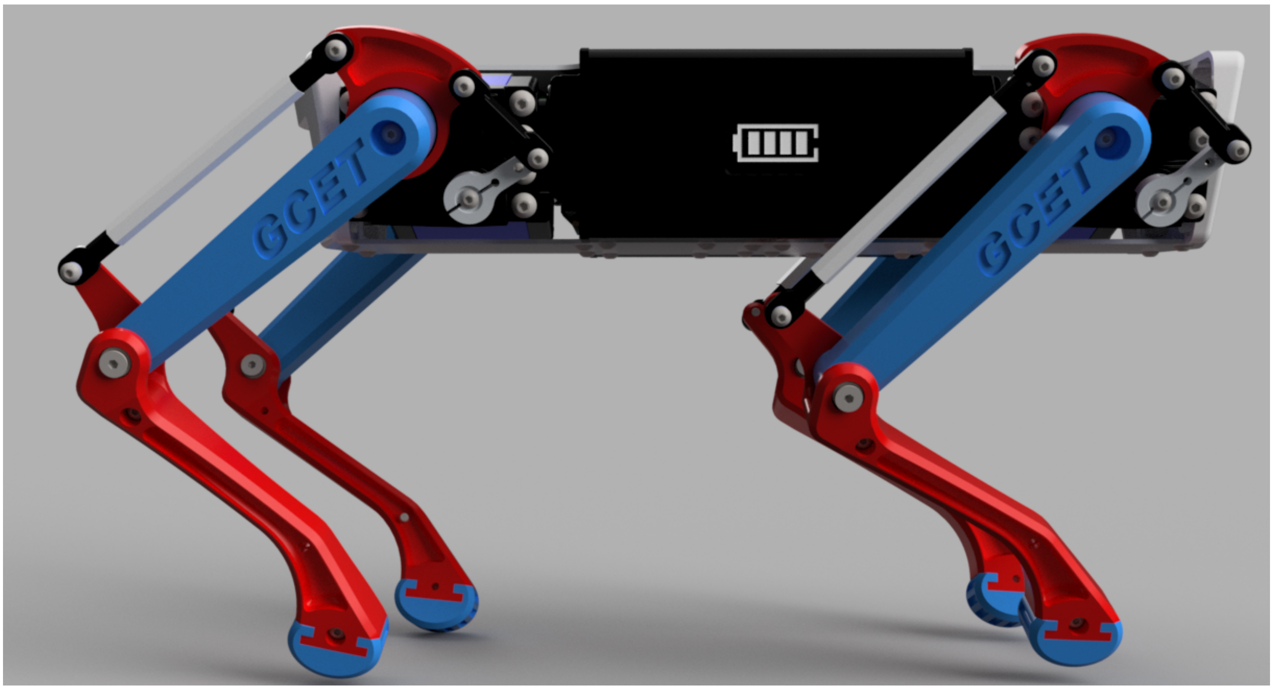


Four Legged Robot with Quadrupedal Motion

Prepared for: Proof Of Concept Stage, Robofest 2.0

Prepared by: Team GCET

Team Details

College Name : G.H. Patel College of Engineering and Technology.

Address: Bakrol Road, Mota Bazzar, Vallabh-Vidyanagar,
Anand-388120.

Mentor: Dr. Ajay Patel.
Mentor Email: ajaypatel@gcet.ac.in
Mentor Mobile No.: +91 9427037011

Team Members:

Sr. No	Name of Student/ Research scholar	Photo	Department	Mobile No.
1	Anmol Shah		Mechatronics	+91 9925512860
2	Shubham Sharma		Mechatronics	+91 6354357902
3	Aviral Sharma		Electronics & Communication	+91 7574072740
4	Gurmann Singh Bhullar		Mechatronics	+91 8153066213
5	Kunal Rajpurohit		Mechanical	+91 7733996753

Abstract

"When people Love what they do, what they do is ***Amazing.***"

