



## **BÁO CÁO ROS XÂY DỰNG VÀ MÔ PHỎNG THIẾT LẬP ĐIỀU KHIỂN ROBOT**

Họ và tên Nguyễn Đào Đức Thắng

Mã sinh viên 22027509

## **I, Mục tiêu dự án**

### **Kiểm tra thiết kế robot**

- Xác nhận cấu trúc cơ khí của robot, bao gồm các liên kết, khớp và cơ chế truyền động.
- Đánh giá khả năng hoạt động của hệ thống điều khiển.

### **Phát triển thuật toán điều khiển**

- Kiểm tra thuật toán điều khiển phản hồi, PID, điều khiển quỹ đạo cho robot di chuyển.
- Tối ưu hóa thuật toán di chuyển cho robot bốn chân, robot có bánh xe hoặc robot bay.

### **Mô phỏng cảm biến và xử lý dữ liệu**

- Tích hợp và kiểm tra dữ liệu từ cảm biến LiDAR, camera, IMU, GPS, v.v.
- Xây dựng thuật toán xử lý ảnh, điều hướng, SLAM và tránh chướng ngại vật.

### **Mô phỏng tương tác vật lý**

- Mô phỏng động lực học của robot (khối lượng, trọng lực, ma sát, va chạm).
- Kiểm tra khả năng di chuyển trên các địa hình phức tạp.

### **Tăng tốc phát triển và giảm chi phí**

- Giảm thiểu rủi ro hư hỏng robot trong thử nghiệm thực tế.
- Tăng tốc quá trình phát triển bằng cách thử nghiệm nhiều kịch bản khác nhau mà không cần thay đổi phần cứng.

### **Kết hợp với ROS để kiểm tra hệ thống điều khiển**

- Kết hợp Gazebo với **ROS (Robot Operating System)** để thử nghiệm và tinh chỉnh hệ thống phần mềm điều khiển.
- Mô phỏng hệ thống phân tán, robot tự động và phối hợp nhóm robot.

## II, Động học robot

Robot bánh xích di chuyển dựa trên nguyên tắc vi sai:

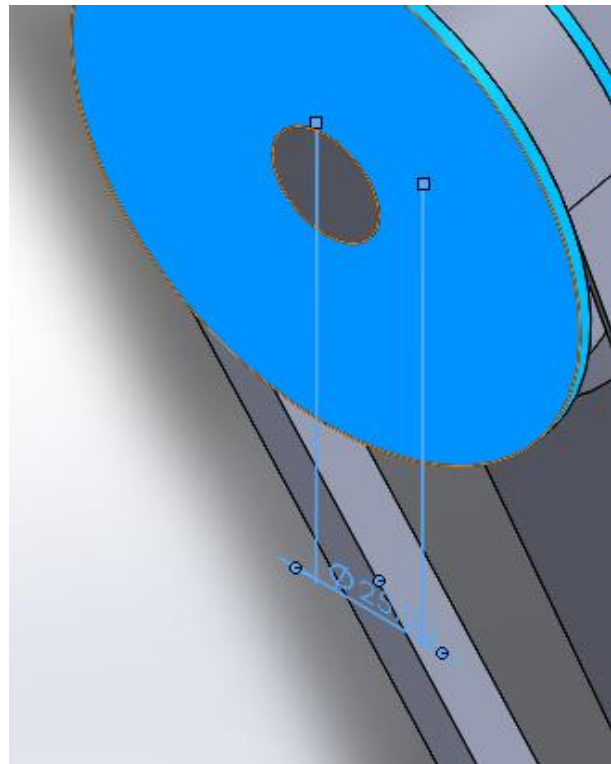
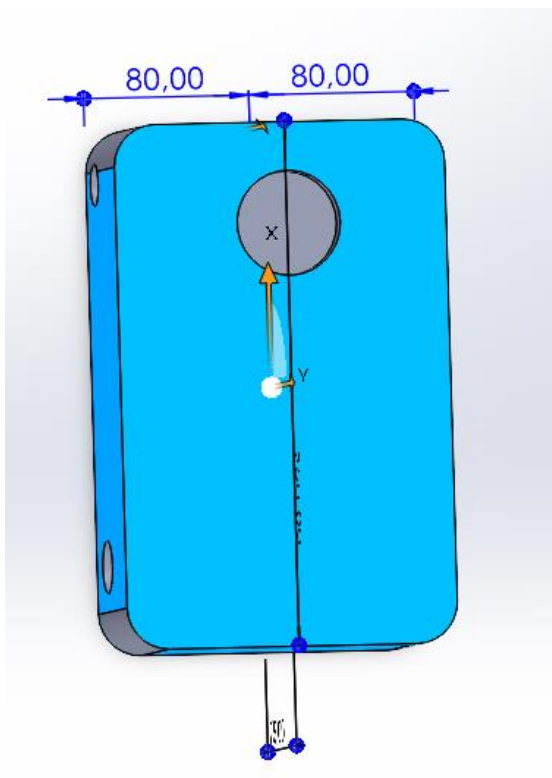
- Hai dải xích hoạt động độc lập, tốc độ khác nhau giúp robot quay trái/phải.
- Khi hai xích có cùng tốc độ, robot di chuyển thẳng.
- Khi một xích đứng yên và xích còn lại quay, robot quay tại chỗ.

