DESIGN AND CONTROL OF A FETCHING QUADRUPED

Individual Progress of Jitao Li

By

Jitao Li

Individual Progress for ECE 445, Senior Design, Fall 2025

TA: Xuekun Zhang

April 17, 2025

Project No. 3

Abstract

This individual progress report details my contributions to our quadruped fetching robot project, focusing specifically on linking arm motion with visual input. My primary responsibility involves implementing coordinate transformations between multiple reference frames: the camera coordinate system, the robot dog's base frame, and the robotic arm's coordinate system. This critical component enables the integration of visual perception with physical manipulation by converting detected object positions from camera space to actionable coordinates for the robotic arm. I developed and implemented both the camera-to-dog (T_cam2dog) and dog-to-arm (T_dog2arm) transformation matrices, as well as the inverse kinematics solutions necessary for precise arm positioning. The transformation component serves as the bridge between the vision system and mechanical manipulation, enabling the robot to accurately locate and interact with target objects in its environment. Current progress includes successful implementation of the coordinate transformation and inverse kinematics algorithms, while ongoing work focuses on the transformation accuracy and integrating the module with other subsystems.

Contents

1. Introduction

1.1 Team Project Overview

Our team is developing an integrated robotic system that combines a commercial quadruped robot platform (Unitree Go2) with a custom-designed robotic arm for autonomous object fetching capabilities. The project addresses a significant limitation in current commercial robotic dog platforms - their inability to manipulate objects despite their advanced mobility features. To achieve this integration, our system employs five primary modules: a Robot Dog Control Unit for platform mobility, a Robot Dog Vision Module utilizing YOLOv8n for object detection, a Transformation Module for coordinate system conversion, a Robot Arm Design Module for mechanical manipulation, and a Robot Arm Control Module for precise motor control. The complete system demonstrates how the combination of mobility and manipulation can extend the utility of robotic dogs for practical applications in environments requiring object retrieval, while maintaining the platform's inherent mobility and stability characteristics.

1.2 Individual Responsibilities and Role in the Greater Project

As team leader, I am primarily responsible for the development of Transformation Module. My role focuses on creating the crucial bridge between the system's perception and action capabilities. My primary responsibilities encompass developing the coordinate transformation system between three key reference frames: the camera coordinate system (from D435i RGBD camera), the robot dog's base coordinate system, and the robotic arm's base coordinate system. This involves implementing and calibrating two essential transformation matrices (T_cam2dog and T_dog2arm), as well as developing the inverse kinematics solution for the 5-DOF robotic arm. The transformation module I develop enables the system to accurately convert detected object positions from camera space into actionable coordinates for the robotic arm, while handling challenges such as singularities and joint limits. Through close collaboration with both the vision and arm control teams, my work ensures smooth data flow between subsystems and directly enables the system to translate visual information into precise physical actions, making it a critical component in achieving the project's object manipulation goals.

2 Individual Design Work

2.1 Design Consideration

2.1.1 Camera Selection

I considered adding a separate high-quality camera to expand view range and improve image quality. However, this addition would have created several problems: it would have made the robot dog's load heavier by adding over 200g, reduced battery runtime by 10%, and required complex mounting hardware and software to sync two video feeds. Instead, we kept the built-in RGB-D camera (Intel D435i [1]), which already provides synchronized color and depth video of 30 frames per second at most with minimal delay. This camera works seamlessly with the robot's processing unit and needs no extra setup. It delivers depth output stream of decent quality (1280*720 resolution), has an FOV of 69*42, and has a range of .3m to 3m, which are all ideal for indoor testing. This simpler approach avoided unnecessary complications in both hardware and software.