This Report introduces robust, dynamic quadruped designed by team GCET. The terrain over which, these robots must be able to move is often uneven, slippery or muddy, which gives rise to many challenges, particularly stability. Two of the most common methods for robot mobility are wheels and legs. The necessity of legged robots is increasing day by day, as compared to the wheeled robots. Legged robots are more advantageous and versatile than wheeled robots, on uneven terrain such as military operations, remote locations, dangerous environments, excavation and construction works and medical applications.This paper presents, design of a quadruped robot having four legs, whose design is derived from Animal structure. An attempt was also made, to perform inverse kinematic analysis on the design for validation.

INDEX

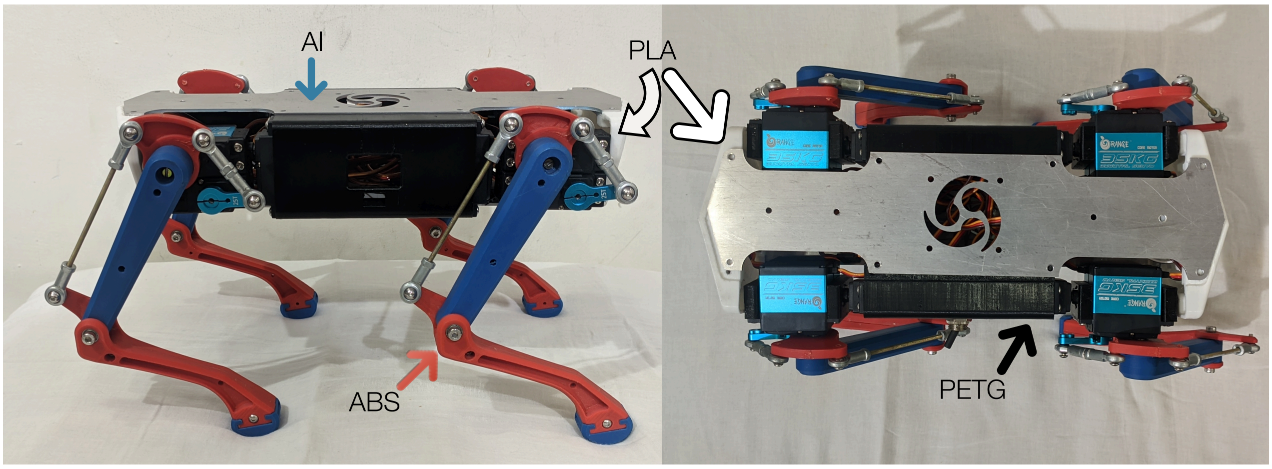
DESIGN

1.OVERVIEW

Parameter	Symbol	Value	Unit
Mass	M	2.40	Kg
Body Length	L_{body}	306	mm
Body Width	W_{body}	185	mm
Body Height	H_{body}	185	mm
Leg Link Length	L_1, L_2	100	mm

The Body is designed specifically in a way so as to adapt to different environments quickly. It is thoughtfully engineered so as to achieve maximum motions with minimum mechanical hinderances, therefore achieving fullest work volume for our Quadruped Robot. We have chosen 3D printing to create the body structure so as to perfectly achieve the required form. In 3D printing, we have used 3 Materials - PETG, ABS, and PLA. PLA is used in Body, Face, and Back Shield as these parts experience less stress while ABS is used for Leg Modules as they require high load-bearing capacity. The Servo mountings for hip moments are embedded in body structure only therefore we have used PETG material for higher durability. The Linkages used are of Stainless Steel for its strength and corrosion resistance.

