Wheeled-Legged Robot Design Document

Team 3
Gabriel Gao(ngao4)
Jerry Wang(runxuan6)
Zehao Yuan(zehaoy2)

Fall 2023
TA: Tianxiang Zheng

Contents

1 Introduction **2**
1.1 Problem . . . 2
1.2 Solution . . . 2
1.3 Visual Aid . . . 3
1.4 High-level Requirements List . . . 3

2 Design **4**
2.1 Block Diagrams . . . 4
2.2 Hardware Subsystem Description . . . 5
2.2.1 Hybrid Mobility Subsystem . . . 5
2.2.2 Central Control subsystem . . . 8
2.2.3 Testing Platform Subsystem . . . 11
2.2.4 Payload Compartment Subsystem (3D-printed) . . . 11
2.2.5 Remote Controller Subsystem . . . 12
2.2.6 Power System . . . 13
2.3 Software Description . . . 14
2.3.1 Physical Model . . . 14
2.3.2 High-level Software Design . . . 17
2.3.3 Motor Communication . . . 20
2.4 Tolerance Analysis . . . 23

3 Cost and Schedule **24**
3.1 Cost Analysis . . . 24
3.2 Schedule . . . 26

4 Ethics and Safety **26**
4.1 Safety . . . 26
4.2 Ethics . . . 27

5 Citation **28**

1 Introduction

1.1 Problem

The inspiration for this project stems from the challenges associated with conventional wheeled delivery robots, most of which are equipped with a four-wheel chassis. This configuration hampers their ability to traverse terrains with obstacles, bumps, and stairs—all commonplace in urban environments. A wheeled-legged balancing robot, conversely, can navigate these obstacles with ease, positioning it as a particularly attractive solution for delivery services in urban areas.

1.2 Solution

This project seeks to address this gap by focusing on the development of a wheel-legged robot that can adeptly maneuver through challenging urban terrains.

To ensure feasibility within our time frame, our initial focus will be on creating a dynamic, single robotic leg capable of demonstrating its potential as an electric suspension system capable of bearing weight and adapting to various terrains. This proof-of-concept will serve as a cornerstone for the future development of a complete robot, potentially revolutionizing urban delivery systems.

1.3 Visual Aid

Figure 1: Visual Aid SolidWork after rendering

1.4 High-level Requirements List

1. The leg motors should provide a 4 Nm continuous torque to let the overall structure stand on the support of the legs and be able to apply about 1-2.5lb loads on the top of the platform.

2. The robot should respond to the remote control within 300ms and be able to execute pre-calculated trajectories in real time to demonstrate jumping and partial walking gaits.

3. The system's power management should ensure sustained and stable operation of all components for at least 15 minutes, with the ability to quickly shut down in 2s in emergencies, safeguarding both the robot and its environment.

