DESIGN AND CONTROL OF A FETCHING QUADRUPED

By

Jitao Li

Teng Hou

Yikai Cao

Wenkang Li

Design Document for ECE 445, Senior Design, Spring 2025

TA: Xuekun Zhang

13 April 2025

Project No. 3

Abstract

Our project presents the design and integration of a lightweight robotic arm with a commercially available quadruped robotic dog to enable fetching skills. Despite the existence of various robotic dog platforms, the lack of manipulation capability has limited their functionality, particularly for tasks like fetching. Our approach integrates a manipulator with the quadruped using external sensing feedback, allowing the robotic system to autonomously retrieve objects. The robotic arm is designed to be compact and lightweight, ensuring minimal impact on the dog's mobility while enhancing its capability. The system demonstrates how the combination of mobility and manipulation can extend the utility of robotic dogs for practical applications in environments requiring object retrieval.

Contents

1. Introduction

1.1 Problem and Solution

Various commercial robotic platforms exist, but none demonstrate retrieval capabilities—primarily due to the absence of integrated manipulation systems. A compatible robotic arm for these platforms must satisfy critical criteria: minimal weight to maintain mobility, precision in object interaction, and structural durability. The combined dog-manipulator system will execute basic retrieval tasks through vision-guided operations. This integration necessitates custom manipulator engineering, seamless component synchronization, and specialized control architecture to achieve functional object manipulation capabilities.

We focus on developing an integrated robotic system that combines a mobile robot dog platform with a custom-designed manipulator arm. The system will perform object retrieval tasks through advanced visual recognition algorithms that enable precise grasping functionality. We are engineering a purpose-built robotic arm characterized by minimal weight impact, high positional accuracy, and operational resilience—all specifically optimized for compatibility with commercial quadrupedal robot platforms. A key technical challenge we address is establishing coordinated control between the locomotion system and the mounted manipulator. This integration directly addresses a significant capability gap in current commercial platforms, which typically lack object manipulation functionality despite their advanced mobility features.

1.2 Visual Aid

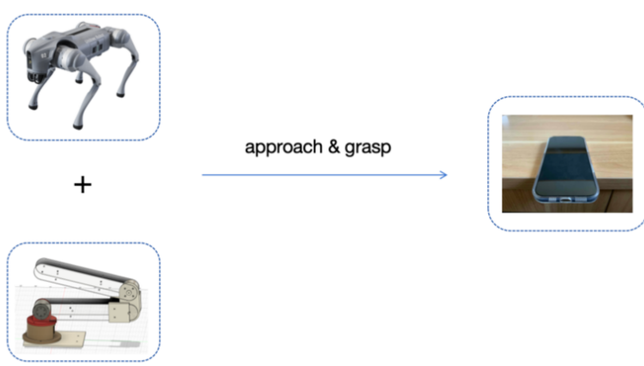


Figure 1: Pictorial representation of our project

1.3 High-level Requirements

To achieve the objectives of this project, the system must meet several key requirements. First, the robotic arm must be designed to be lightweight and compact enough to be mounted on the quadruped robot without compromising its functionality. Specifically, the arm should provide at least 5 degrees of freedom (DoF) to ensure it can perform necessary movements such as reaching, grasping, and lifting objects. The design must consider the impact on the robot's mobility and ensure that the arm does not hinder the robot dog's movement. Furthermore, the total cost of the robotic arm and integration system must remain under 1500 RMB to ensure the project is cost-effective while maintaining sufficient functionality.

Additionally, the system must be capable of identifying and tracking target objects in real time using visual feedback from external sensors, such as cameras. This will require a robust vision system capable of object recognition and tracking to guide the robotic arm's movements effectively. The integration of the arm with the quadruped robot must be seamless, with a coordinated control system ensuring that the movements of both the robot dog and the arm are synchronized. This coordination is essential to achieve precise and accurate object grasping, ensuring that the robot can successfully pick up and retrieve objects from its environment.