Material	Property	Used
Aluminium (Al)	Light Weight , Strength	Upper & Lower Cover
Polylactic Acid (PLA)	High Surface Energy	Body Face and Back Shield
Acrylonitrile butadiene styrene (ABS)	High Strength	Leg Modules
Polyethylene terephthalate (PETG)	High Durability	Body Structure
Stainless Steel (SS)	Non-Corrosive , Strength	Linkages



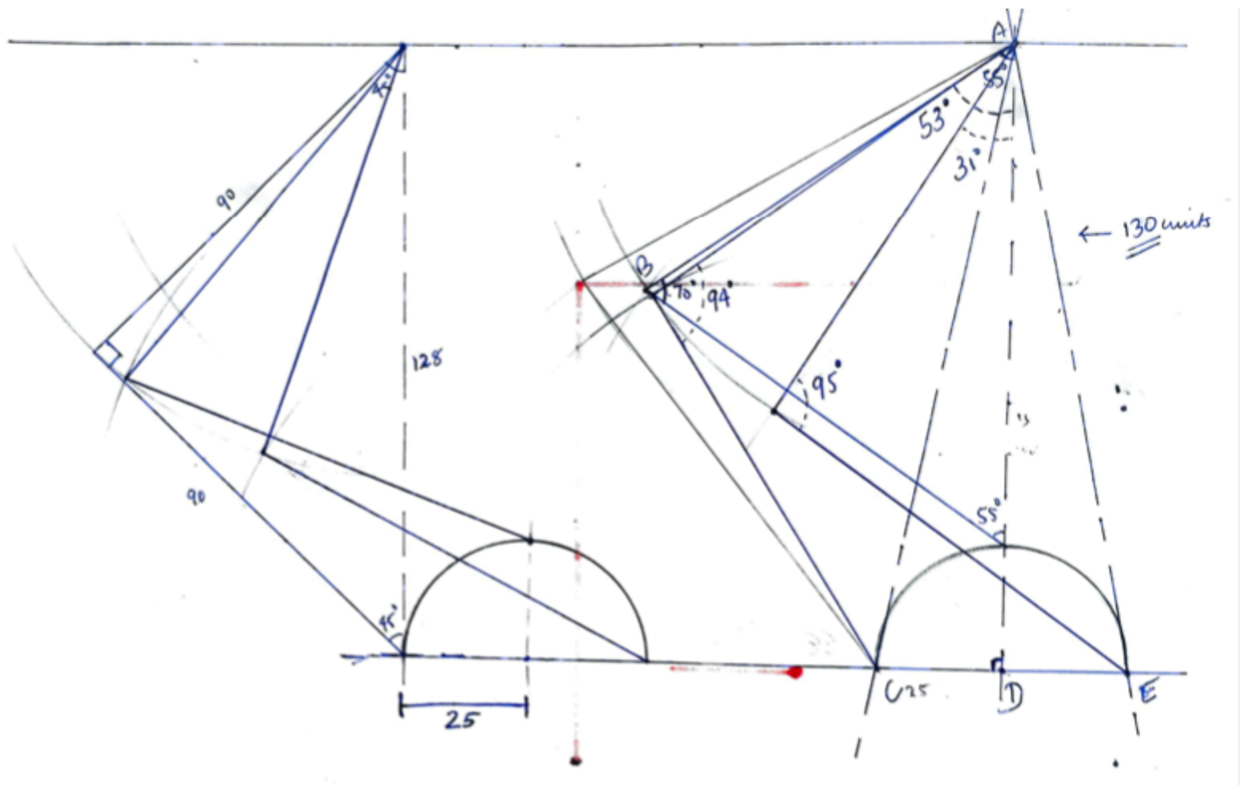
Side View (Actual Photograph)

Top View (Actual Photograph)