2 Design

2.1 Block Diagrams

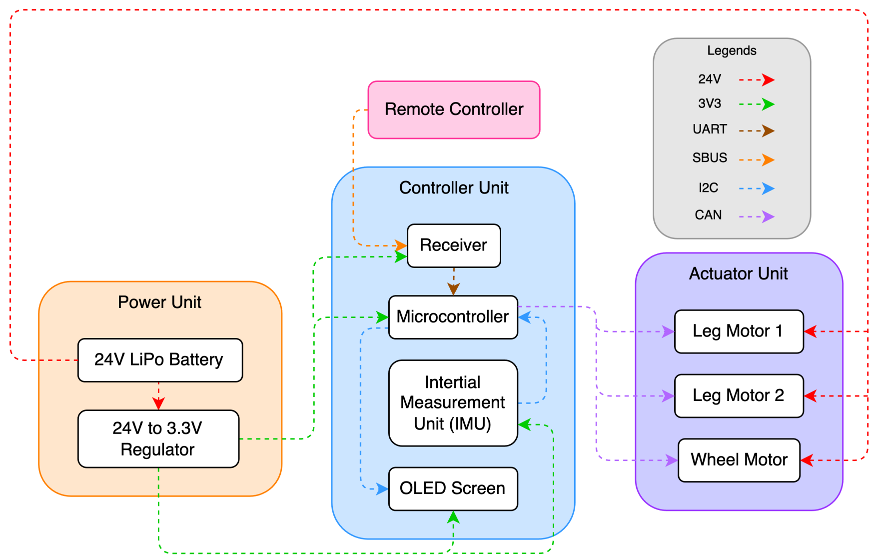

Figure 2: Block diagram of our design

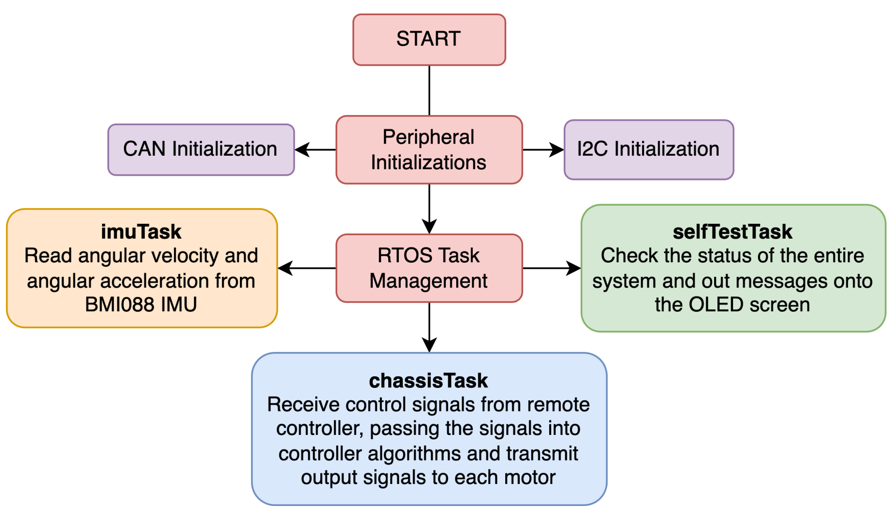

Figure 3: Software block diagram of our design

2.2 Hardware Subsystem Description

2.2.1 Hybrid Mobility Subsystem

1. **Actuated Legs**

   The Wheeled-Legged Balancing Robot Leg project uses two actuator motors (DM-J4310-2EC) to empower the legged system, enabling navigation across uneven surfaces, obstacles, and stairs. The legs double as an advanced electromagnetic suspension system, adept at making swift adjustments in damping and stiffness to consistently sustain a stable and level platform. The motor has a power supply ranging from 20-27V, allowing direct connection to a 24V battery system. The legged system has 4Nm continuous torque, 120RPM, and can consistently communicate with the STM32 through a can bus, ensuring the leg has enough performance to quickly adjust the angle in diverse terrains and conditions.

Requirement	**Verification**
• The motor should be able to supply a continuous torque of at least 4Nm $\pm$ 5% at 120RPM $\pm$ 5%.	• Using a script to read the torque value from the data given by the motors or Using the rotary torque sensor to measure the torque. Check whether the values from either method are larger than 4 Nm $\pm$ 5%. • Using a tachometer to measure the motor's RPM to confirm it can achieve 120 RPM $\pm$ 5% under the specified torque load.
• The motors should be able to be powered by 20 - 27V.	• Using an adjustable power supply to test the motors at various voltages within the range of 20-27V to ensure consistent and efficient performance.

Table 1: Actuated Legs–Requirements & Verification Pt.1

Requirement	Verification
• The motors must maintain a communication rate of 1kHz ± 10HZ with the STM32 through the CAN bus.	• Use a CAN bus analyzer or other appropriate testing tools to measure the communication rate between the motors and the STM32. Ensure the communication rate remains stable at 1kHz ± 10HZ.
• The encoder of the motor should have an angle resolution of less than 1 degrees • Able to remember the correct motor zero point after power cutoff.	• Run multiple commands to rotate the motor to a certain angle, and measure whether each angle rotation matches the control command. Rotate the motor at a certain angle and read the motor angle feedback, comparing whether the angle feedback is within 1 degree of the actual difference. • After setting the motor to zero point, power off and rotate the motor angle. After powering on, check if the motor remembers the correct zero position

Table 2: Actuated Legs–Requirements & Verification Pt.2

2. **Wheeled Drive**

   The Wheeled-Legged Balancing Robot uses a direct drive BLDC motor (M3508) to efficiently propel the wheels for proficient movement on flat surfaces. This setup employs a 60mm diameter polyurethane wheel, ensuring durability and stability during operation. The M3508 motor can maintain a communication rate of 1kHz ± 10HZ with the STM32 to guarantee enough communication rate for smooth and controlled movements across diverse terrains.

   This motor needs to deliver a continuous torque of at least 0.1Nm and a rotational speed of at least 2000 RPM. This specification is pivotal to moving the entire robotic system under an approximate load of 2.5lb, ensuring the robot's robust performance and reliability in various operational conditions.

Requirement	**Verification**
• M3508 motor maintains a communication rate of 1kHz ± 10Hz with the STM32.	• Use a CAN bus analyzer or other appropriate testing tools to measure the communication rate between the motors and the STM32. Confirm the communication rate remains stable at 1kHz ± 10HZ.
• The motors should be able to be powered by 20 - 27V.	• Using an adjustable power supply to test the motors at various voltages within the range of 20-27V to ensure consistent and efficient performance.
• M3508 motor delivers a continuous torque of at least 0.1Nm and achieves a speed of at least 2000 RPM. • M3508 motor can move the robot under a load of around 2.5lb.	• Using a script to read the torque value from the data given by the motors or Using the rotary torque sensor to measure the torque. Check whether the values from either method are larger than 0.1 Nm ± 5% • Using a tachometer to measure the motor's RPM to confirm it can achieve 2000 RPM ± 5% under the specified torque load. • Load the robot with a weight of 2.5lb and verify the motor moves the robot effectively.
• The encoder of the motor should have an angle resolution of less than 1 degrees	• Run multiple commands to rotate the motor to a certain angle, and measure whether each angle rotation matches the control command. Rotate the motor at a certain angle and read the motor angle feedback, comparing whether the angle feedback is within 1 degree of the actual difference.

Table 3: Wheeled Drive–Requirements & Verification

2.2.2 Central Control subsystem

1. **PCB and Microcontroller**

   The central element of our design is the STM32F103 microcontroller It processes vital inputs from the IMU (BMI088) using SPI signals, ensuring accurate and real-time responses. This microcontroller also effectively manages the movements of the motors via the CAN bus, ensuring coordinated and precise movement. Our designed PCB includes the necessary power supply elements and a Dbus remote control signal inverter, STM32F103 chip, and the BMI088 IMU. Providing a compact and efficient layout for optimal performance.

