Project Proposal

Introduction

Project Title: Fun Name for Robot
Team Number: 37
Team Members: Megha Esturi, Ishanvi Lakhani, Deepika Agrawal

PROBLEM :

In the battlebot competition, we need to meet the following constraints:

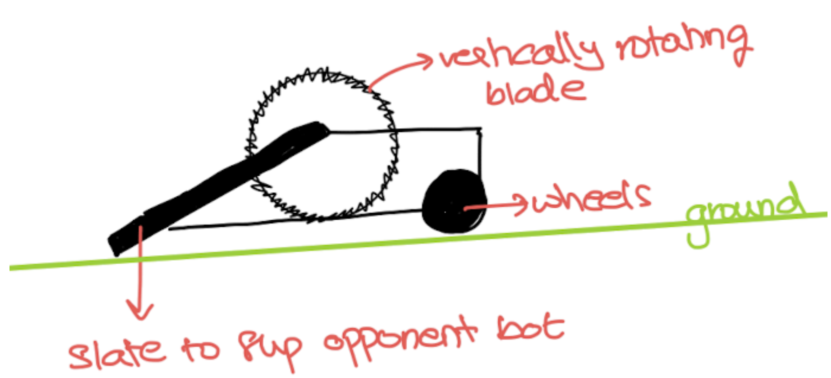
- Max weight 2 lbs
- 3D-printed materials
- Locomotion system and fighting tool
- Wireless control via Bluetooth or Wifi

SOLUTION :

Our goal is to create a battlebot that will defeat other bots without having our functionality disrupted. Our battle bot will have 2 vertically rotating circular blades on each side of it and will also have a slanted slate, which would flip the opponents battlebot if it got too close. The spinning of the blades and the flipping motion of the slate along with the normal movements of the battlebot using its wheels will all be via PC input controlled by the user. This communication will be established via WIFI.

VISUAL AID :

The following is a side view of what our battlebot is going to look like.