2.LEG DESIGN AND ACTUATION

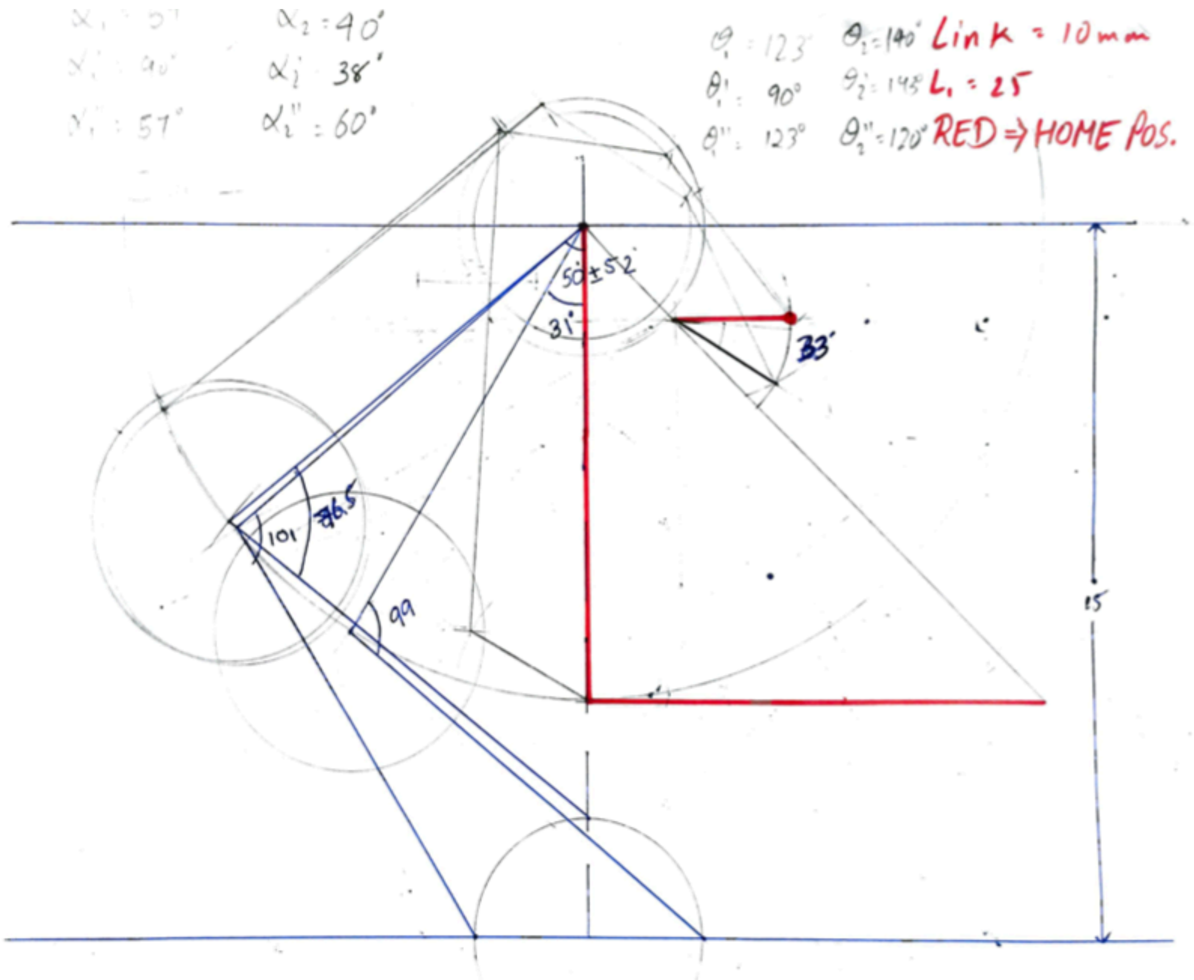
Trajectory of the Quadruped Robot is calculated using the approach angle and the departure angle in a semi-circular path.Here the length of leg is constant and the approach angle is variable with respect to height of the bot.

Approach Angle*	Intermediate Angle*	Departure Angle*
A'=53	A"=55	A'"=31
B'=94	B"=70	B'"=95
All this approach angles are in respect to Protocol Height i.e. 130 mm *All Angles are in degrees.		