Phương trình động học

$$v = \frac{v_L + v_R}{2}$$

$$\omega = \frac{v_R - v_L}{b}$$

Kích thước



## III Thiết kế robot ,cách đặt hệ trục

- Robot bánh xích với thiết kế nhỏ gọn, phù hợp cho các nhiệm vụ di chuyển trên địa hình phức tạp.
- Hệ thống bánh xích giúp robot có khả năng bám đường tốt, di chuyển ổn định trên bề mặt gồ ghề.
- Phần thân có dạng hộp chữ nhật với các góc bo tròn, giúp giảm lực cản khi di chuyển.

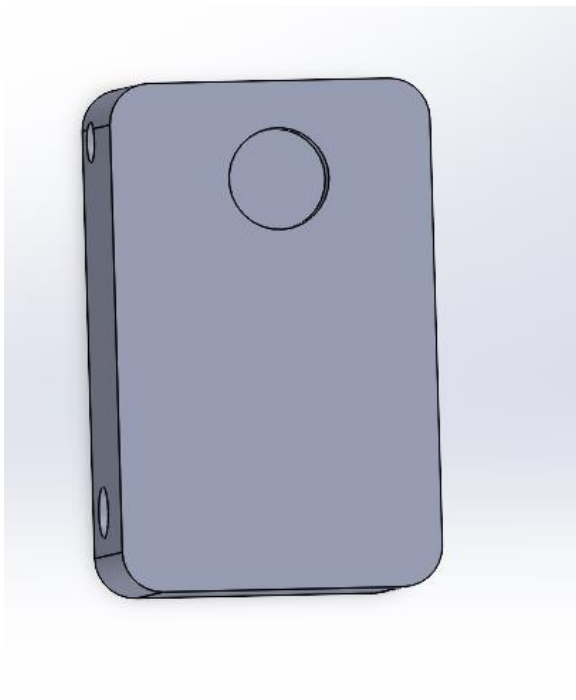
### **Cấu Trúc Chính**

- **Khung gầm:**
  - Thiết kế dạng hộp kín có thể chứa linh kiện điện tử, động cơ và nguồn cấp điện bên trong.
- **Hệ thống bánh xích:**
  - Bao gồm hai bánh chủ động lớn ở hai bên và một số bánh phụ để dẫn hướng xích.
  - Xích bao quanh các bánh xe giúp robot di chuyển êm ái và ổn định.