Requirement	**Verification**
• Capable of processing inputs efficiently and directing outputs to various peripherals.	• Perform an on-off test and voltage drop test on the entire PCB board using a multimeter. After the first two tests are successful, power up the PCB board and measure the voltage of the circuit components to see if they are regular. Then using scripts to test the functionality of each interface.
• Must have protection mechanisms against power surges or short circuits.	• Check the input and output capacitors before powering the PCB board to ensure they can filter voltage fluctuations. • Using the adjustable DC power supply to test the design under various voltage and current conditions to ensure protection mechanisms work effectively.
• STM32F103 should run at 72MHZ $\pm$ 5MHZ.	• Use a frequency meter to confirm the operating frequency of the STM32F103 during operation.

Table 4: Central Control–Requirements & Verification Pt.1

Requirement	Verification
• PCB should consume $\leq$ 1W to ensure efficient power consumption.	• Using the adjustable DC power supply, which can show the input voltage, input current, and input power, to measure the power consumption of the design to ensure it does not exceed 1W.
• Maintaining Can bus load under 80% between the IMU and the motors to ensure stable communication.	• Run a script and Use a CAN bus analyzer to monitor the CAN bus load, ensuring it stays below 80% for stable communication.
• Capable of processing and interpreting signals from the IMU accurately.	• Put the PCB board on a horizontal table. Then, power the PCB board and run the script to check whether the IMU returns the zero point to the microcontroller. Then, rotate the PCB board 45 degrees counterclockwise 8 times and check whether the IMU returns the correct data for each rotation.
• Minimize the time used in signal processing for real-time applications to 5ms$\pm$ 5%.	• Using the inner clock to record signal processing time during the operation.
• Must provide precise control signals to the motors via the CAN bus.	• Connect the PCB board and the motors and use the scripts to send the rotation signal to the motors. Check whether the motors rotate to the desired position.

Table 5: Central Control–Requirements & Verification Pt.2

2. **Attitude Sensing**

   The robot use a 6-axis IMU to perpetually track its orientation and movement. This continuous monitoring allows the system to make immediate adjustments, ensuring the robot's stability and precise navigation throughout its operation. The IMU plays a crucial role in providing the necessary data that aids the robot in adapting to different terrains and situations.

Requirement	**Verification**
• Low bias error and drift. • Low noise levels, High resolution	• Read the data from IMU in a normal environment and Compare it with the data from the reference sensor. • Let the PCB work at a high electromagnetic interference environment and check the return data with the reference sensor.
• Suitable range and high sampling rate to get reasonable data.	• In a stable state of the IMU, connect the data output pins of the IMU to an oscilloscope and observe the waveform of the output signal. By measuring the time interval between two consecutive peaks, we can estimate the sampling rate.
•Ability to be calibrated and temperature-compensated	• Use a script to send the calibrating signal to IMU and use the return data to check whether the IMU is calibrated. Then, Keep monitoring the temperature data from IMU during the operation to check whether it can be temperature-compensated.
•Low power consumption	• Using the adjustable DC power supply, which can show the input voltage, input current, and input power, to measure the power consumption during operation.
•Common interfaces (I2C, SPI, UART, or CAN)	• Using simple scrips to check whether the microcontroller can send the signals to IMU and receive the IMU signals.

Table 6: Attitude Sensing – Requirements & Verification

2.2.3 Testing Platform Subsystem

The leg is attached to a harness as depicted in this sketch. This harness simplifies the model by limiting the robot's movement to a circular path, yet allowing for z-axis jumps. The robot movement configuration space should be a cylinder with a base radius of 40cm and a height of 50cm.

Requirement	**Verification**
• The harness must be stable enough to stay **stationary** during the robot movement.	• Check that there should be no offset between the harness and the ground when the robot jumps or moves at top speed. A boundary can be drawn around the harness to check whether the harness moves away from its original position.
• The robot movement configuration space constrained by the harness should be a cylinder with a base radius of 40cm and a height of 50cm.	• Using rules to measure the length of the vertical link and horizontal link of the harness. The length of the vertical link should be larger than 60cm and the horizontal link should be 35-40cm.

Table 7: Testing Platform - Requirements & Verification

2.2.4 Payload Compartment Subsystem (3D-printed)

The payload compartment subsystem is a special section that is designed to securely hold and transport items. This subsystem can shield them from disturbances during the journey. It should be a 10cm x 15cm x 10cm box with covering. This compartment should withstand 1-2.5lb loads.

Requirement	**Verification**
• The compartment should be a **10cm x 15cm x 10cm** box with covering.	• Using rules to measure the length, width, and height of the compartment to ensure the error is no more than 5%

Table 8: Payload Compartment - Requirements & Verification Pt.1

Requirement	Verification
• The compartment should withstand **1-2.5lb** loads.	• Using an electronic weighing machine to measure 2.75lb objects and put them inside the compartment. Then, lift the compartment by hand and move it for 10 min. Check whether the compartment has any damage like fissures.

Table 9: Payload Compartment - Requirements & Verification Pt.2

2.2.5 Remote Controller Subsystem

A 2.4 GHz RC remote controller using the Dbus protocol will enable the user to control the robot. The user can use this remote control to send commands to the robot for forward, backward, elevation, and jumping actions. This portable device allows for real-time control over varying distances, with safety features like an emergency kill switch.

Requirement	Verification
• The user can use this remote control to send commands to the robot successfully.	• Using the script on the microcontroller to print the commands sent by the remote controller so that we can visualize the commands on the computer screen.
• The remote control must maintain reliable communication with delay $\leq$ 10ms.	• When the microcontroller receives the data from the remote controller, we can add the time stamp at that moment. Then we can let the remote controller send the same signal continuously. The time interval between the time stamps is the communication delay. Check whether the delay $\leq$ 10ms.