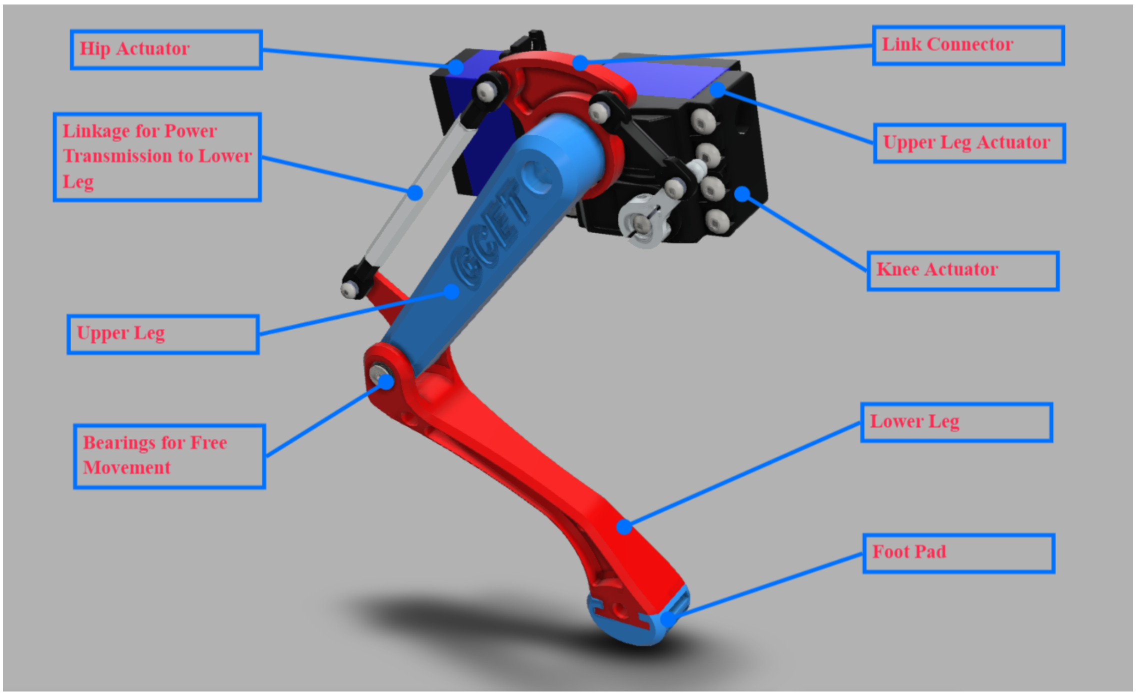


Leg Trajectory

Here the approach angle form a complete isosceles triangle with the displacement of a half offset of semi-circular arc and all the angles are calculated using the trigonometric functions. And the trajectory is calculated by simultaneous control of linkages and the PID loop for balancing the robot .