2. Design

2.1 Diagram

2.1.1 Block Diagram

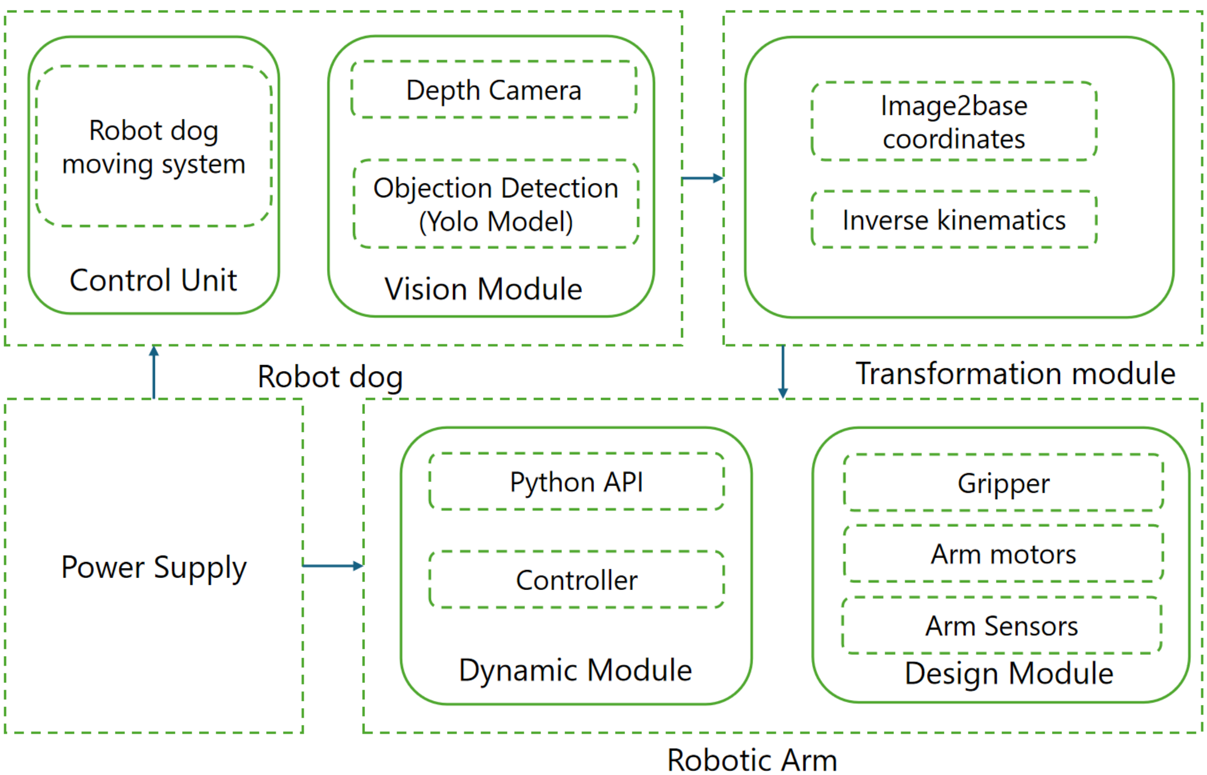


Figure 2: The block diagram

2.1.2 Physical Diagram

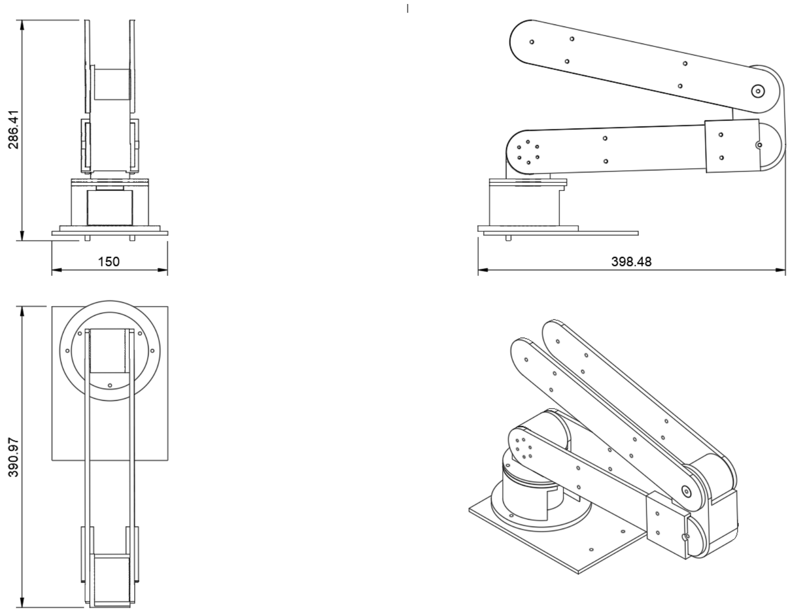


Figure 3: The engineering drawings of the robotic arm without gripping jaw.

