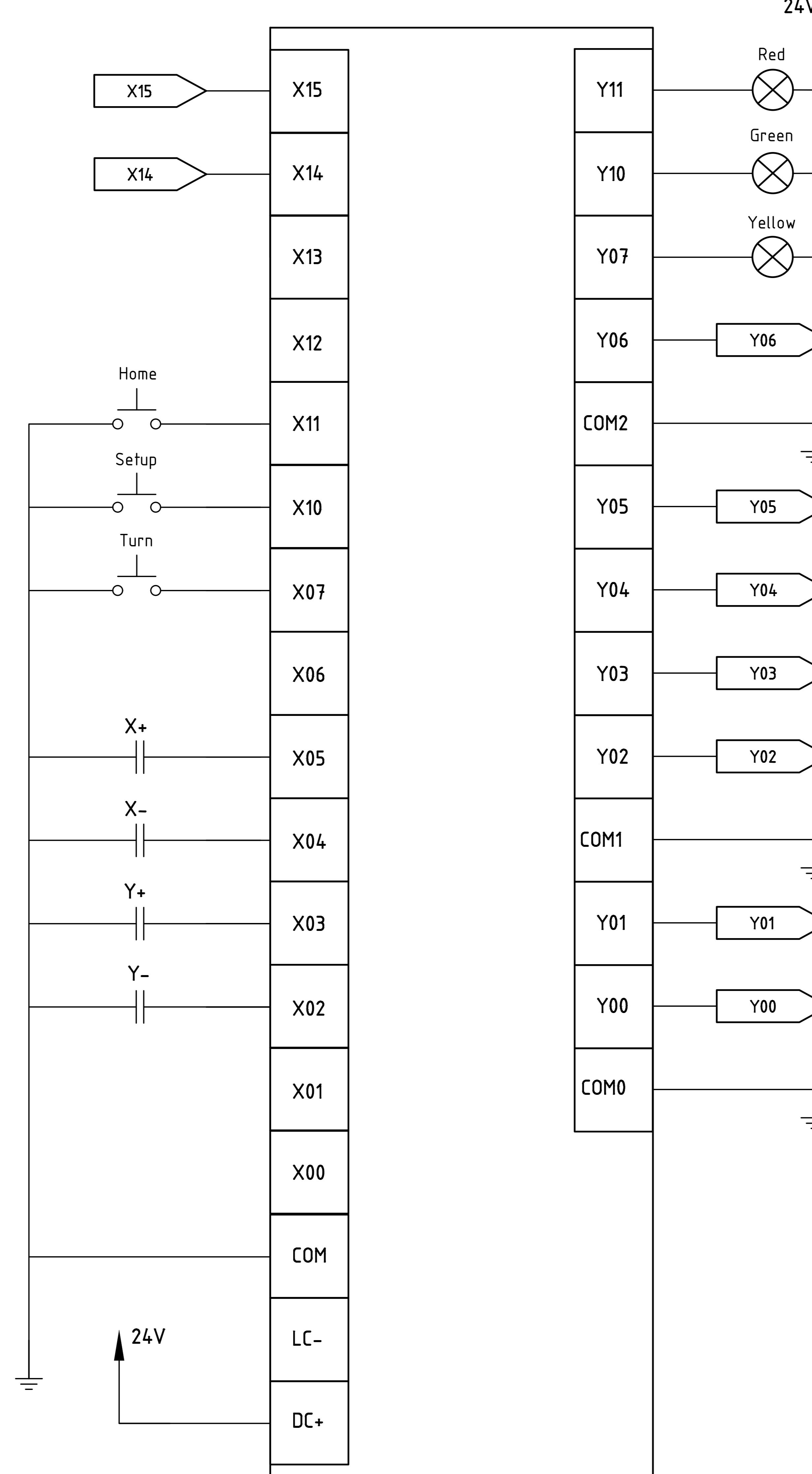
Cảm biến tiệm cận



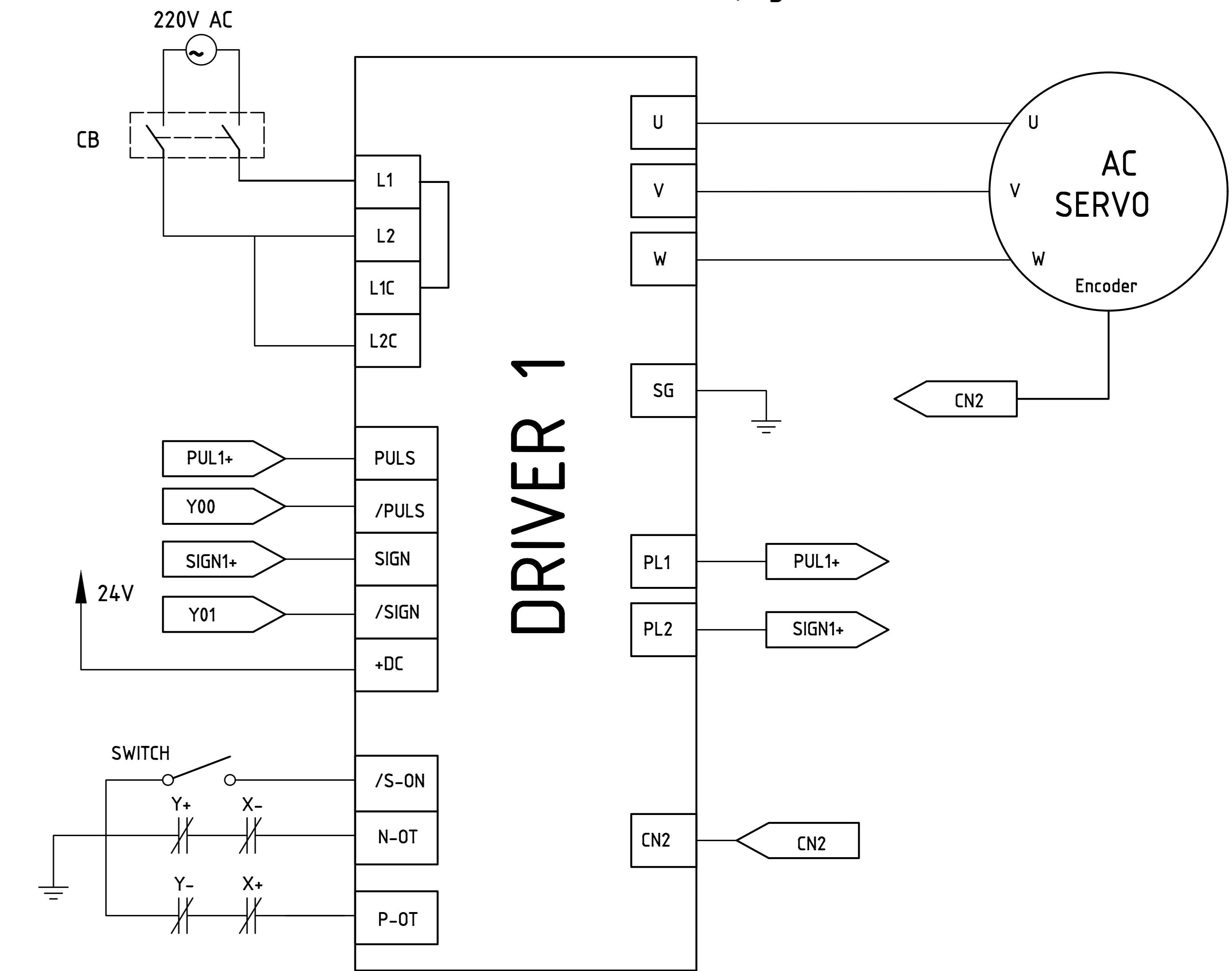
Nguồn



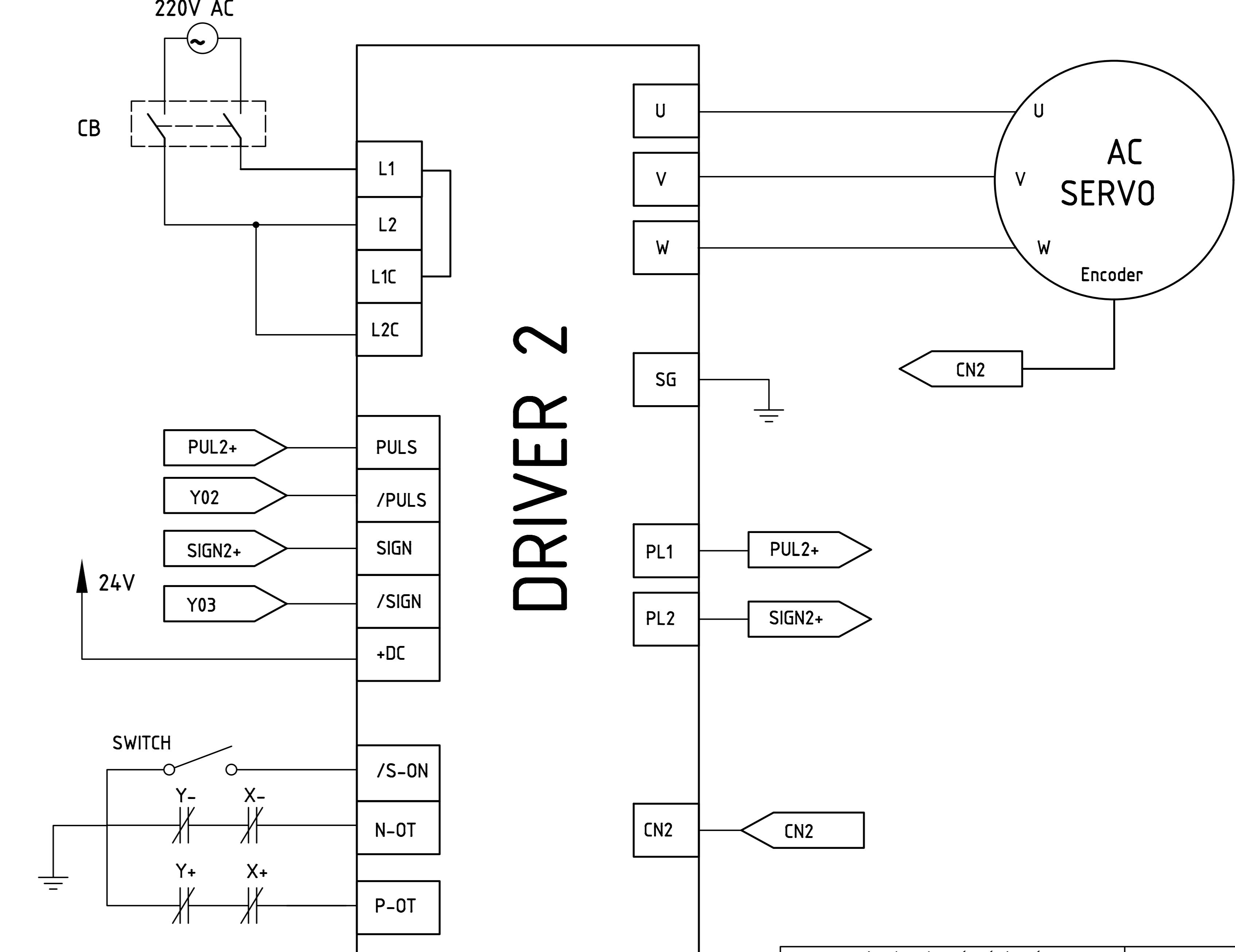
Bộ điều khiển PLC FX3U



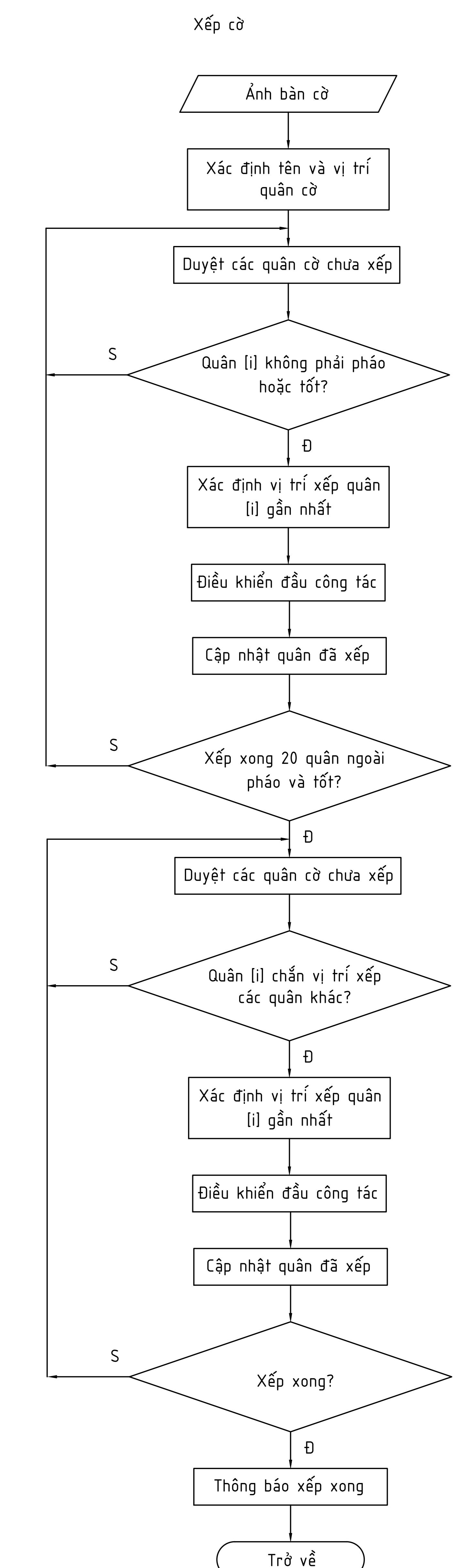
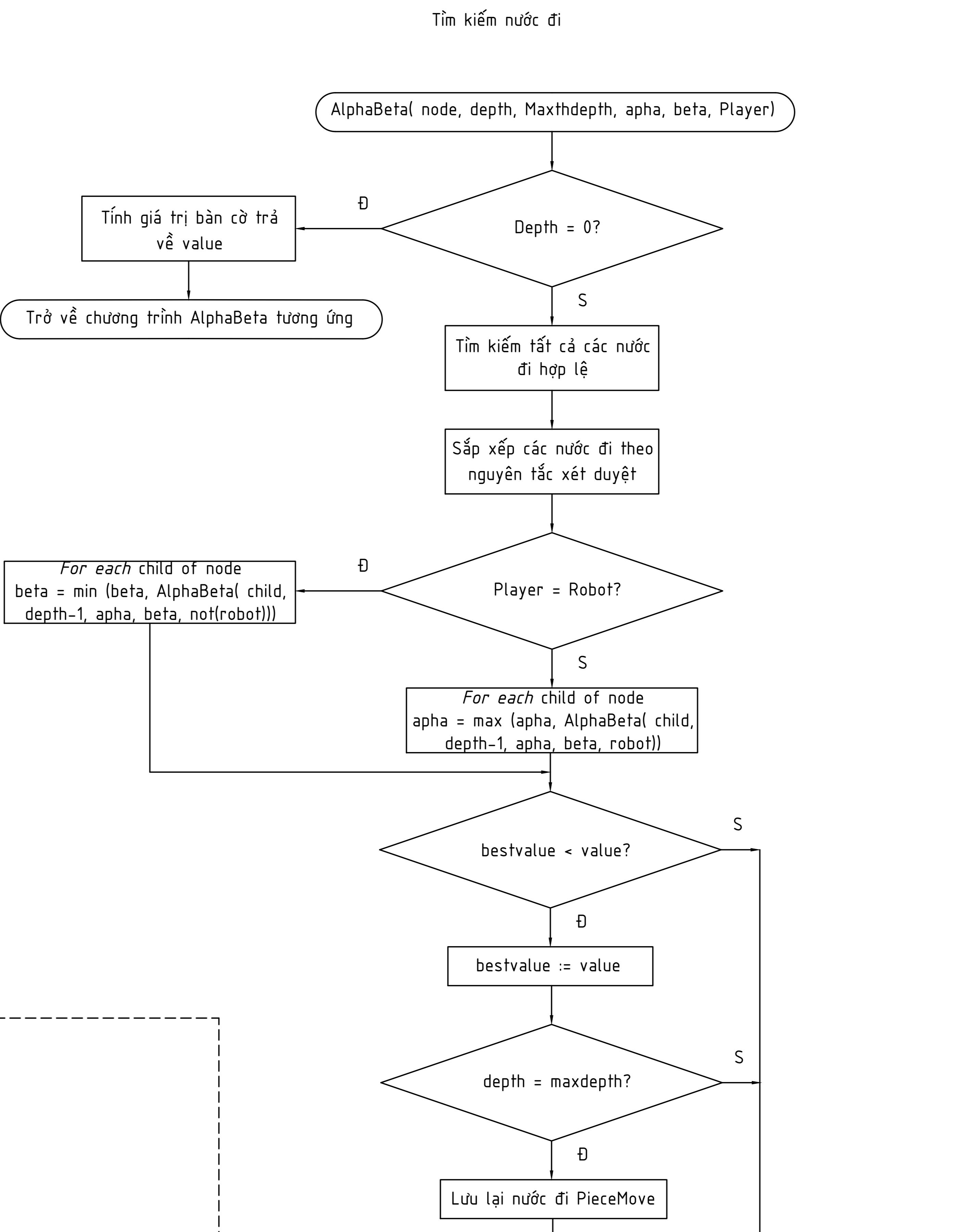
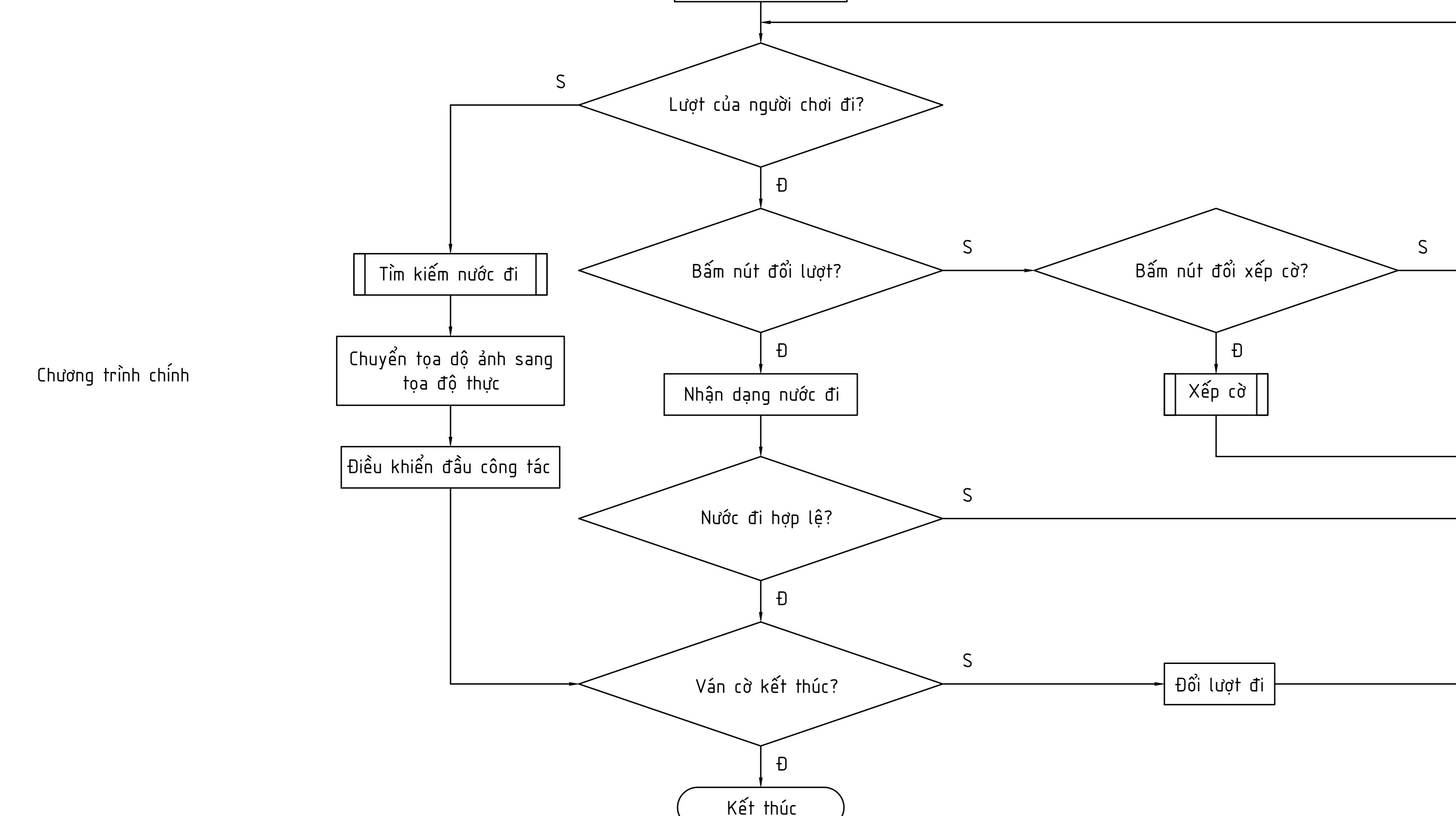