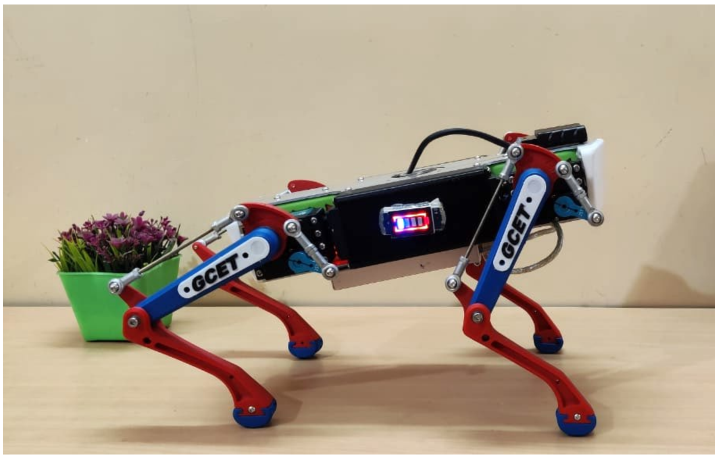


Linkage Trajectory



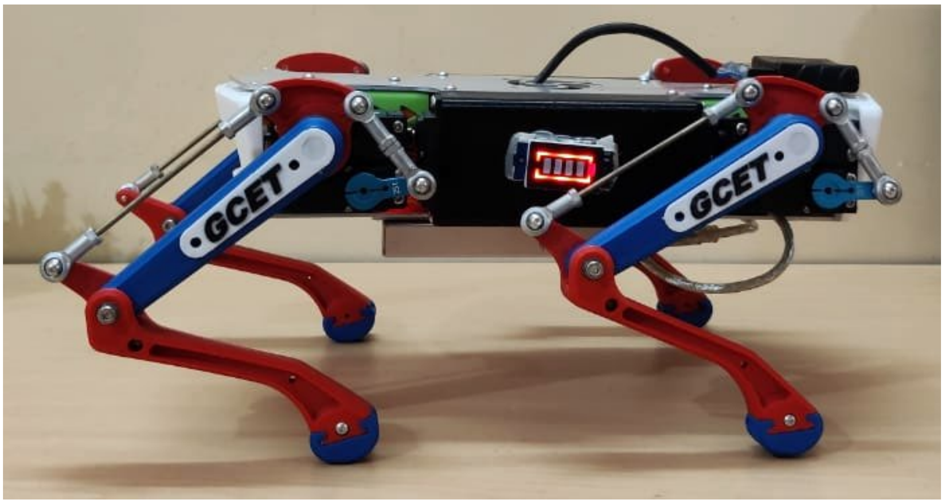
Leg Design- with all its 3 actuators

Seek



Pitch

Using this specially designed algorithm we developed a code that commands the robot to Seek or Pitch the Quadruped with minimal hindrances at a desired height without disturbing the balance of the Quadruped.