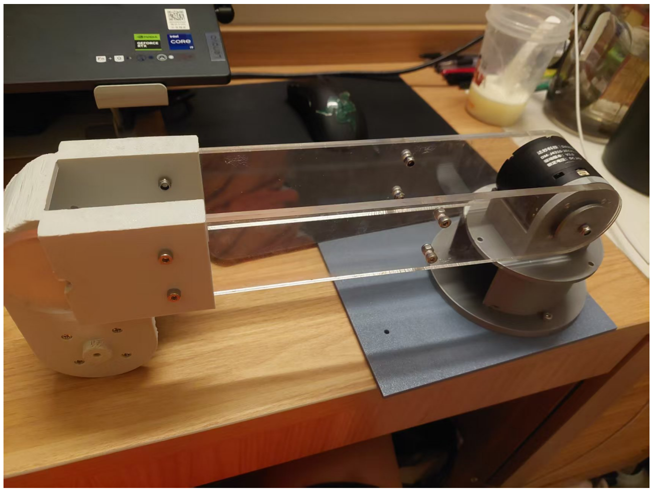


Figure 4: Custom-built robot arm

Our custom-built arm utilizes a series of connecting joints with mounting holes for servo motors or actuators, creating 5 degrees of freedom. Its lightweight, minimalist design suggests optimization for mounting on mobile platforms such as robot dogs, balancing reach capabilities with weight constraints. The mechanical structure includes reinforced pivot points and mounting provisions for a gripper at the terminal segment.

2.2 Subsystem Descriptions

2.2.1 Robot Dog Control Unit

The Robot Dog Control Unit is designed to autonomously maneuver the robotic dog toward any target object detected in its environment. This module leverages the Unitree Go2's built-in locomotion system to ensure smooth, reliable, and precise movement. Once the onboard object detection sensors identify a designated target, the control unit seamlessly processes the information through its advanced decision-making algorithms. It then commands the movement system to position the robotic dog directly in front of the target, ensuring optimal engagement for subsequent operations such as inspection, interaction, or data collection.

2.2.2 Robot Dog Vision Module

Dataset Collection:

The initial phase of our Vision Module is centered on dataset collection, which is pivotal for ensuring diversity and accuracy in subsequent tasks. We leveraged an open-source dataset comprising approximately two thousand images captured from mobile phones. This dataset was chosen not only for its accessibility but also for the diverse range of scenarios and lighting conditions it encapsulates, closely resembling real-world settings encountered by the robot. The images include various perspectives and object compositions, providing a robust foundation for object detection tasks. Careful pre-processing was performed to ensure consistency in resolution and quality across the dataset, and data augmentation techniques were employed to simulate different angles, contrasts, and occlusions. These steps significantly bolster the dataset's utility by addressing potential variations and ensuring that the model is exposed to a comprehensive set of conditions during training. Such rigorous dataset preparation is integral in reducing the gap between training and deployment environments, ultimately enhancing the generalization capability of the detection model when it is applied in real-time operations.

Model Training:

Once the dataset was curated, the next phase involved training the model using the YOLOv8n framework as our baseline. Building on the pre-trained YOLOv8n model, we fine-tuned the network over 100 training rounds with the mobile phone dataset. This focused training process was meticulously designed to optimize the model for high accuracy in object detection tasks. Throughout the training phase, extensive hyperparameter tuning and regular performance evaluations were conducted, leading to notable improvements in the model's detection capabilities. The model consistently achieved an mAP@0.50 [1] score of nearly 95, a remarkable milestone that underscores its proficiency in recognizing various objects with precision. The training process also included advanced data augmentation strategies to increase the resilience of the model to real-world variances. Validation on a separate test set confirmed that the model was not merely memorizing the training data but could adapt to unseen scenarios effectively. This robustness was crucial, as the trained model needed to operate reliably under the dynamic and challenging conditions it would encounter during field deployment. Below is the model prediction result:

Figure 5: Demonstration of our object detection model

Model Deployment on the Robotic Dog:

The final stage of the Vision Module involves deploying the trained YOLO model directly onto the robotic dog platform. Leveraging the robot's integrated high-definition camera, the system captures real-time image data, which is immediately processed by the YOLO model. This integration enables the robot to dynamically identify and locate target objects within its vicinity. The onboard processing unit is optimized to handle the computational requirements of running the YOLO model in real time without significant latency, ensuring that the detection process is both swift and accurate. During field tests, the deployed system consistently demonstrated its effectiveness by detecting various objects in diverse environmental settings, ranging from indoor corridors to outdoor spaces with variable lighting. The seamless integration of the vision system with the mechanical components of the robotic dog has created an agile and responsive detection system. This capability forms a critical part of the robot's operational framework, enabling it to autonomously navigate towards identified objects, thereby expanding its utility in complex, real-world deployment scenarios.