2.1.2 Motor Selection

In the transformation module implementation, precise motor control and accurate position feedback are crucial for achieving reliable coordinate transformations and accurate object manipulation. We selected the DM4310 [2] brushless DC servo motors and RM2006 [3] motor, which feature integrated encoders providing real-time angular position feedback with high resolution and can maintain commanded positions with minimal error (within 1°). These motors support high-frequency CAN bus communication (⩾1kHz update rate), enabling rapid position updates based on visual feedback, which is essential for real-time adjustment of the arm's configuration when the vision system detects changes in target object position or when small corrections are needed during the approach and grasp phases. Moreover, the selected motors provide sufficient torque (up to 1.0 N·m for RM2006) to maintain stable positions under external loads while offering precise position control, and the integrated FOC (Field-Oriented Control) in the DM4310 ensures smooth motion control and position holding. These comprehensive motor characteristics directly support the implementation of our transformation matrices (T_cam2dog and T_dog2arm) and inverse kinematics solutions, enabling precise and reliable coordinate transformations from visual detection to physical manipulation, thereby ensuring accurate object grasping and manipulation capabilities in our integrated system.

2.2 Diagrams

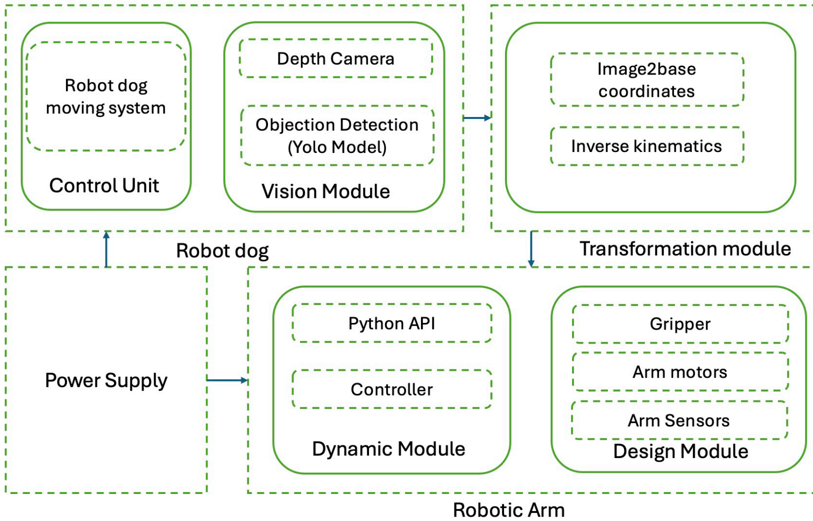

Figure 1. The entire project block diagram

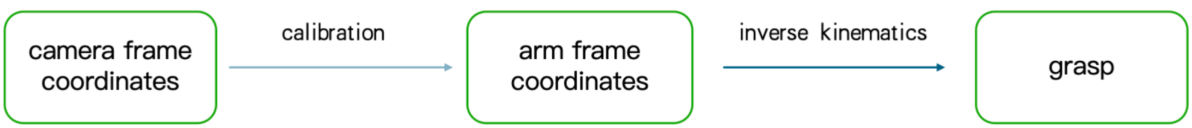

Figure 2. Detailed diagram for the role of the transformation component

2.3 Testing/Verification

2.3.1 Camera Calibration

The verification of the transformation from the camera frame to the arm frame involves eye-to-hand calibration. This calibration process involves two main steps: intrinsic camera calibration and extrinsic calibration. For intrinsic calibration, I use a standard 8×6 checkerboard pattern with 30mm square size, capturing approximately 30 images of the pattern from different viewpoints to determine the camera's internal parameters including focal length, principal point, and distortion coefficients. For extrinsic calibration, I place the checkerboard at known positions relative to the robot arm's base frame and collect paired data of the checkerboard corners' positions in both camera coordinates (obtained through corner detection) and arm base coordinates (known from the placement). Using these correspondences, I compute the rigid body transformation matrix between the camera frame and arm base frame through a least-squares optimization method that minimizes the reprojection error. The resulting transformation matrix is then decomposed into rotation and translation components, allowing us to convert any point detected in the camera frame to its corresponding position in the arm base frame. To validate the calibration accuracy, I measure the transformation error by placing test objects at known positions and comparing their transformed coordinates with ground truth measurements, aiming to achieve an average position error of less than 5mm across the working space, which is sufficient for our object grasping tasks.