Table 10: Remote Controller - Requirements & Verification Pt.1

Requirement	Verification
• The remote control must maintain an emergency kill switch to stop the robot in 2s.	• Let the robot run at a low speed and then toggle the emergency kill switch. Use a timer to record the stop time and check whether the robot is killed $\leq$ 2s.

Table 11: Remote Controller - Requirements & Verification Pt.2

2.2.6 Power System

The robot is planned to be powered by a 6s (24V) Lithium Battery, chosen for its capability to offer high power output within a safe voltage range. This type of battery stands out for its smaller dimensions and higher energy density, advantageous features for the compact and efficient design of the robot. To accommodate the microcontroller, a DC-DC converter will step down the voltage from 24V to 5V and then further to 3.3V, ensuring the right power level is supplied to each component. The ease of purchasing 24V lithium-ion batteries adds to their suitability for this project, simplify the assembly process and easy to replace.

Requirement	Verification
DC-DC Converter 1: • Input: 20-27V Output: 5V $\pm$ 0.5V, $\geq$ 1A.	• Use a multimeter to measure input and output voltage and current to ensure they are within specified limits.
DC-DC Converter 2: • Input from Converter 1 Output: 3.3V $\pm$ 0.3V, 500mA $\pm$ 5%.	• Use a multimeter to measure input and output voltage and current to ensure they are within specified limits.
Battery: • Supply: 20-27V, 20A peak, 5A continuous Runtime: $\geq$ 30 minutes at 5A.	• Use a multimeter to measure the output voltage and current from the battery under load. Use a timer to ensure it lasts for at least 30 minutes at a 5A load.

Table 12: Power System - Requirements & Verification Pt.1

Requirement	**Verification**
Safety and Flexibility: • Quick power cut-off: $\leq$ 200ms Switch between battery and wired power.	• Test the cut-off response time with a stopwatch or electronic timer. Check the seamless switching between power sources under operation.

Table 13: Power System - Requirements & Verification Pt.2

2.3 Software Description

2.3.1 Physical Model

To implement this wheeled-legged robot, we need to establish a dynamic model. The whole-body motion of a wheeled-legged bipedal robot can be decoupled into wheeled motion and leg motion. Wheeled motion includes planar motion (body balance, forward and backward movement) and steering motion. Leg motion includes jumping motion, height adjustment motion, and ground adaptive motion.

Variable and Parameter Declaration are shown in Figure 4 and Figure 5 from our Lab Notebook.

Variables		
Label	**Meaning**	**Unit**
x_L, x_R	The displacement of the left and right wheels	m
y	The distance between the body's center of mass and wheel motor rotation axis along y axis	m
ϕ	The roll angle of the body	rad
θ	The pitch angle of the body	rad
ψ	The yaw angle of the body	rad
T_L, T_R	The output torque of the left and right wheel motors	$N \cdot m$
T	The output torque of the leg motors	$N \cdot m$
N_L, N_R	The horizontal component of the force between wheels and the body (along x axis)	N
P_L, P_R	The vertical component of the force between wheels and the body (along y axis)	N
F_L, F_R	Th e frictions of the wheels when moving	N

Figure 4: Variable Declaration

Parameters		
Label	**Meaning**	**Unit**
m	The mass of rotor in the wheel motors	kg
M	The mass of the body	kg
I_w	The moment of inertia of rotor in the wheel motors	$kg \cdot m^2$
I_x	The moment of inertia of the body rotated around the x axis	$kg \cdot m^2$
I_y	The moment of inertia of the body rotated around the y axis	$kg \cdot m^2$
I_y	The moment of inertia of the body rotated around the z axis	$kg \cdot m^2$
R	The radius of the wheel	m
L	The distance between body's center of mass and the rotation axis of the wheel motor	m
D	The distance between left and right wheels	m
g	The acceleration due to the gravity measured	m/s^2

Figure 5: Parameter Declaration

For the Wheeled motion, we can assume

1. The mass of the body can be represented at the center of the mass.
2. Ignore the mass of the led linkage and the effect on wheel motion from leg movement.
3. No sliding on the wheels.

Therefore, As shown in Figure6, based on the aforementioned idealized assumptions, the wheeled motion of the wheeled-legged robot can be equivalently simplified into a double-wheeled inverted pendulum model with adjustable pendulum length.

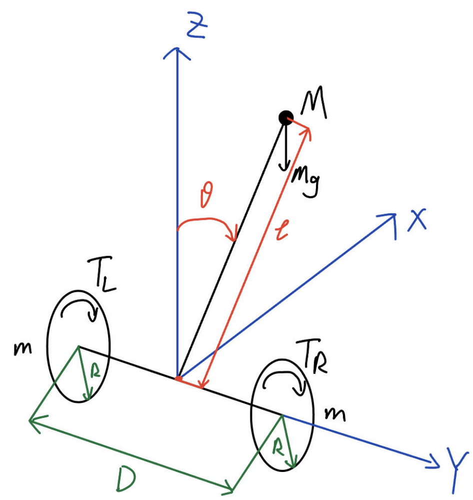

Figure 6: Wheeled Motion Simplified Model

For the leg motion, we can divide the leg motion of the wheeled-legged robot into two scenarios: the support phase and the jumping phase, and model their dynamics separately. In the support phase, both left and right drive wheels maintain continuous contact with the ground during leg motion. During this phase, the leg link's motion speed is relatively slow, and after the extension and retraction motion of the leg links, they stabilize in their target positions, representing height adjustment and ground adaptive motions. The jumping phase refers to the robot's jumping process, where the robot's drive wheels leave the ground after takeoff, and the leg links extend and retract rapidly during the jump.