2.2.3 Transformation Module

Image to Base Coordinates

There are three key coordinate systems that need to be considered: coordinates in regards to on-board RGBD camera (D435i), coordinates with regard to dog, and coordinates with regard to arm's base. The entire process runs as follows: Compute the transformation matrix for camera-to-dog coordinate conversion T_cam2dog, and the transformation matrix for dog-to-arm conversion T_dog2arm. Denoting the object's coordinates in the arm's base space and camera space as C_arm and C_cam respectively, it follows that:

$$C_{arm} = T\_dog2arm \cdot T\_cam2dog \cdot C\_cam$$

To obtain the camera-to-dog transformation matrix (T_cam2dog), multiple calibration processes are required. First, we establish the robot dog's coordinate system by defining its origin and orientation axes. Next, we perform camera intrinsic calibration using a checkerboard pattern to determine the camera's internal parameters such as focal length and distortion coefficients. For the extrinsic calibration we utilize physical measurements, where the exact mounting position and orientation of the camera relative to the dog's reference frame are carefully determined using the specs of the robot dog. These measurements directly provide the translational and rotational components of the transformation matrix.

Obtaining the transformation matrix from the robot dog to the manipulator arm base (T_dog2arm) also requires precise calibration to establish the spatial relationship between these two coordinate systems. First we define the robot dog's coordinate system (centered at its body with x-axis forward, y-axis left, and z-axis up) and the manipulator base coordinate system (at the base of the arm with z-axis along the first joint axis). Since the manipulator is rigidly mounted on the dog, the transformation is again primarily determined by the physical mounting configuration. For calibration, our effective approach utilizes a precision measurement tool such as a 3D scanner to measure key reference points on both the dog's body and the manipulator base. By identifying corresponding points in both coordinate systems, we compute the transformation matrix using least-squares method to find the optimal rigid transformation.

Inverse Kinematics

Inverse kinematics transforms our desired gripper position and orientation in Cartesian space into the corresponding joint angles by utilizing analytical and numerical methods that solve the nonlinear equations relating joint configurations to end-effector pose. The process involves calculating the Jacobian matrix to map differential changes in joint space to task space, then iteratively adjusting joint angles using techniques such as the Newton-Raphson method until the end-effector reaches the target pose within an acceptable error threshold, while also handling challenges such as singularities, joint limits, and multiple solutions through optimization criteria such as minimizing joint movement and maintaining manipulability.

2.2.4 Robot Arm Design Module

The Robot Arm Design Module is a critical component of the overall system, responsible for the physical manipulation tasks required by the robot dog. We designed the robotic arm(shown in Figure 6) according to the ARX R5 robotic arm of Unitree [2].

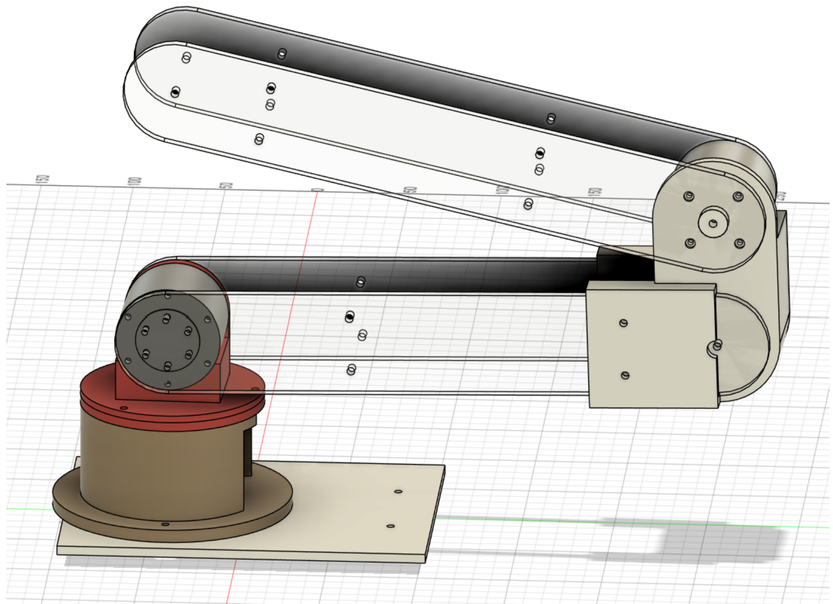


Figure 6. The 3D modelling of robotic arm without gripping jaw

Design Considerations

Tolerance: The tolerance is necessary for the assembly of parts. In our design, we have allocated a tolerance of 2 mm for embedded parts over 10 cm and 1 mm for parts less than 10 cm. For all the screw hole, a tolerance of 0.5 mm has been reserved.