Home Position

Varied Height

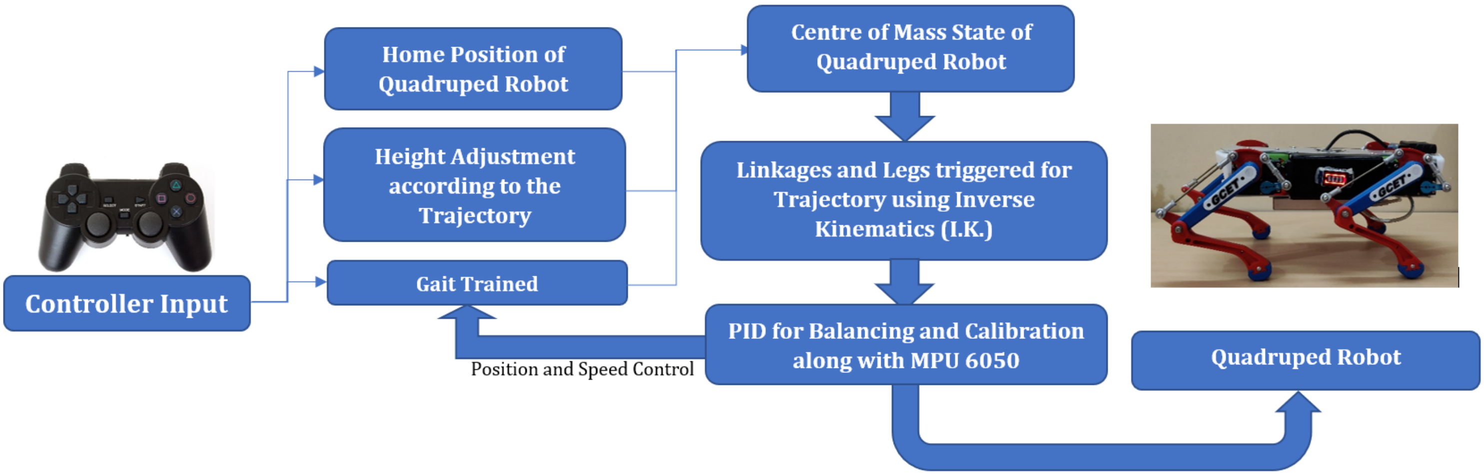
3.CONTROL ARCHITECTURE

GAIT PLANNING

The Quadruped's gait is defined by an event-based finite state machine that uses a leg-independent phase variable to schedule nominal contact and swing phases.This allows for flexible gait definitions and fluid transitions between them.Trotting gaits are presented in this report, but the framework allows implementation of any gait definition with ease. The gaits were designed to mimic natural animal gaits by controlling the independent phases of each leg.This nominal gait plan is modified during unexpected contact events on the legs.

SYSTEM BLOCK DIAGRAM

Here the primary loop for the quadruped is for its most stable state by Centre of Mass, and then the trajectory is calculated based on the Centre of Mass by using the method of INVERSE KINEMATICS. Input is taken from the CONTROLLER and the specific trajectory/function is triggered for the protocol position and height of the robot. The height of the robot can be adjusted and the further angles and trajectory is automatically calculated using INVERSE KINEMATICS .Feedback loop plays significant role for the stability of the robot as the GAIT training is used to achieve more precise and stable movements with flexibility for multiple gestures.



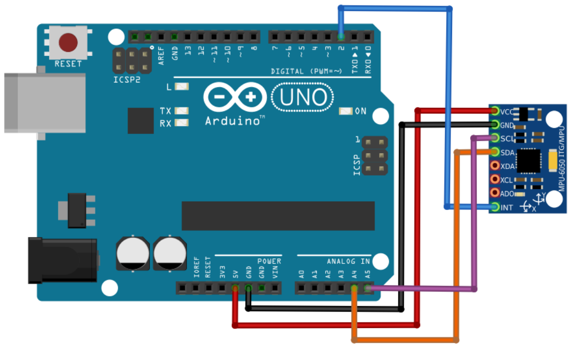
CONTROL SYSTEM

Arduino Uno board is used as main microcontroller to process and communicate with all the PWM signals using SCA and SCL ie. I2C pins on the Arduino board.

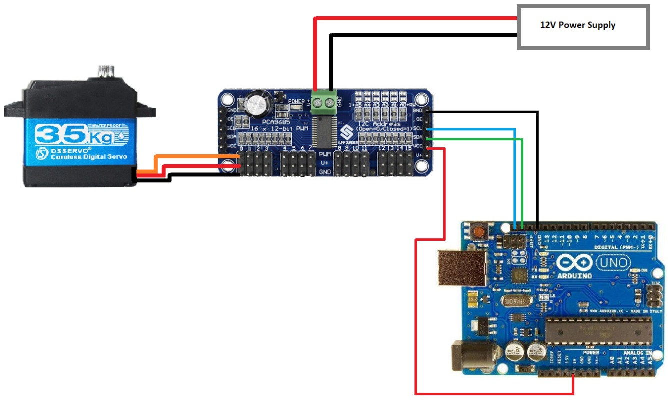
The main receiver is used with combination of a transmitter remote, and every action from transmitter remote acts as a trigger function for a specific combination of trajectory formed by using inverse kinematics.

PID control is used in combination of trajectory for the proper and balanced walking.

The MPU6050 is used to sense the balance in the PID loop that give feedback for every instance of time and also help the quadruped to change the height without loosing stability.