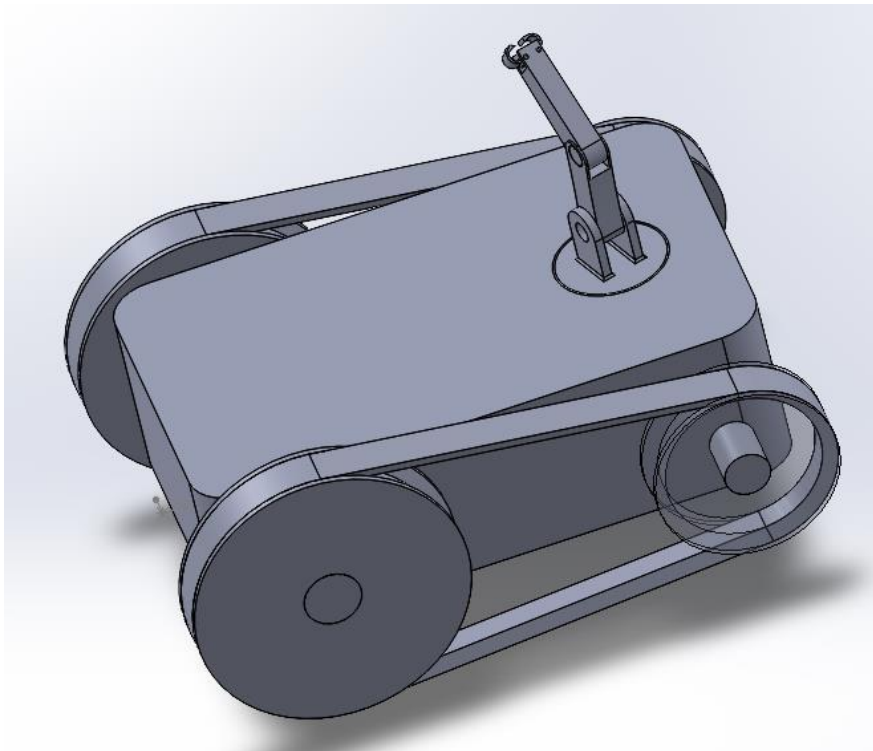
### **Cánh tay hoặc cơ cấu nâng phía trên:**

Có một tay gắp hoặc cơ cấu nâng nhỏ gắn trên phần thân, có thể xoay quanh một trục cố định.