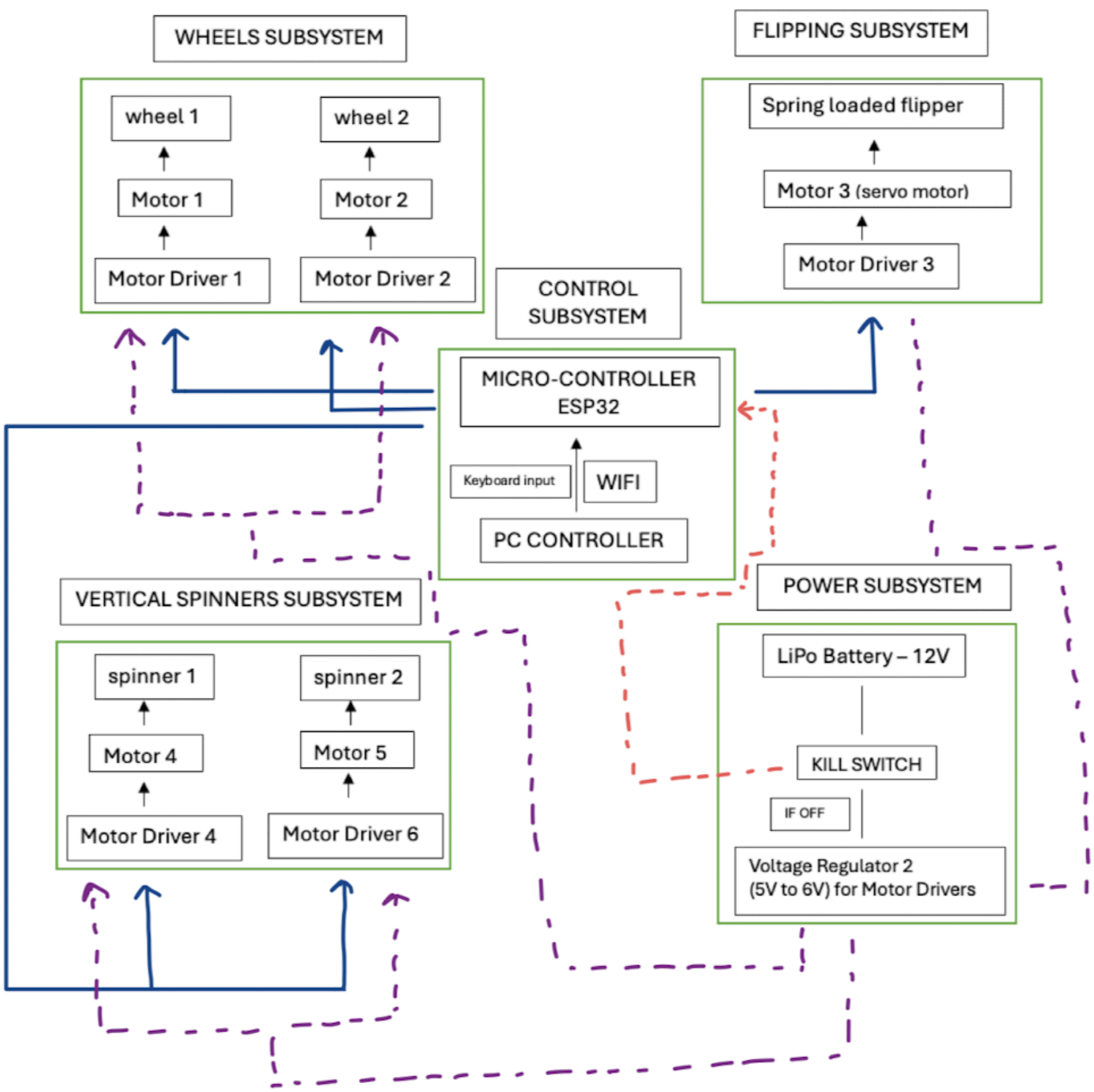


High-Level Requirements List

1. The total weight of the battle bot, including components such as the chassis, electronics, and weaponry, must not exceed 2 lbs.
2. The battle bot must operate efficiently within a voltage range of 12 volts to ensure consistent mobility and weapon operation throughout the match.
3. The battle bot should be able to communicate effectively using wifi and ensure communication is not lost during the battle.

Design

1. Block Diagram



2. Subsystem Overview

<u>WHEEL SUBSYSTEM</u>

This subsystem is responsible for the maneuverability of the battle bot. It consists of two wheels, each driven by a motor. The motors are controlled by motor drivers, which regulate the speed and direction based on signals from the microcontroller. This subsystem allows the bot to move forward, backward, and turn, enabling mobility during the battle. It is connected to the Microcontroller Subsystem, which receives commands from the PC controller to instruct the motor drivers on how to move the wheels.

FLIPPING SYSTEM

This subsystem is designed to perform the key function of flipping the opponent's battle bot. It uses a spring-loaded mechanism, powered by a servo motor, to execute this action. The motor is controlled by a motor driver, which regulates the flipping speed and movement. This subsystem is critical for offense in the battle, providing the bot with the ability to destabilize opponents. The flipping mechanism is triggered by the Microcontroller Subsystem, which receives user inputs from the PC controller and activates the flipper when needed.

VERTICAL SPINNER SUBSYSTEM

This subsystem consists of two rotating blades, located on opposite sides of the bot, designed to damage or destabilize the opponent. Both spinners are powered by a singular motor due to being connected with an axle, which is controlled by a motor driver. This subsystem operates offensively, using rotational force to cause damage. The Microcontroller Subsystem sends instructions to the motor drivers to control the speed and operation of the spinners, allowing the user to strategically engage or disengage the blades during the battle.

POWER SUBSYSTEM

This subsystem provides electrical power to all other subsystems. It is composed of a 12V LiPo battery, which supplies energy to the entire bot. The system includes two voltage regulators: Voltage Regulator 1 steps down the voltage for the ESP32 microcontroller, while Voltage Regulator 2 supplies the correct voltage to the motor drivers. This subsystem ensures that all components receive the required voltage levels to function efficiently. It is directly connected to every subsystem, as the motors, motor drivers, and the microcontroller all depend on a stable power supply to operate.

CONTROL SUBSYSTEM

This subsystem acts as the central brain of the battle bot. It receives commands from the PC Controller via WiFi and sends control signals to the motor drivers of the wheels, spinners, and flipper subsystems. The ESP32 processes user inputs and coordinates the actions of all the mechanical parts, ensuring that the bot performs as directed during the battle. It connects to the Power Subsystem for power and communicates with the Wheels Subsystem, Flipping Subsystem, and Vertical Spinners Subsystem to control their operation.