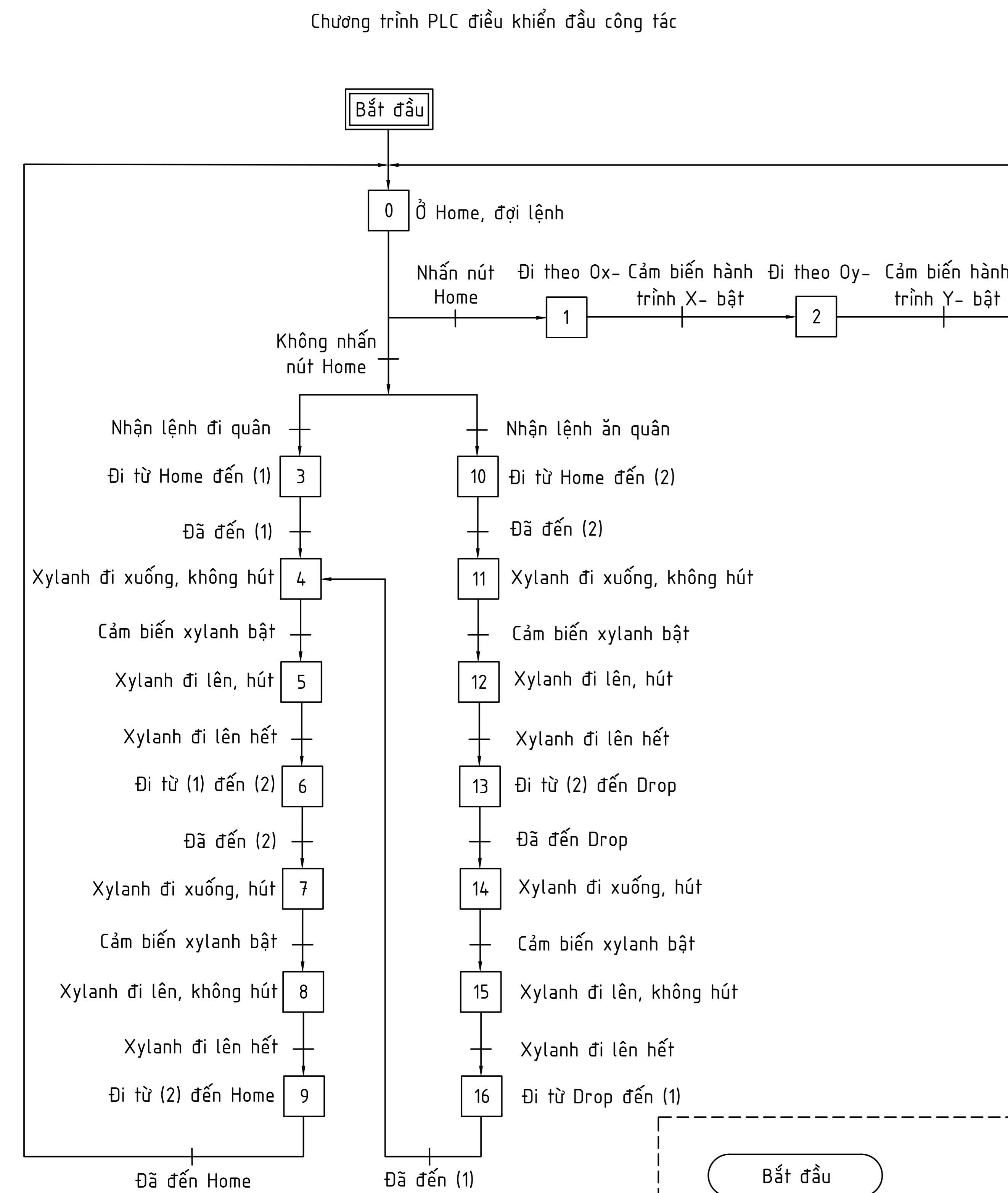
Driver 1 - Động cơ 1



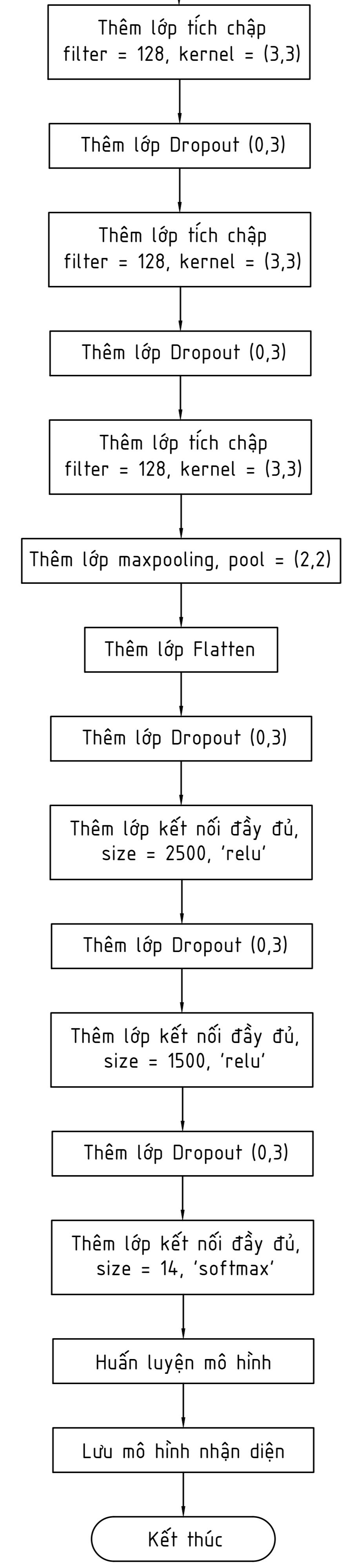
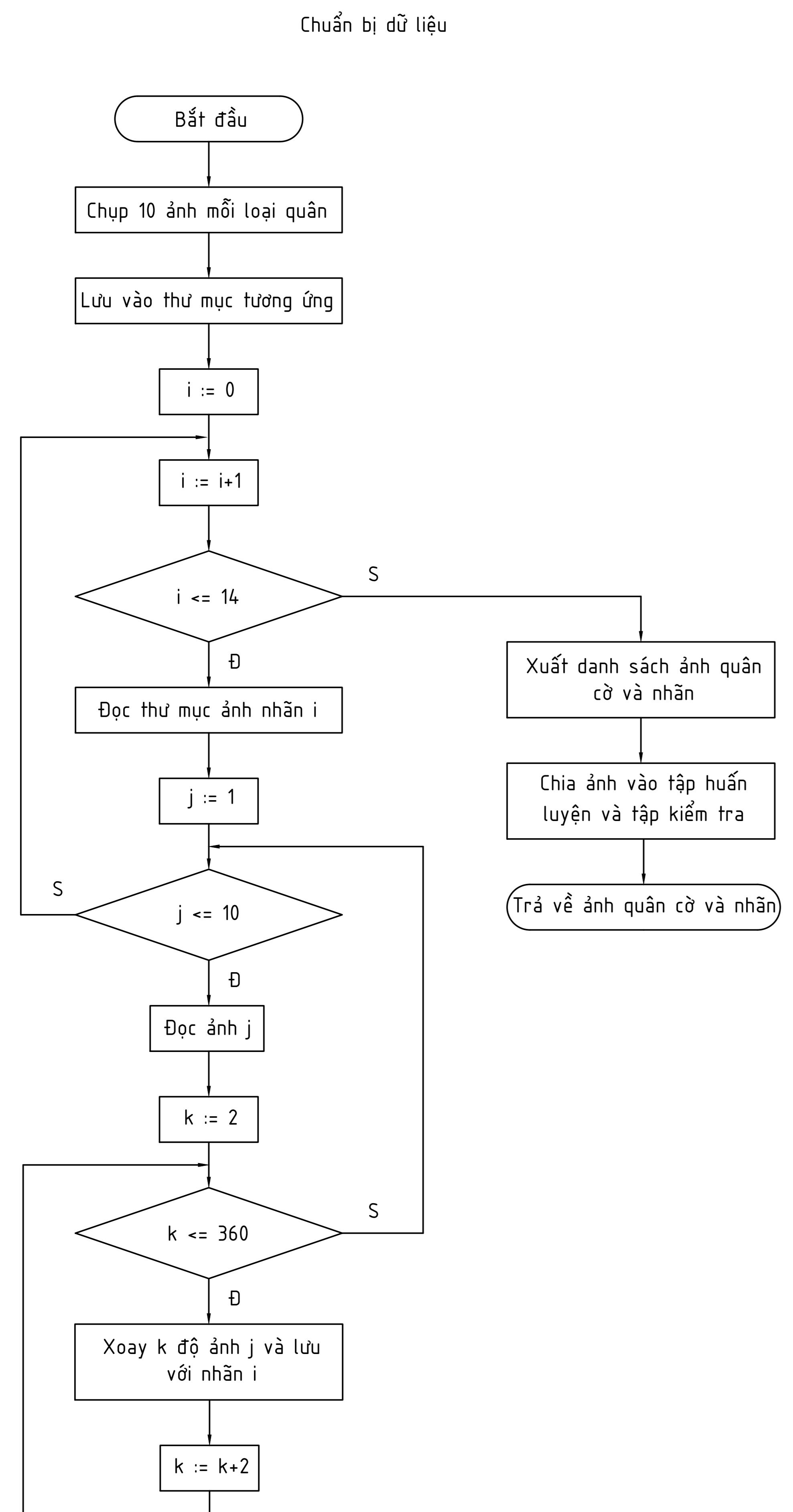
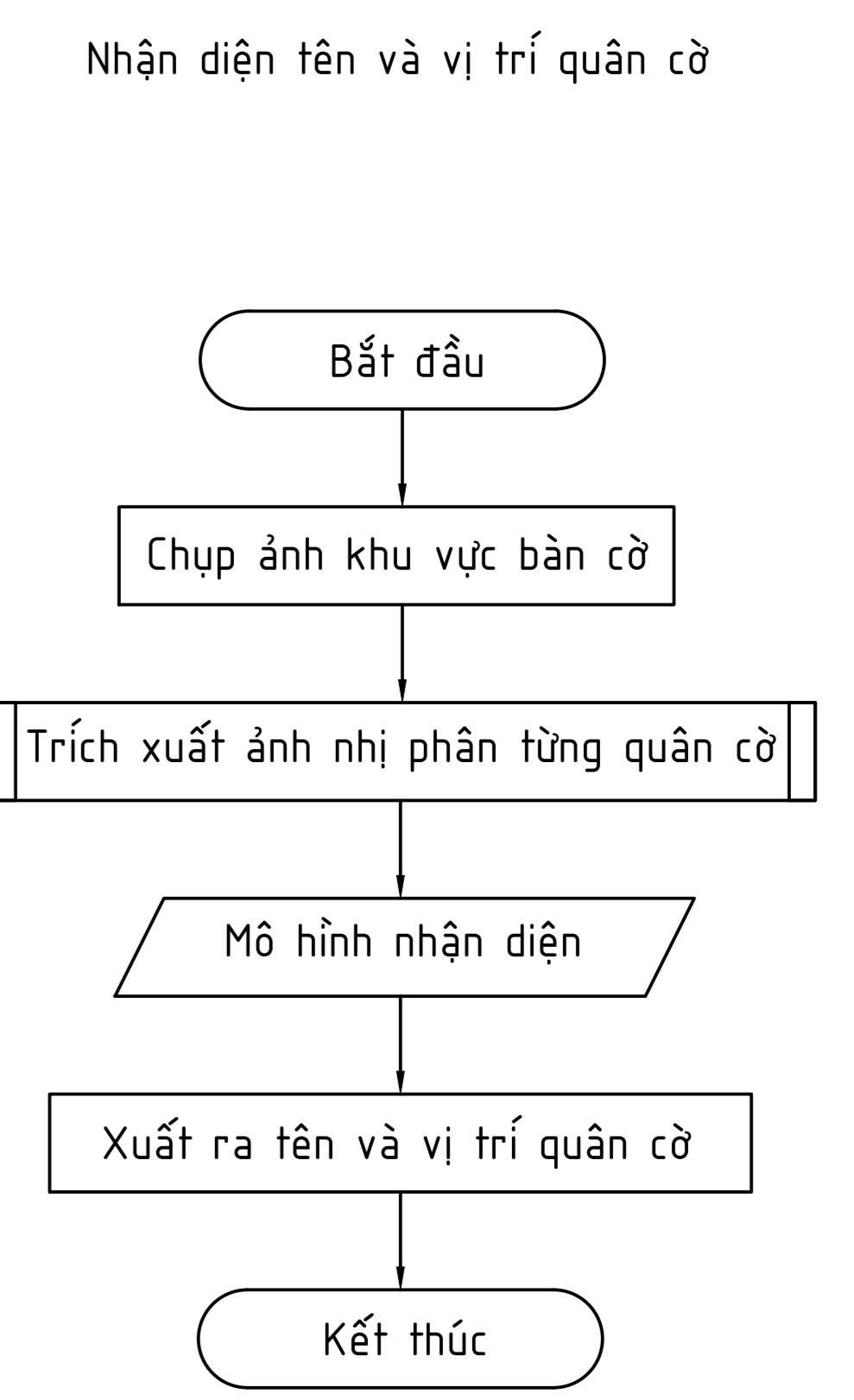
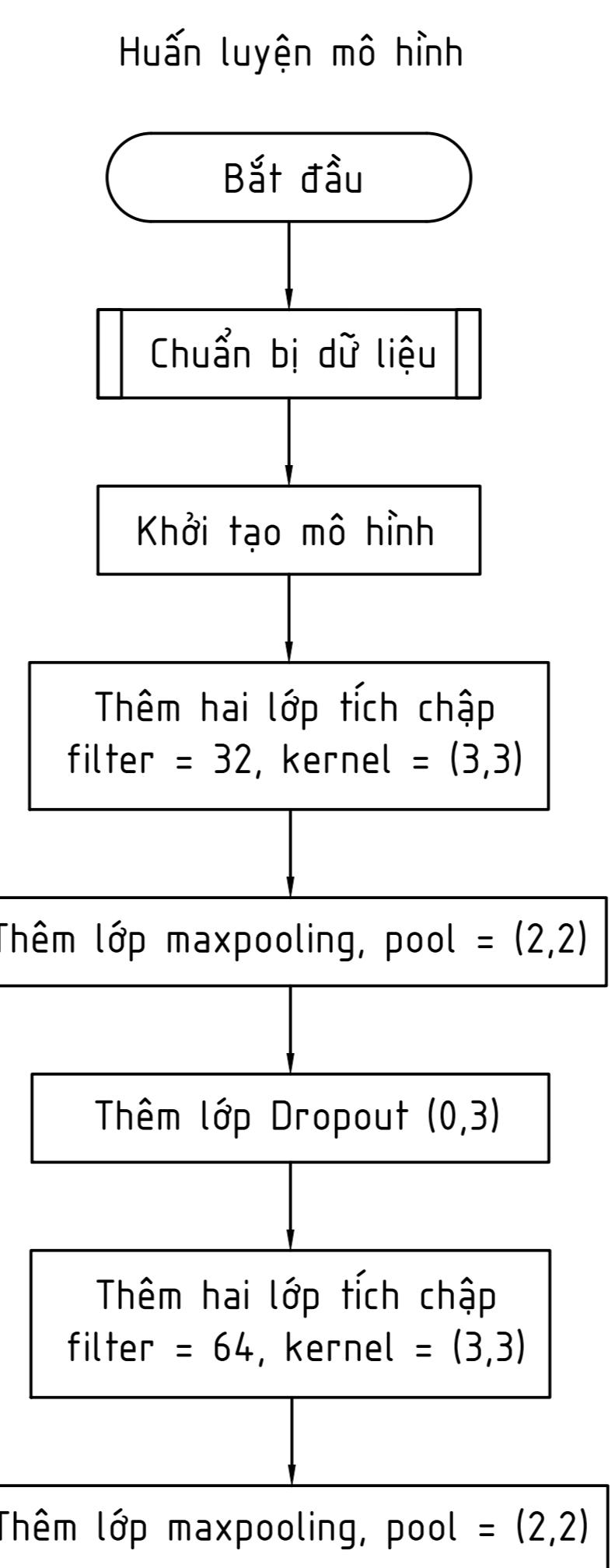
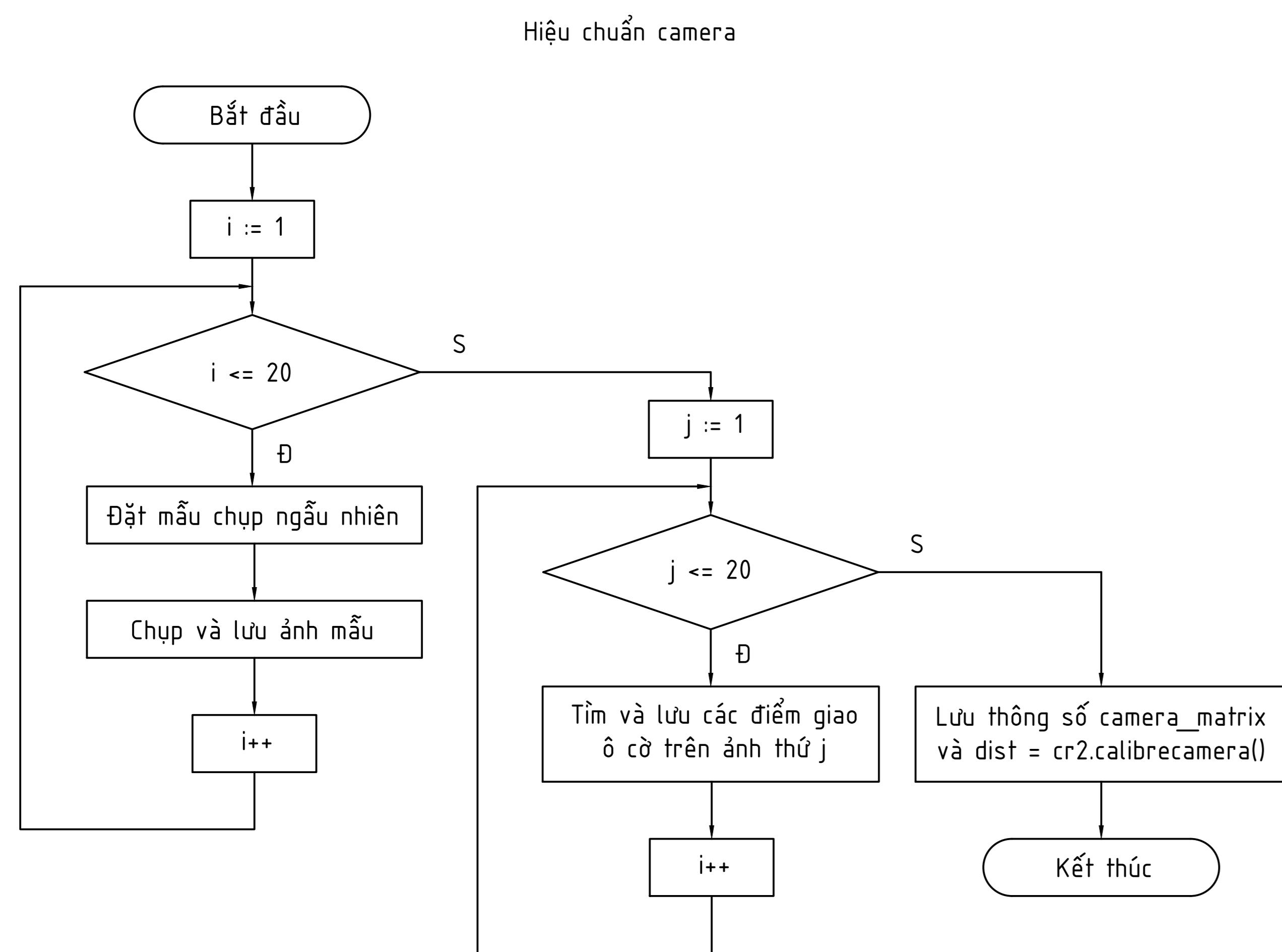
Driver 2 - Động cơ 2