### **Thiết kế trên solid**



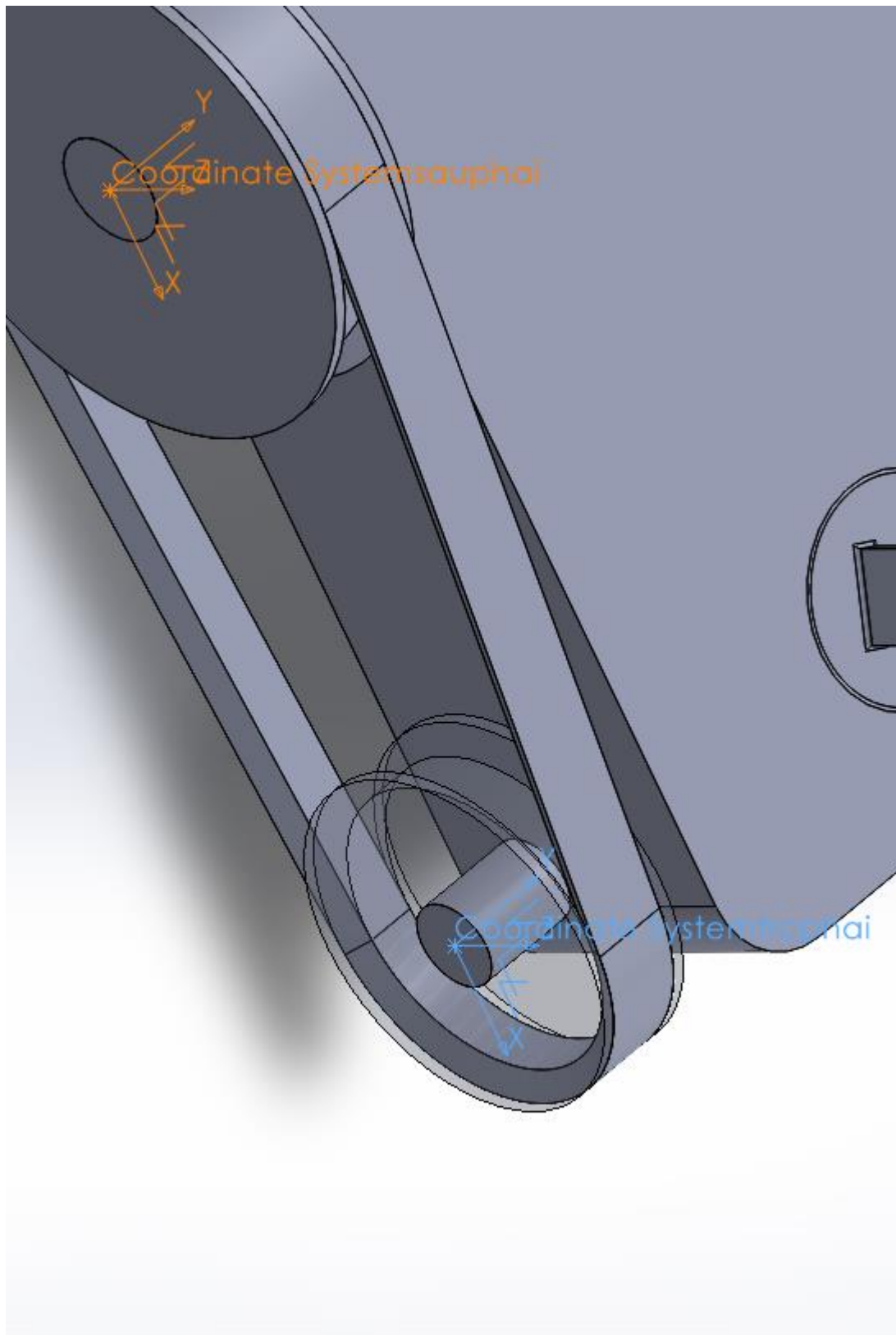
Thân bo tròn giảm lực cản khi di chuyển



Xe hoàn chỉnh

Đặt hệ trục tọa độ

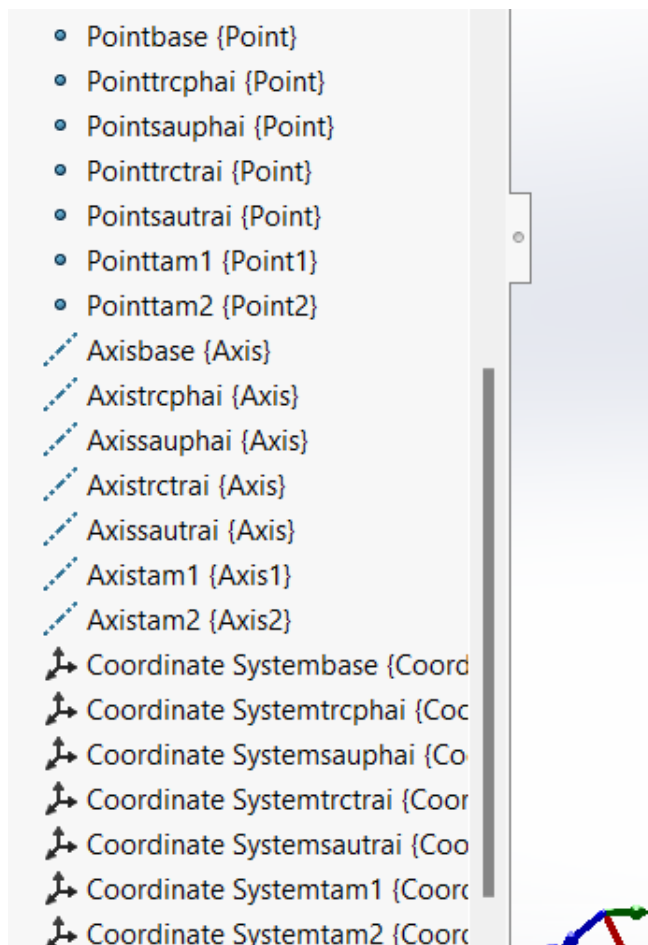




**Trục X: Hướng về phía trước hoặc sau robot**

**Trục Y: Hướng sang trái hoặc phải của robot**

**Trục Z: Hướng lên hoặc xuống**



Sau khi xuất URDF thành công



## VI Mô tả file URDF

### Cấu trúc robot giả định

- **base\_footprint**: Liên kết cơ sở, đại diện cho vị trí của robot trên mặt đất (thường là một liên kết "ảo" không có hình học).
- **base\_link**: Thân chính của robot, gắn với base\_footprint qua một biến đổi tĩnh (static transform).
- **link1**: Phần đầu tiên của cánh tay robot, gắn với base\_link qua một khớp quay (link1\_joint).
- **link2**: Phần thứ hai của cánh tay robot, gắn với link1 qua một khớp quay (link2\_joint).
- **base\_disk** (dựa trên hình ảnh): Một chi tiết hình đĩa, có thể là một phần của base\_link hoặc một liên kết khác, gắn qua một khớp quay.

### Links

- base\_footprint: Liên kết ảo, gốc tọa độ.
- base\_link: Thân chính, gắn với base\_footprint qua khớp cố định.
- link1: Phần đầu cánh tay, gắn với base\_link qua khớp quay (link1\_joint).
- link2: Phần thứ hai cánh tay, gắn với link1 qua khớp quay (link2\_joint).
- base\_disk (dựa trên hình): Chi tiết hình đĩa, có thể gắn với base\_link qua khớp quay.

### Joint

- Khớp cố định: base\_footprint  $\rightarrow$  base\_link.