Several idealized assumptions are made regarding the robot's leg motion:

1. The mass of the robot is effectively concentrated at the center of mass.
2. The influence of wheeled motion on leg motion is neglected.
3. The mass of the leg links is ignored.
4. The robot maintains a balanced state throughout the leg motion process, i.e., the pitch angle remains zero.
5. The two joint motors on the same side respond synchronously with equal magnitudes of control torque but in opposite directions.

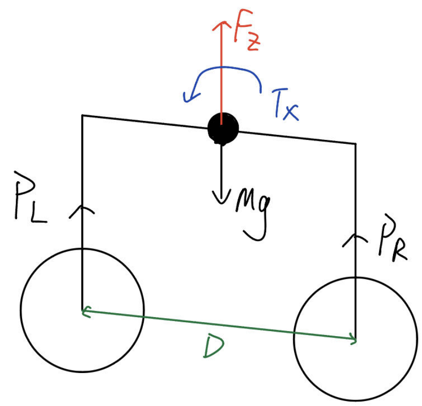

Figure 7: Leg Motion Simplified Model

Therefore, As shown in Figure7, based on the above fundamental assumptions and by treating the leg's five-link structure as an equivalent rigid rod whose length varies with the joint motor angles, we establish a static model for the support phase.

2.3.2 High-level Software Design

1. **IMU task**

   The primary function of the IMU task is to read the relative angular velocity and acceleration from the BMI088 module and perform attitude calculations for the IMU. In the main program loop, the imuTask first reads the angular velocity and angular velocity of each axis from the gyroscope and accelerometer. The program samples the gyroscope data and calculates the average error. This error is subtracted from the original values of each axis of the gyroscope to reduce the impact of drift on the gimbal. Using angular velocity and speed data, the program uses the Mahony AHRS algorithm to perform attitude calculation for the IMU, obtaining the quaternion matrix, roll, pitch, and direction data.

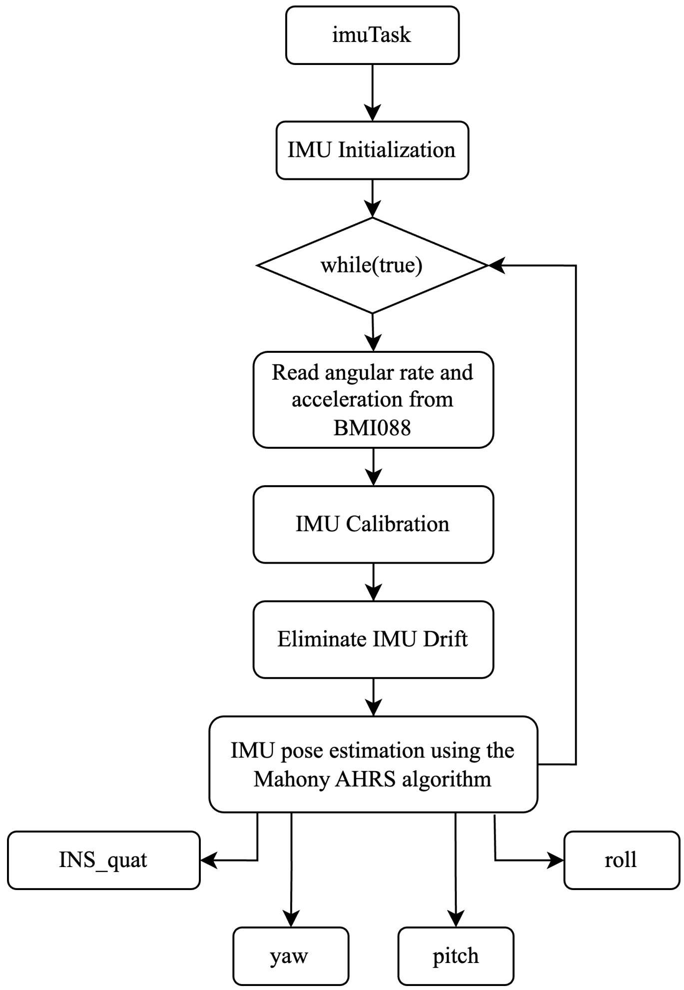

Figure 8: IMU Task

2. **Chassis Task**

   The primary function of the chassis task is to decouple the high-level commands from the microcontroller into motor-level commands, process the commands, and then send them to the motors. When the robot is started, it will automatically initialize the all motors to ensure the motor works regularly. After that, the robot will initialize the control algorithm to prepare for the data processing. Then, when the user uses the remote controller to send high-level commands like moving forward or rotating, the microcontroller will combine the IMU data and the remote controller data to calculate the corresponding ideal torque that the motors should output. Then, it will send the ideal torque to the control system and the controller will process the data again and output the real desired torque to each motor. Finally, the microcontroller will send the data to each motor and repeat the previous steps again. The flow chat is shown in Figure 9.