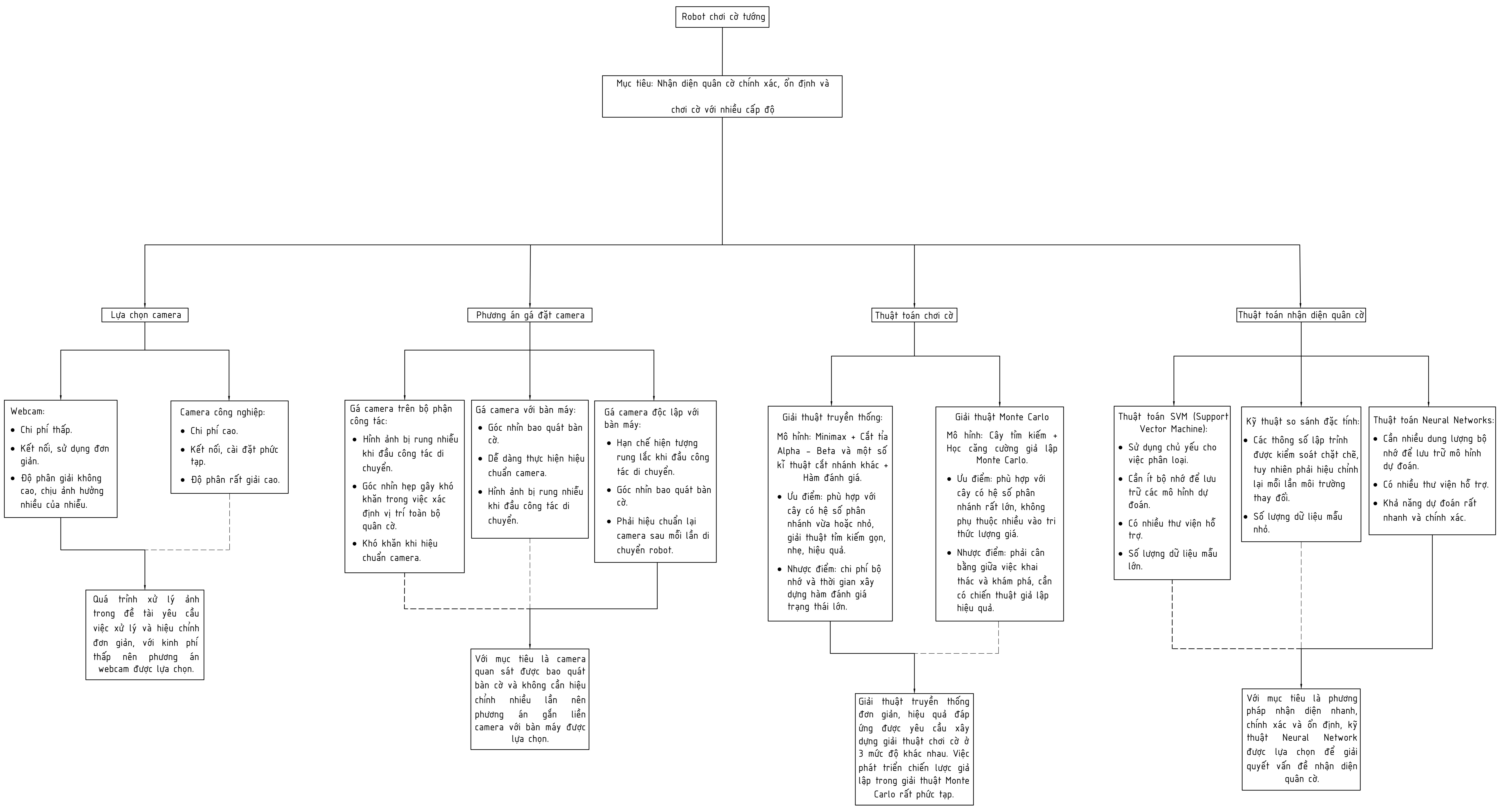
| PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG | | | | | LUẬN VĂN TỐT NGHIỆP | | |
|--------------------------------|-------------|--------|------|--|---------------------|----------|------------|
| Chức năng | Họ và tên | Chữ ký | Ngày | | Số Đồ ĐẦU DÂY | Số lượng | Khối lượng |