- Khớp quay: base\_link  $\rightarrow$  link1 (điều khiển bởi link1\_joint\_controller), link1  $\rightarrow$  link2 (điều khiển bởi link2\_joint\_controller).

## V Mô tả cơ chế điều khiển trên gazebo

### 1. Điều khiển di chuyển robot

- **Topic**: /cmd\_vel (kiểu Twist).
- **Phím điều khiển**:
  - w: Tăng tốc độ tiến (linear.x tăng, tối đa 150 m/s).
  - x: Giảm tốc độ lùi (linear.x giảm).
  - a: Rẽ trái (angular.z tăng, tối đa 150 rad/s).
  - d: Rẽ phải (angular.z giảm).
  - s: Dừng (đặt linear.x và angular.z về 0).
- **Cơ chế**: Lệnh Twist được gửi qua self.cmd\_pub.publish(self.twist), Gazebo nhận và áp dụng để di chuyển robot (thường là base\_link hoặc bánh xe).



## 2. Điều khiển tay máy (2 khớp)

- **Topic:** /link1\_joint\_controller/command và /link2\_joint\_controller/command (kiểu Float64).
- **Phím điều khiển:**
  - i: Tăng tốc độ cả hai khớp (link1\_vel và link2\_vel tăng, tối đa 150 rad/s).
  - k: Giảm tốc độ cả hai khớp (link1\_vel và link2\_vel giảm).
  - j: Quay trái (giảm tốc độ khớp).
  - l: Quay phải (tăng tốc độ khớp).
- **Cơ chế:** Lệnh Float64 được gửi qua self.link1\_pub.publish(self.link1\_vel) và self.link2\_pub.publish(self.link2\_vel), Gazebo nhận và điều khiển khớp link1 và link2.

## 3. Tích hợp với Gazebo

- File launch đã khởi động Gazebo, tải mô hình robot (robo.urdf), và thiết lập các bộ điều khiển (link1\_joint\_controller, link2\_joint\_controller).
- File Python gửi lệnh qua các topic, Gazebo nhận và áp dụng để mô phỏng chuyển động của robot.

## 4. Thoát chương trình

- Nhấn Ctrl+C để thoát, robot dừng (tốc độ về 0).
-