2.3.2 Forward Kinematics

The forward kinematics of our 5-DOF (including the gripper) robotic arm is established using the Denavit-Hartenberg (DH) parameters to systematically describe the geometric relationships between consecutive joint frames. For each joint i, I defined four DH parameters: the link length ai, the link twist αi, the joint offset di, and the joint angle θi. Based on these parameters, I constructed individual homogeneous transformation matrices between adjacent joints using the standard DH transformation formula: Ti-1,i = Rot(z,θi)Trans(z,di)Trans(x,ai)Rot(x,αi). The complete forward kinematics transformation from the base frame to the end-effector, T0,end, is then obtained by multiplying these individual transformation matrices sequentially: T0,end = T0,1 × T1,2 × T2,3 × T3,4 × T4,5. This transformation matrix maps each set of joint angles [θ1, θ2, θ3, θ4] to the corresponding end-effector pose, represented as a 4×4 matrix containing both position and orientation information. The resulting transformation enables us to accurately track the end-effector's position and orientation in the base frame coordinates, which is essential for coordinating the arm's movements with the visual feedback system during object manipulation tasks.

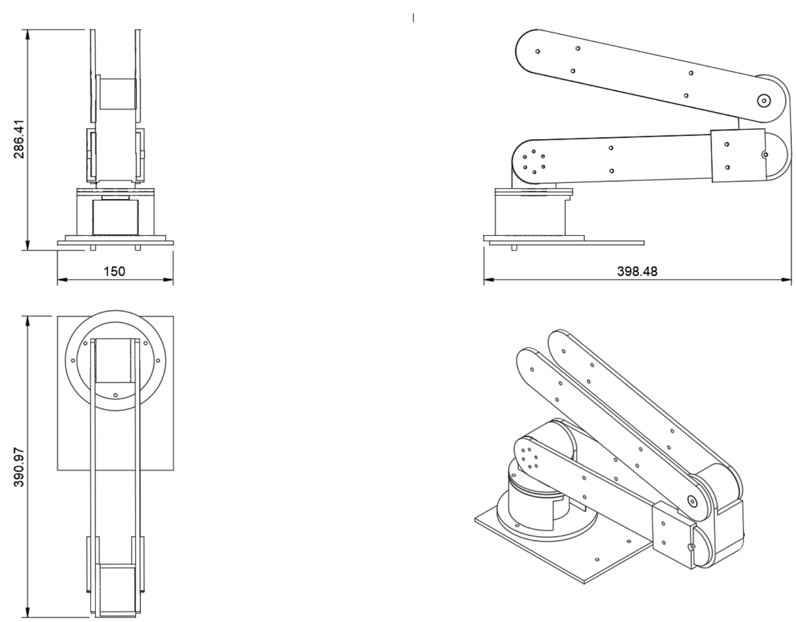

Figure 3: Design of our robot arm

2.3.3 Inverse Kinematics

For this part, I make sure that knowing the placement of the end point of the robot arm and its orientation, I can obtain a viable set of joint angles that enables the arm to move to that position. Denote the length of the first arm link as l1, second arm link as l2, length of the end effector as l3, base height as baseheight. Also the rotational angle of the base as θ1, angle of the shoulder as θ2, angle of the elbow as θ3, and finally rotational angle of the end effector relative to the second arm link as θ4.

Let (x,y,z) be target coordinate of the wrist in space. We can view the arm as rotating around the z axis in its base frame. As a result, when projected onto the x-y plane, we can see that

$$\theta_1 = \arctan(y, x)$$

Let the projected distance from the base center to wrist on the x-y plane be r,

$$r = \sqrt{x^2 + y^2}$$

And s is the wrist height,

$$s = z - base\mathrm{e}height$$

Then the distance from shoulder (at base height) to wrist is

$$d = \sqrt{r^2 + s^2}$$

Using cosine theorem, the angle of the elbow is

$$\cos(\theta_3) = \frac{r^2 + s^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

There are two solutions, corresponding to elbow-up and elbow-down configurations, in which we choose elbow-up as the default:

$$\begin{aligned} \theta_{3,\,\mathrm{up}} &= \arccos(\cos\theta_3) \\ \theta_{3,\,\mathrm{down}} &= -\arccos(\cos\theta_3) \end{aligned}$$

Continue to solve the triangle constituted by l1,l2 and d in the l1-l2 plane, we can get the shoulder angle

$$\theta_2 = \arctan\ 2(s, r) - \arctan\ 2\big(l_2 \sin(\theta_3), l_1 + l_2 \cos(\theta_3)\big)$$

And θ4 is the orientation of the end effector (gripper).