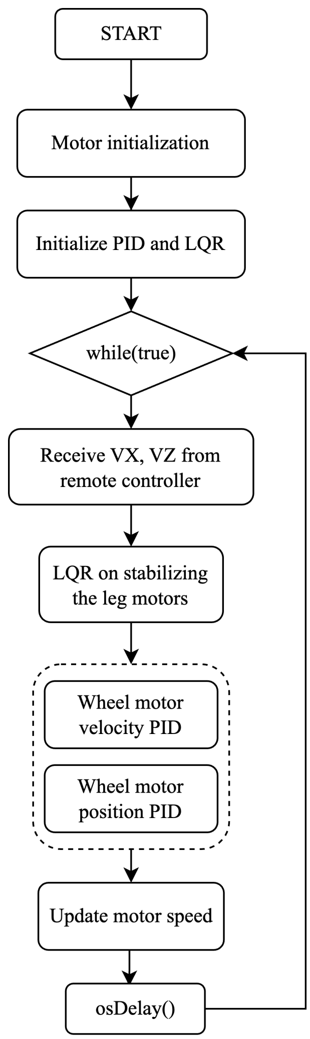

Figure 9: Chassis Task

2.3.3 Motor Communication

We are using the CAN bus to communicate with all motors. Different motors have different package structures of feedback and control frames.

DM-J4310-2EC

The DM4310 motor has three different operating modes: MIT, position-velocity, and velocity mode. The MIT mode is the mode that we will be using for our leg motors. This mode has five control parameters: **P_des** (desired position), **V_dest** (desired velocity), **kp** (positional proportion coefficient), **kd** (positional differential coefficient), and **T_ff** (desired torque). An advantage of using the MIT mode is that we can easily extend it to different control schemes. For example, setting kp and T_ff to 0 and Kd to a constant value allows the motor to rotate at a constant velocity; setting kp V_dest to 0 and kd to a constant value allows us to control the motor using torque. The block diagram of the control system for MIT mode is shown in figure 10.

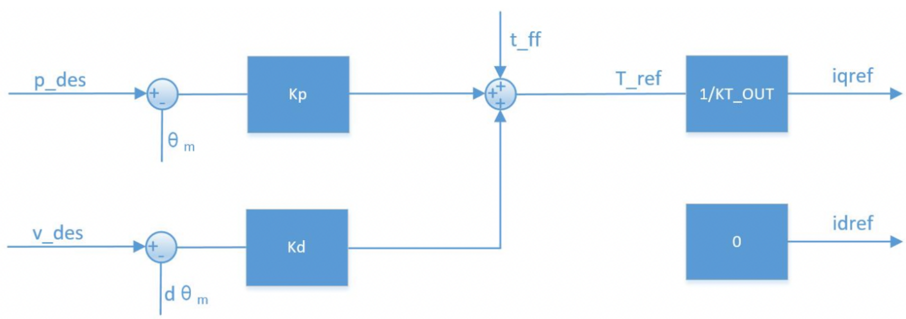

Figure 10: MIT Mode Control System Block Diagram

The feedback frame of the DM4310 motor is the same for all three modes. It consists of a controller ID, an error code, position, velocity, torque, MOS temperature, and rotor temperature.

msg	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	ID|ERR<<4	POS[15:8]	POS[7:0]	VEL[11:4]	VEL[3:0]|T[11:8]	T[7:0]	T_MOS	T_Rotor

The control frame is distinct for each mode. For MIT mode, the control frame consists of a configured CAN ID, desired position, desired velocity, Kp, Kd, and torque. The five parameters are split into the 8 bytes of the control frame as shown in the table below. In particular, position takes 16 bits, velocity, kp, kd, and torque takes 12 bits each.

msg	D[0]	D[1]	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]
MST_ID	p_des[15:8]	p_des[7:0]	v_des[11:4]	v_des[3:0]|Kp[11:8]	Kp[7:0]	Kd[11:4]	Kd[3:0]|t_ff[11:8]	t_ff[7:0]

Each parameter is transmitted as an integer. Before transmitting parameters, we need to convert each field from float to int. Similarly, when we receive a frame, we need to convert each parameter from integer back to float. The conversion can be achieved by the following functions, where x in uint_to_float is the integer value we are converting from and x in float_to_uint is the float value we are converting from. x_min and x_max are the maximum and minimum values for each parameter and bits is the number of bits for each parameter.

```
float uint_to_float(int x, float x_min, float x_max, int bits){
    float span = x_max - x_min;
    float offset = x_min;
    return ((float)x)*span/((float)((1 << bits)-1)) + offset;
}
```

```
int float_to_uint(float x, float x_min, float x_max, int bits) {
    float span = x_max - x_min;
    float offset = x_min;
    return (int) ((x-offset)*((float)((1 << bits)-1))/span);
}
```

M3508 & C620

M3508 is the wheel motor that we will be using and C620 is its ESC that communicates with our microcontroller with CAN[1]. The control and feedback frame for C620 is completely different from those of DM4310. Each control frame can contain the target current for four motors. The two identifiers (0x200 and 0x1FF) control the current output of each of the four speed controller by their ID. 0x200 corresponds to controllers with ID 1 to 4, and 0x1FF corresponds to controllers with ID 5 to 8. Different to DM4310, we are sending current commands directly to control M3508, meaning that we need to perform closed-loop control to adjust the speed and position of each motor in our microcontroller.