PC CONTROL SYSTEM

This subsystem serves as the user interface, allowing the operator to control the battle bot remotely. It sends keyboard inputs over WiFi to the ESP32 microcontroller, which processes these commands and relays them to the appropriate subsystems. This subsystem enables the real-time control of the bot's movement, spinners, and flipping mechanism. It is connected to the Microcontroller Subsystem via WiFi, establishing communication for bot control.

3. Subsystem Requirements

WHEEL SUBSYSTEM

We have 2 wheels. Each will be connected to N20 MicroGear Motors and each motor will be connected to a L298N dual channel motor driver. The operating voltage of the motor itself is between 3V and 12V and ideally it would run well at 6V so it provided good speed and torque. The RPM provided by it would be around 30 - 1000. The operating voltage of the motor driver is between 5V to 35V and so we would provide it with 6V since we already will have a voltage regulator making the voltage come down from 12 to 6V. Since we want each motor driver and motor and thereby wheel to get its own power supply, the motors etc will be connected in parallel to each other. The speed and motion of the wheels will change based on signals coming from the pulse width modulator. The microcontroller we are using (ESP32) already has inbuilt PWM capabilities which will allow it to effectively communicate with user input.

- Requirement 1 : The motor and motor drivers wheels must be able to communicate with the microcontroller effectively and relay the instructions given by the user.
- Requirement 2 : The motor and motor drivers should be able to work effectively with a supply of 6V and 100 - 200 mA when they are connected in parallel.
- Requirement 3 : The wheels should be able to allow the robot to move fast enough, i.e. rpm must be high approx. 150 - 300.

FLIPPING SYSTEM

We will have an arm that sits on top of the bot, and can flip upwards to flip the opponent robot. We will be using a Tower Pro MG995 Continuous Rotation Motor that can stop precisely after 360 degrees, typically operates at a voltage range of 4.8V to 7.2V, and weighs approximately 55 grams (0.12 lbs) . The motor can be controlled directly with a PWM signal from a microcontroller (the ESP-32 we plan to implement has one), which allows precise control over when and how much the motor rotates. We can program the servo to rotate the required 360°, compressing the spring, and then stop. When the flip is triggered, the spring would release, performing the flipping action. The spring we intend to use is a compression spring, specifically the uxcell Steel Coil Extended Compressed Spring. Once the action is complete, the motor could reset to its original position and be ready for the next trigger.

- Requirement 1 (Power Supply): Our power source should be stable, and should be able to provide the necessary voltage (4.8V to 7.2V) and enough current to drive the Tower Pro MG995 Motor
- Requirement 2 (Control System): Our ESP-32 will act as the control system for our motor, because it will provide precise PWM signals to the servo motor. This includes

programming the timing and angles for both the compression of the spring and the reset position after the flipping occurs.

- Requirement 3 (Design and Assembly): We will need to securely attach the arm to the top of the robot and properly position the compression spring under the arm. We will mount the spring in a way that it can best store and release energy, as well as the motor in a way that we will achieve the desired flipping motion.

VERTICAL SPINNER SUBSYSTEM

We have two vertical spinner systems connected to each other with the help of an axle. Therefore, we only need one motor and motor driver to control the movement of the axel. We will be using a Mini RS-380 Motor DC 6V. We will use the same motor driver that we used for the wheels and use L298N and connect that. Both the motor and motor driver should be able to operate at 6V.

- Requirement 1 : The motor and motor driver must be able to communicate with the microcontroller effectively with a maximum of a few ms of delay.
- Requirement 2 : The motor and motor drivers should be able to work effectively with a supply of 6V and 300mA to 1A when they are connected in parallel.
- Requirement 3 : The vertical spinners should move fast enough so the defense mechanism is strong, i.e. rpm must be high approx. 1500 - 2000.

POWER SUBSYSTEM

The power subsystem will consist of a 12V LiPo battery with one voltage regulator. The ESP32 has an inbuilt voltage regulator, so it will get fed 12V. The voltage regulator we will create is going to bring the voltage down to around 5V for the spinners, wheels, and the flipper.