Motor: Choosing the right motor is critical for the design. The motor must provide sufficient torque to move the robotic arm effectively. Additionally, it should be capable of reading its own rotation angle and sending this information back to the development board.

Lightweight design: We use acrylic sheet and PLA as materials for the robotic arm. employing 3D printing and laser cutting to expedite production and reduce the iteration cycle. However, these materials have the disadvantage of low strength compared to metals. So we need to design the arm as lightweight as possible while maintaining its strength.

Components

Our design mainly currently comprises 10 parts: base, bottom motor, bottom motor mounting bracket, link, shoulder, shoulder motor, upper arm, elbow, elbow motor and lower arm. The connection between each part is mainly achieved by different lengths of M3 and M4 screws. There are three other parts: wrist, wrist motor and gripping jaw which we will complete design in the future.

Assemble

The exploded diagram of the design is shown in Figure 7. The structures of the upper arm and lower arm are identical, with arm 1 and arm 2 forming these components. The assembly process is as follows:

1.Fasten the bottom motor mounting bracket to the base using two M4 screws.

2.Insert the bottom motor into the mounting bracket and secure it with six M3 screws.

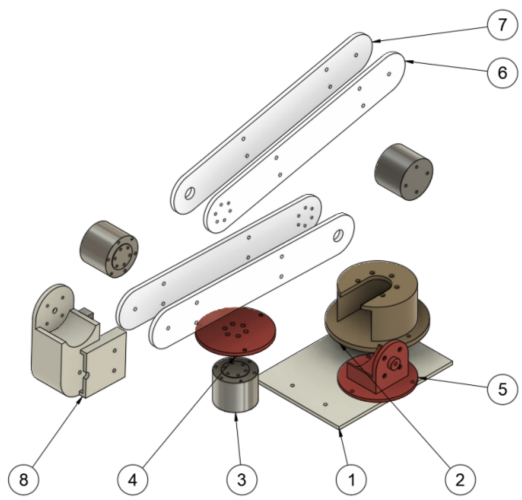
3.Attach the link to the motor's output shaft using six M3 screws, ensuring that three M4 screws are positioned at the outer part of the link.

4.Use these three M4 screws to secure the shoulder.

5.Laterally attach the bottom face of the shoulder motor to the shoulder.

6.Connect arm 1 of the upper arm to the motor's output shaft and join arm 2 to arm 1 using M4 screws and extension nuts.

7.Finally, secure arm 1 and arm 2 to the elbow, followed by fixing the elbow motor and lower arm.



明细表		
项目	数量	零件代号
1	1	Base
2	1	Bottom motor mounting bracket
3	3	Motor
4	1	Link
5	1	Shoulder
6	2	Arm 1
7	2	Arm 2
8	1	Elbow

Figure 7. The explosion diagram of the design without gripping jaw

2.2.5 Robot Arm Control Module

The Robot Arm Control Module is responsible for interpreting high-level commands and converting them into precise joint actuation using a motor-driver-control loop architecture. This module integrates the RoboMaster Developer Board A, RM2006 Brushless Motor, and DM4310 motors over the CAN bus. The Developer Board executes control algorithms, monitors real-time feedback from the motor encoder, and issues position commands to motors, which performs real-time motor commutation using field-oriented control (FOC).

Schematic:

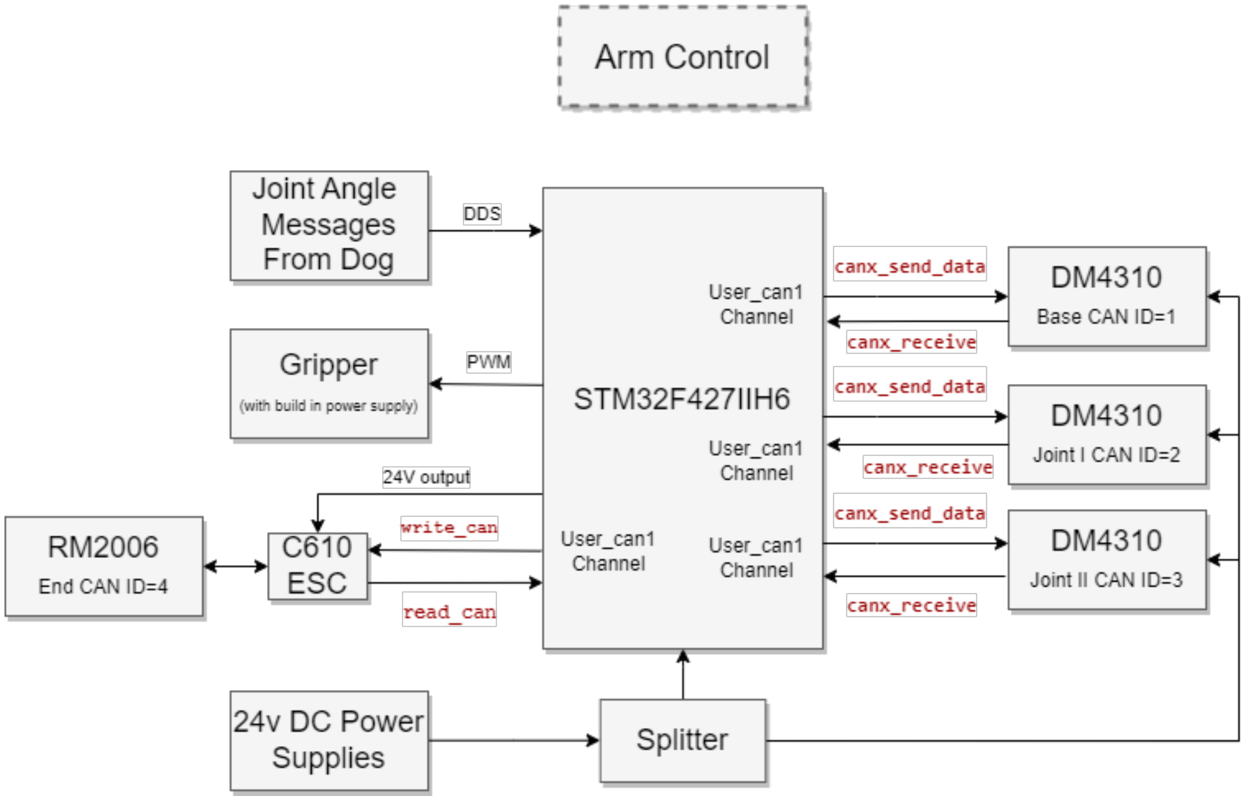


Figure 8: Schematic of robot control module

Central Control Unit: RoboMaster Developer Board A

The RoboMaster Developer Board A [3] serves as the core controller of the robot arm joint. It features a 32-bit STM32F4 microcontroller, high-speed CAN and UART interfaces, and multiple GPIOs. It performs position or torque control by interpreting CAN feedback and issuing real-time motor commands.

- Interfaces:
  - CAN communication with DM4310 for command and feedback.
  - UART debugging and log output.
  - GPIO for emergency stop, limit switches, and LED status indicators.

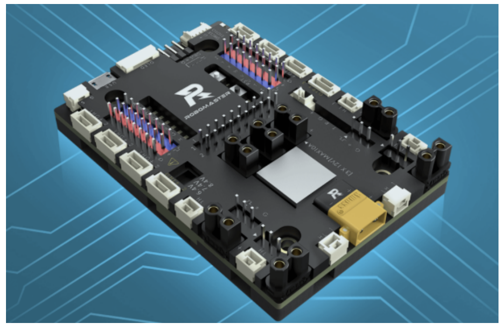


Figure 9: Robomaster Developer Board A

Motor: RoboMaster RM2006

The RM2006 [4] is a compact 24V brushless DC motor with a 36:1 planetary gearbox and integrated Hall encoder. It provides up to 1.0 N·m of continuous torque and is ideal for robotic joints requiring high accuracy and moderate load. The onboard encoder supports position tracking through the DM4310 driver.

- Electrical specs:
    - Rated voltage: 24V
    - Torque constant: 0.18 N·m/A

Figure 10: Robomaster RM2006

Motor: DM4310 Brushless DC Servo Motor

The DM4310 [5] is a brushless DC servo motor with integrated FOC control and position feedback. It operates in position mode via the CAN protocol and integrates an encoder for closed-loop control. The DM4310 receives target angle commands from the Developer Board and maintains joint position accurately under external loads. The motor is designed for precise robotic joint actuation, making it ideal for applications such as robotic arms, where accurate and responsive movement is critical.