Data Fields	Description	Speed Controller ID
DATA[0]	Controls the current value in higher order byte (8 bits)	1
DATA[1]	Controls the current value in lower order byte (8 bits)	
DATA[2]	Controls the current value in higher order byte (8 bits)	2
DATA[3]	Controls the current value in lower order byte (8 bits)	
DATA[4]	Controls the current value in higher order byte (8 bits)	3
DATA[5]	Controls the current value in lower order byte (8 bits)	
DATA[6]	Controls the current value in higher order byte (8 bits)	4
DATA[7]	Controls the current value in lower order byte (8 bits)	

Data Fields	Description	Speed Controller ID
DATA[0]	Controls the current value in higher order byte (8 bits)	5
DATA[1]	Controls the current value in lower order byte (8 bits)	
DATA[2]	Controls the current value in higher order byte (8 bits)	6
DATA[3]	Controls the current value in lower order byte (8 bits)	
DATA[4]	Controls the current value in higher order byte (8 bits)	7
DATA[5]	Controls the current value in lower order byte (8 bits)	
DATA[6]	Controls the current value in higher order byte (8 bits)	8
DATA[7]	Controls the current value in lower order byte (8 bits)	

Figure 11: CAN Frame for Data Send to C620

The feedback frame of C620 is shown in figure 12. It includes the motor's angle, rotational speed, torque current, and motor temperature. The feedback frequency of C620 is 1KHz by default.

Data Fields	Description
DATA[0]	Controls the rotor mechanical angle in higher order byte (8 bits)
DATA[1]	Controls the rotor mechanical angle in lower order byte (8 bits)
DATA[2]	Controls the rotational speed in higher order byte (8 bits)
DATA[3]	Controls the rotational speed in lower order byte (8 bits)
DATA[4]	Actual torque current in higher order byte (8 bits)
DATA[5]	Actual torque current in lower order byte (8 bits)
DATA[6]	Motor temperature
DATA[7]	Null

Figure 12: CAN Frame for Data Received from C620

2.4 Tolerance Analysis

1. **Power Analysis:**

   Battery:

   We are using a Lithium Polymer battery to power our entire robot. Therefore, it is essential that the battery is powerful enough to support all electronics onboard, especially the three motors. The battery that we are using, DJI TB48S, has a rated power of 129.96 W and discharge rate of 10C, meaning that it can support a power consumption of 1299.6 W. The two DM-J4310-2EC motors each has a rated power of 60 W, and the M3508 motor has a rated power of 240 W. The total rated power of all motors are 360 W, so the the battery power is more than enough to support the entire system.

   DC-DC Module:

   The DC-DC module converts the 24V input down to a 3.3V output, to power the microcontroller which requires a voltage range of 2.0V-3.6V and a current of 50mA. This DC-DC module should operate with a output tolerance of $\pm 9\%$, ensuring stability and reliability in the system's power supply. Therefore, the DC-DC module can accept the voltage from 20V to 28V, and output 3.0-3.6V.

2. **IMU Analysis:**

   We want our IMU and motor(DM4310 and M3508) encoders to be updated frequently enough that we can have accurate enough data to update the torque of our leg motors. Ideally, the update rate of IMU is more than 200Hz and the maximum update rate of motor encoders is 1000Hz. For each data processing thread, the running time is from 50ms to 100ms. Therefore, for data processing of the microcontroller, IMU and encoders have sufficient update rate to supply the latest and accurate data to the microcontroller.

   Sensitivity tolerance requirement for the gyroscope:

   (a) RFS2000: $\pm 1\%$

   (b) Sensitivity Change over Temperature (TCS) for RFS2000: ± 0.03 %/K

   (c) Sensitivity Supply Voltage Drift (SVDD) for RFS2000: ≤ 0.4 %/V

   (d) Nonlinearity for RFS1000, RFS2000: ± 0.05 %FS

   (e) Zero-rate Offset for RFS2000: $\pm 1°/s$

   (f) Zero-rate Offset Change over Temperature (TCO) for RFS2000: $\pm 0.015°/s$ per K

   (g) Zero-rate Offset Supply Voltage Drift for RFS2000: $\leq 0.1°/s/V$

   Data rate tolerance:

   (a) $\pm 0.3\%$ for a data rate of 523Hz

   (b) $\pm 1\%$ for a data rate of 230Hz

3 Cost and Schedule

3.1 Cost Analysis

The total cost without shipping cost in Figure13 is \$351.65. By adding 10% sales tax and 7% shipment cost, the total cost for parts is \$411.43. For the labor cost, we expect we can have a salary of $\$40/hr * 2.5 * 65hr = \$6500$

per person. We have three team members. Therefore, the total labor cost should be $\$6500 * 3 = \$19500$. By adding the cost of the parts, the total cost of our project should be $\$19500 + \$411.43 = \$19911.43$.

Comment	Designator	Manufacturer	Quantity	Price
100nF Unpolarized capacitor	C1,C2,C3,C4	Walsin Tech Corp	4	0.5710
10nF Unpolarized capacitor	C5, C6	Walsin Tech Corp	2	0.0098
1uF Unpolarized capacitor	C7,C8,C9	Walsin Tech Corp	3	0.0267
10pF Unpolarized capacitor	C10,C11	Samsung Electro-Mechanics	2	0.0214
22uF Unpolarized capacitor	C12,C13	Murata Electronics	2	0.2103
100nF Unpolarized capacitor	C14,C15	Knowles	2	0.3441
RED Light emitting diode	D1	Hubei KENTO Elec	1	0.0051
120R Ferrite bead	FB1	Sunltech Tech	1	0.0104
Conn_01x04_Pin Generic connector	J1,J2,J4		3	0.1077
USB_C_Plug_USB2.0	P1	DEALON	1	0.0516
PMBT3904YS 200mA IC	Q1	Nexperia	1	0.0534
10k Resistor	R1,R6	Viking Tech	2	0.0569
1k5 Resistor	R2,R3,R4,R5	Viking Tech	4	0.0012
4k7 Resistor	R9	Viking Tech	1	0.0025
SW_SPDT Switch	SW1	DongGuan KINGTEK Industrial C	1	0.3409
SW_Push Push button switch	SW2	C&K	1	0.4437
STM32F103C8T6	U1	STMicroelectronics	1	1.4180
AMS1117-3.3	U2	Advanced Monolithic Systems	1	0.1399
MAX3051	U3	Analog Devices Inc./Maxim Inte	1	0.9100
BMI088	U4	Bosch Sensortec	1	2.9489
16MHz Four pin crystal	Y1	TXC Corp	1	0.1764
DM-J4310-2EC		DAMIAO Tech	2	164.3700
M3508		DJI	1	68.4900
C620 ESC		DJI	1	54.6500
100C 1000mAh 22.2V Lipo Battery		OVONIC	1	23.9900
XT60 Plug		AMASS	1	0.3032
HATCHBOX PLA PRO+ 3D Printer Filament		Hatchbox	1	28.0000
I2C 0.96-inch OLED Display		AZDelivery	1	4.0000
		Total:		351.6531