- Requirement 1: The subsystem must include a 12V LiPo battery with enough power for the duration of the battle and a capacity of at least 1000mAh depending on the power consumption.
- Requirement 2: The subsystem must include a voltage regulator and it must provide a stable voltage of 5V.
- Requirements 3: The voltage regulator must be capable of supplying at least 3A of continuous output.
- Requirement 4: There will be a kill switch that will be manually triggered by the PC to cut off all power in case of a hazard

CONTROL SUBSYSTEM

The microcontroller we have chosen is the ESP32-WROOM-32 with an integrated 2.4 GHz wi-fi module. Because the wifi is integrated and there is an integrated antennae, we will not need a UART or SPI. Additionally, this microcontroller has low power consumption and allows for multiple tasks to happen simultaneously which we would need for the flipper, spinners, and wheel control.

- Requirement 1: The microcontroller must be able to support WiFi communication with a throughput of at LEAST 5 Mbps for reliable communication to the wheels and the weapon systems.

- Requirement 2: It must provide a minimum of 6 GPIO pins for digital I/O to interface with the two wheels, vertical blades, and the flipping mechanism.
    - 2 pins for motor control
    - 2 PWM pins for speed control of the wheels
    - 1 pin for motor control of vertical spinners
    - 1 pin for motor control of the flipping mechanism
- Requirement 3: The input voltage will be between 5V and 12V and will be regulated down to it's operating voltage of 3.3V
- Requirement 4: Communication latency must be less than 50 ms to ensure responsiveness
- Requirements 5: The PC software should support bidirectional communication allowing for control commands to be sent to the microcontroller.
- Requirement 6: The PC to microcontroller will communicate through a WiFi access point with a TCP client

Tolerance Analysis

The subsystem that poses the greatest risk is the vertical spinner subsystem due to these factors:

- High RPM: Since the spinners will be at a relatively high RPM, it will operate at high speeds generating a lot of energy. If there was a malfunction, the spinners could fly off and cause injury or de-balance the bot. If the spinners cannot handle the stress of the rotation, then it could also lead to a failure.
- Electrical Failure: If there is any electrical damage, there could be a chance of a fire which in turn could damage the robot also.
- Overheating: Since the spinner is spinning at a high RPM, this could cause overheating. If the motor driver is not ventilated well, the motor could fail.
- Axle Fatigue: Since the parts are made of plastic and there is so much rotation, so the axle could weaken or snap.

RPM Tolerance Analysis

- w = (RPM*2pi)/60

For 1500 RPM

- w = (1500*2pi)/60 = 157.08 rad/s

For 2000 RPM

- w = (2000*2pi)/60 = 209.44 rad/s

We can set a tolerance for +- 100 RPM

- RPM Range: 1400 RPM to 2100 RPM
- Angular Velocity Range: 146.61 rad/s to 219.91 rad/s

Ethics

There are a few ethical considerations we have to keep in mind throughout this project. The first is prioritizing the safety and welfare for all participants and will comply with safety standards to minimize risks. Further, we will use sustainable materials in our design, and disclose any potential risks while building our battlebot. We will treat all team members and competitors with respect, avoiding discrimination and harassment. Finally, we will ensure we have the necessary skills and seek help when needed, and make sure that all of our work is our own, and we are not unfairly plagiarizing others.

Safety

The primary risks in building our battlebot include injuries from the spinning blades, the flipping mechanism, and potential hazards from high-speed motors and LiPo batteries. To address these risks, we will use protective enclosures, secure battery management, and limit motor speed and torque to safe levels. Additionally, all team members have completed lab safety training and will follow these guidelines to make sure we are safely using the lab equipment. We will also follow university safety policies regarding lab work and equipment handling.

Citations and References

ESP_WROOM_LINK
ESP_WROOM_32E Datasheet
ESP_WROOM_32 INFO

DC Gear Motor → Wheels
N20 Gear Motors
N20 Gear Motor Datasheet
Dual H-Bridge Motor Driver Datasheet
Brushed High Speed DC Motor

Servo Motor MG995 360 Degree Continuous Rotation - ProtoSupplies → Flipping Mechanism
Servo Motor Vs DC Motor: What are the Main Differences?
Compression Springs vs. Tension Springs: A Guide to Choosing the Right One - Chaoyi Spring
A Deep Dive into Spring Types: The Mechanics of Compression Springs