Features:

- Integrated FOC controller and position loop

- CAN-based position control protocol

- Encoder-based feedback loop for closed-loop control

- High holding torque for robotic joint stability

Figure 11: DM4310 Brushless DC Servo Motor

2.3 Subsystem Requirements and Verifications

Please see the attached Appendix A

2.4 Tolerance Analysis

Yolo model:

Although the YOLO model does not achieve 100% accuracy in predicting bounding boxes—resulting in slight positional deviations—this tolerance is acceptable for our application. The typical error in bounding box predictions is less than one centimeter, which may lead to minor inaccuracies in determining the exact location of a mobile phone. However, the gripper is designed with a tolerance that can accommodate several centimeters of error relative to the phone's width. Consequently, the small discrepancies introduced by the YOLO model are effectively absorbed by the robotic system, ensuring that object detection and subsequent manipulation tasks can be carried out successfully without compromising overall performance.

Robot Arm:

During the robotic arm's movement, small angle errors may occur due to factors such as manufacturing tolerances, assembly discrepancies, sensor precision, and load variations. These errors, while present, are generally minor and fall within an acceptable tolerance range, which is accounted for during the design phase. Compensation mechanisms in the control algorithm, along with precise sensor feedback, help minimize the impact of these deviations. As a result, the errors are manageable and do not significantly affect the arm's performance, making them tolerable within the system's operational limits. Assume the error of angle is $\Delta\theta$, the position error in Cartesian space could be expressed as:

$$\Delta\, p_{effector} = J(\theta) \cdot \Delta\, \theta$$

This error is typically small, and assuming the Jacobian is well-conditioned, it results in an end-effector error stays within an acceptable tolerance, ensuring that the arm's performance remains accurate and reliable.

3. Cost & Schedule:

3.1 Cost

All cost is in RMB.

Part	Item(s)	Cost
Control Unit	RM Developer board A	429
Motors	RM2006 * 1 DM4310 * 3	2234
24V power supply	WHEELTEC P760S Splitter	277
Wiring	XT60 MtoF * 1 XT30 MtoF * 3	76
Test object	Smartphone model	35
Structural design	3D printing Material	200
Structural design	Acrylic plate	40
Structural design	M3 ×10 Screw	2.27
Structural design	M3 × 8 Screw	2.2
Structural design	WD-40 lubricant	17.9
Structural design	M4×50 Extension nut	4
Gripping module	Gripping jaw	96

3.2 Schedule

2.17-3.9	Plan Project, Write RFA and Proposal (All)
3.10-3.16	Servo selection (Yikai); Design arm components (Wenkang); Go2 simulation setup, camera feed retrieval (Jitao, Teng)
3.17-3.23	Design arm components with servo accomodation(Wenkang); set up and configure a single motor (Yikai); Visual detection dataset construction (Teng, Jitao)
3.24-3.30-	Finalize arm design (Yikai, Wenkang); Train visual detection model (Jitao, Teng)
3.31-4.6	Print out arm components (Wenkang); Achieve multi-servo coordination (Yikai); Deploy visual detection model on Go2 (Jitao, Teng)
4.7-4.13	Assemble arm (Yikai, Wenkang); Achieve forward kinematics of arm in MuJoCo (Jitao); Test visual model on Go2 (Teng)
4.14-4.20	Debug arm to function, with angle input/movement and angle feedback, without error (All)

4.21-4.27	Achieve inverse kinematics of arm and grasp in MuJoCo (Jitao); Assemble arm to dog (Yikai, Wenkang); Set up dog to move towards detected object (Teng)
4.28-5.4	Test grasping using arm (Jitao, Yikai); Integrate arm and dog integration in MuJoCo (Teng, Wenkang)
5.5-5.11	Sim2Real, Integrate arm and dog system to grasp smartphone model (All)
5.12-5.18	Debug demo (All)

4. Ethics & Safety

We always adhere to what we have committed to in our proposal.

Ethics:

We are committed to upholding the highest ethical standards and ensuring the integrity of our project by strictly adhering to the IEEE Code of Ethics [6]. In doing so, we clearly specify the intended application and operational constraints of our system (IEEE Code 2) and accurately describe workspace and payload capabilities (IEEE Code 5). We also recognize and promptly address any technical deficiencies, while properly crediting all team members for their contributions (IEEE Code 6). To protect personal privacy, our vision system avoids collecting unnecessary information (IEEE Code 1), and we consider the environmental footprint of our design and production processes—particularly in relation to 3D printing (IEEE Code 1). Furthermore, we establish open communication channels for reporting concerns and issues (IEEE Code 7). Throughout our work, we prioritize public safety and well-being, remain vigilant about potential negative social or environmental impacts, and maintain honesty and integrity in all professional activities, thus avoiding unethical conduct such as bribery or illegal actions.