| Người vẽ | P. T. Thành | | | | | | |
| Hướng dẫn | N. D. Hồng | | | | | | |
| Duyệt | P. C. Bằng | | | | | | |
| | | | | | Tờ A0 | Số tờ 1 | |
| | | | | | | | |



| PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ROBOT CHƠI CỜ TƯỚNG | | | | LUẬN VĂN TỐT NGHIỆP | | |
|---|-------------|--------|------|----------------------------------|------------|-------|
| Chức năng | Họ và tên | Chữ ký | Ngày | Số lượng | Khối lượng | Tỷ lệ |
| Người vẽ | P. T. Thành | | | | | |
| | N. Q. Hưng | | | | | |
| Hướng dẫn | P. C. Bằng | | | Tờ: A0 | Số tờ: 1 | |
| Duyệt | P. C. Bằng | | | ĐẠI HỌC BÁCH KHOA TP.HỒ CHÍ MINH | | |
| | | | | | | |



| PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ROBOT CHƠI CỜ TƯỞNG | | | | LUẬN VĂN TỐT NGHIỆP | | |
|--|-------------|--------|------|---------------------|-----------|-------|
| Chức năng | Họ và tên | Chữ ký | Ngày | LƯU ĐỒ GIẢI THUẬT | XỬ LÝ ẢNH | Tỷ lệ |
| Người vẽ | P. T. Thành | | | | | |
| Hướng dẫn | P. C. Bằng | | | | | |
| Duyệt | P. C. Bằng | | | | | |



| PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG ROBOT CHƠI CỜ TƯỚNG | | | | | LUẬN VĂN TỐT NGHIỆP | | |
|---|-------------|--------|------|--------------------|----------------------------------|------------|-------|
| Chức năng | Họ và tên | Chữ ký | Ngày | LỰA CHỌN PHƯƠNG ÁN | Số lượng | Khối lượng | Tỷ lệ |
| Người vẽ | P. T. Thành | | | | | | |
| | N. Q. Hưng | | | | | | |
| Hướng dẫn | P. C. Bằng | | | | Tờ: A0 | Số tờ: 1 | |
| Duyệt | P. C. Bằng | | | | ĐẠI HỌC BÁCH KHOA TP.HỒ CHÍ MINH | | |