Simulation with target position at (200,150,200) and target orientation at (0,0,1):

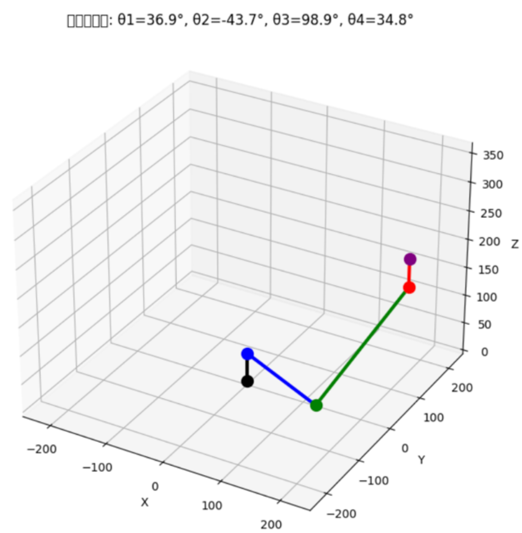

Figure 4: Simulation result of inverse kinematics

3. Conclusion

3.1 Self-assessment

Throughout the project, I am in charge of the development and implementation of the transformation from vision to motion, bridging the robot dog and arm components of the project. I achieved the primary objective of enabling coordinate conversions between the camera, robot dog, and robotic arm systems, while working towards meeting our target accuracy requirement of under 5mm. Moreover, I successfully implemented the inverse kinematics algorithm for the robotic arm. There were several challenges and areas for potential improvement. The calibration process currently requires manual placement of the checkerboard pattern, which could be automated to improve efficiency. Additionally, dynamic scenarios with dog movement introduce additional errors that could be addressed through more sophisticated real-time compensation algorithms. Looking ahead, implementing an automated calibration procedure and incorporating dynamic error compensation would significantly enhance the system's robustness and ease of deployment. Overall, the transformation module will successfully enable the integrated system to perform accurate object manipulation tasks, contributing significantly to the project's overall functionality and demonstrating the effectiveness of our approach to combining mobility and manipulation capabilities.

3.2 Plans for Remaining Work

My next phase of development will focus on comprehensive system validation and integration. First, I will implement and test the forward and inverse kinematics algorithms in the Mujoco simulation environment, which provides accurate physics simulation and visualization capabilities. This simulation testing will allow me to verify the accuracy of the transformation calculations and optimize the kinematic parameters without risking physical hardware. Additionally, I will integrate the robotic arm model with the quadruped dog model in Mujoco to simulate and validate the combined system dynamics, particularly focusing on how the dog's movement affects the arm's positioning accuracy. For real system integration, I will collaborate with the vision team to establish reliable communication protocols on Go2's development board for receiving object detection and positioning data, ensuring smooth data flow from visual perception to transformation calculations. Similarly, I will work with the hardware control team to implement efficient command interfaces, translating the calculated joint positions into proper motor control signals. The final stage will be comprehensive testing of the complete system with the team, starting with static grasping scenarios and progressively moving to more challenging cases involving the dog's motion.

References

[1] "Intel® RealSense™ Depth Camera D435i," Intel® RealSense™ Technology. https://www.intelrealsense.com/depth-camera-d435i/ (accessed Apr. 18, 2025).

[2] DaMiao TechnoBook. Retrieved from https://www.damiaokeji.com/index.php?c=show&id=84.

[3] DJI RoboMaster. RM C610 Brushless Motor Driver User Manual. Shenzhen, China: DJI, 2018.