Safety:

To ensure the safety of both team members and equipment, we implement high-voltage protection when working with motor drivers and power systems and conduct regular checks of all power cables and connections. We provide clear instructions for operating the robot dog and manipulator arm, along with an emergency stop mechanism that can immediately halt operation if necessary. This is further supported by torque limiters and mechanical stops to prevent overload and over-extension. Regular maintenance of all mechanical components and a limit on the end effector's maximum grip force ensure safe handling of objects. Dedicated testing areas in the lab, along with adequate ventilation—particularly during 3D printing—help maintain a secure working environment. We also prohibit team members from working alone in the laboratory, enforce strict guidelines for handling and charging batteries or other potentially hazardous materials, and document these measures in standard operating procedures to ensure consistent safety protocols throughout the project.

References

[1] Ultralytics. (2025, March 16). Home. Ultralytics YOLO Docs. https://docs.ultralytics.com/

[2] Unitree ARX R5 robotic arm. https://www.arx-x.com/?product/22.html

[3] DJI RoboMaster. RoboMaster Development Board User Manual. Shenzhen, China: DJI, 2018.

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

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

[6] IEEE, IEEE Code of Ethics, Online, Available: https://www.ieee.org/about/corporate/governance/p7-8.html, 2020.

Appendix A Requirement and Verification Table

Table: System Requirements and Verifications			
Module	Requirement	Verification	Verification status (Y or N)
Robot Dog Control Unit	1. The robot dog can walk and change the direction normally.	1. It is easy to verify that the dog could move and rotate.	Y
Robot Dog Vision Module	1. Module should detect the object with a predicted box. 2. The robot dog should search the object automatically and move toward the object.	1. Running the trained yolo model and get the box data with the value output. 2. After executing the python script that built with sdk2, the model should be automatically running and give the command to the dog moving to the object.	Y N
Transformation Module	1. Accurately compute transformation matrix from camera coordinates to dog coordinates (T_cam2dog) 2. Accurately compute transformation matrix from dog coordinates to arm base coordinates (T_dog2arm) 3. Implement inverse kinematics to convert Cartesian coordinates to joint angles for the manipulator arm	1. Validate transformation accuracy using known reference points. 2. Validate transformation using least-squares optimization and measure transformation error with known test positions. 3. Test the IK algorithm with multiple target positions throughout the workspace. Measure positioning accuracy of the end effector and verify that joint angle solutions respect physical constraints and singularities are properly handled.	N N Y
Robot Arm Design Module	1. The robot arm must support a minimum load of 1 kg. 2. The shoulder part of arm must achieve a range of motion of at least 180°. 3. The bottom motor of arm must achieve a range of motion of at least 360°.	1. Load testing with calibrated weights. 2. Manual range of motion test. 3. Manual range of motion test. 4. Manual range of motion test. 5. Dimensional inspection of parts.	N Y Y Y N

	4. The elbow part of arm must achieve a range of motion of at least 180°. 5. The assembly must have a tolerance of 2 mm for parts > 10 cm and 1 mm for parts < 10 cm.		
Robot Arm Control Module	For developer board: 1. Must maintain a CAN update loop at 1 kHz 2. Must parse encoder position feedback from RM2006 3. Must compute control loop with latency <1 ms 4. Must handle motor enable, disable, and fault reset states For RM2006 motor: 1. Must hold position under external torque (≤ 1.0 N·m) 2. Must provide encoder data with <5° resolution 3. Must not exceed 100°C in sustained operation 4. Must communicate via encoder passthrough For DM4310 motor: 1. Must maintain position mode under target angle commands 2. Must return current position over CAN feedback 3. Must resist external torque disturbances 4. Must support high-frequency CAN updates (≥1kHz)	For developer board: 1. Measure round-trip latency using timestamped CAN packets 2. Simulate position steps and verify via debug UART output 3. Profile control task with \`micros()\` time-stamping in FreeRTOS 4. Trigger errors via induced fault and observe auto recovery For RM2006 motor: 1. Apply external load and verify angle holding within 1° 2. Read encoder value and verify against external protractor 3. Attach thermocouple during 5-minute torque test 4. Read CAN message encoder value during joint rotation For DM4310 motor: 1. Send static target angles and verify holding accuracy within 1° 2. Poll CAN frames and confirm real-time angle updates 3. Apply force to joint and confirm positional recovery within 1° 4. Measure response timing via oscilloscope on LED trigger or GPIO	Y Y Y Y Y Y Y N Y Y Y N