Figure 13: Bill of materials for the entire robot

3.2 Schedule

Week	Task	Person
2023/9/25-2023/10/01	Modify Project Proposal and Prepare for Design Document	Everyone
	Do wheeled motion(forward and backward) analysis	Gabriel
	Finish up PCB design	Jerry
	Complete mechanical structure modeling	Zehao
2023/10/02-2023/10/08	Prepare for Design Review	Everyone
	Do wheeled motion(rotation) analysis	Gabriel
	Pin Configurations in STM32CubeIDE	Jerry
	Testbench design and build, finish power-pcb design	Zehao
2023/10/09-2023/10/15	First Round of PCB Order	Everyone
	Do Leg motion(support) analysis	Gabriel
	Construct drivers for each component (Motors, IMU ...)	Jerry
	Assemble the mechanical parts, design and print out payload parts	Zehao
2023/10/16-2023/10/22	Second Round of PCB Order	Everyone
	Do Leg motion(jump) analysis	Gabriel
	Test functionality of individual motors with an existing development board & review PCB design	Jerry
	Connect the structure to the test bench and revise design	Zehao
2023/10/23-2023/10/29	Individual Process Report & Third Round of PCB Order	Everyone
	LQR regulator implementation	Gabriel
	Test PCB and make changes if necessary	Jerry
	Modeling a complete wheeled legged robot	Zehao
2023/10/30-2023/11/05	Check progress of everyone and help each other if needed	Everyone
	Implement VMC Control Algorithm	Gabriel
	Pin configuration and code deployment on PCB	Jerry
	Making parts and sending CAD files to factory	Zehao
2023/11/06-2023/11/12	Check progress of everyone and help each other if needed	Everyone
	Remote controller communication	Gabriel
	Integrate PCB into the robot	Jerry
	Assembling whole robot	Zehao
2023/11/13-2023/11/19	Mock Demo and Team Contract Fulfillment	Everyone
	Moter driver implementation	Gabriel
	Test and revise the PCB board	Jerry & Zehao
2023/11/20-2023/11/26	Integrating all parts and test the code functionality	Everyone
2023/11/27-2023/12/03	Final Demo	Everyone
2023/12/04-2023/12/07	Final Presentation and Final Paper	Everyone

Figure 14: Detailed Schedule

4 Ethics and Safety

4.1 Safety

Within our project, there are several safety hazards:

Firstly, if the wheeled-legged robot behaves unexpectedly or even losses control, it might cause harm to those around it. To avoid the unnecessary damages, we will implement two types(hardware and software) of emergency stops on our robots. For the hardware, we will install the wireless relay, which works like the emergency stop button, at the output of our battery to disable power to the motors and sensors. For the software, we will implement a safe thread and use a switch on our remote control to achieve a software emergency stop. If we turn on the switch, the safe thread will set the output of motors zero to stop all motors.

Another hazard is the usage of 24 Volt Lithium battery. Improper use of lithium batteries may cause them to expand, catch fire or even explode. In order to avoid accidents, will strictly adhere to the guidelines provided for the use of lithium batteries. We should check the appearance of the battery before using them, whether it is swollen or broken. We will use manufacturer-recommended charging methods, making sure that the load during use is less than the maximum discharge power of the battery, and constantly monitoring the battery temperature. In addition, during storage, the battery voltage will be adjusted to an appropriate storage voltage and keep them away from other flammable and explosive materials.

We will continue to update our safety manual if we become aware of or perceive any hidden safety hazards in the future.

4.2 Ethics

The structure of the wheel legs does not in itself present a ethical problem. However, when this structure is maliciously applied to some mobile robots, perhaps this will cause some ethical problems. Wheel-legged architectures can be built into automated mobile platforms that carry weapons or monitors. This would violate the **Three Laws of Robotics**[2] and **IEEE Code of Ethics**[3]. To avoid this situation, we will not be open-sourcing our core technical parts, such as PCB drawing, motion modeling, core code. At the same time, we will review our consumers(**avoiding malicious or illegal use**) and in the future will also apply our products in positive directions, such as urban logistics and transportation.

5 Citation

References

[1] DJI. C620 Manual. [Online]. Available: https://rm-static.djicdn.com/tem/17348/RoboMaster%20C620%20Brushless%20DC%20Motor%20Speed%20Controller%20V1.01.pdf

[2] Encyclopaedia, *Three Laws of Robotics*. Encyclopaedia Britannica, inc., 2023. [Online]. Available: https://www.britannica.com/topic/Three-Laws-of-Robotics

[3] IEEE. (2016) IEEE Code of Ethics. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html