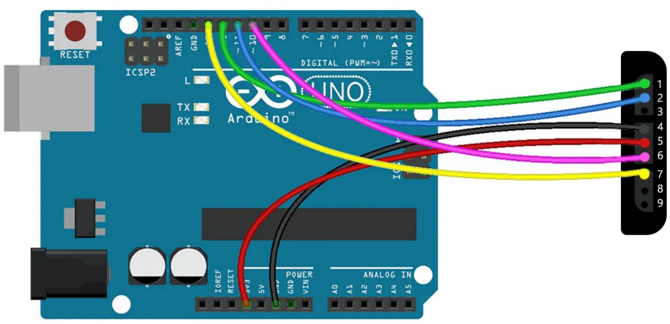


MPU6050 Connection



PCA9685_Circuit

The key feature of the MPU5060 in PID loop is that apart from providing stability while walking this also helps the quadruped to increase or decrease the height and the relative height can be changed without changing the trajectory while walking and also provide flexibility for the quadruped robot to automatically compute the angles of legs and linkages.



Micro-Controller Connections With PS2 Receiver

PS2 Transmitter and Receiver

Components specification

☑ Orange FT5330M 35kgcm

The Servo motor is specially made for robotics with custom design. Also designed to work on 8.4V supply voltage. The motor provides massive torque of 35kg-cm.

Operating Voltage: 6~8.4V

Operating Temperature range: -10C~50C

Operating frequency: 50-330Hz

Neutral Position: 1500 usec

Dead bandwidth: 3 usec

Pulse Width range from: 500 to 2500 usec

Electrical Specifications:	7.2V	8.4V
Operating speed (at no load)	0.13sec/ 60°	0.10sec/60°
Running current (at no load)	250mA	300mA
Stall torque (at lock)	31kg.cm	35kg.cm
Stall current (at lock)	3.2A	3.5A
Idle current (at stopped)	4mA	5mA

☑ Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller.The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits.

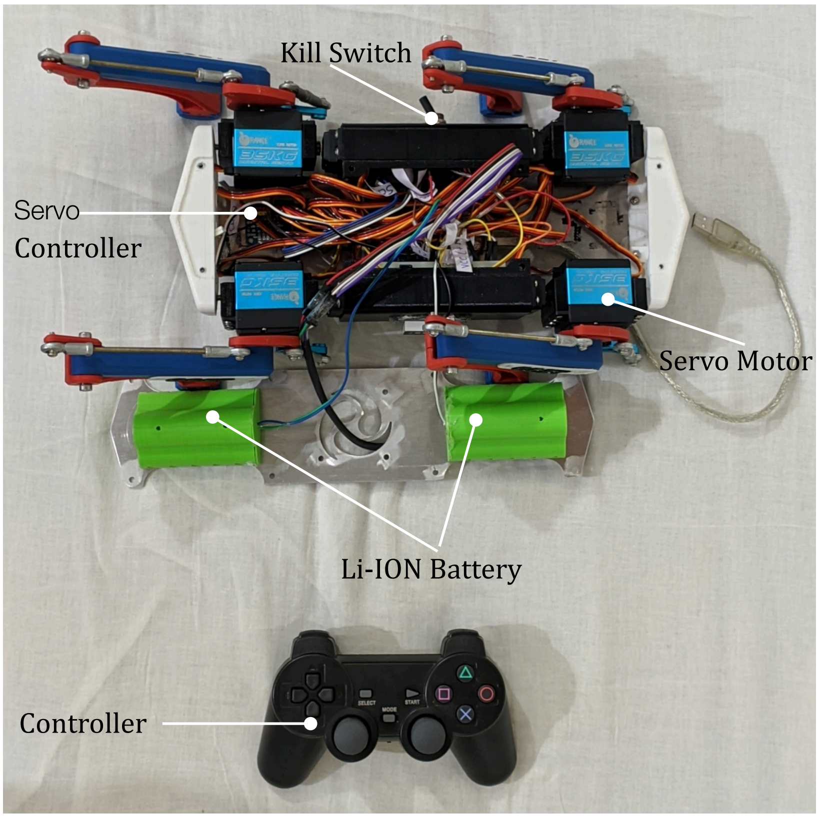
Microcontroller: Microchip ATmega328P
Operating Voltage: 5 Volts
Input Voltage: 7 to 20 Volts
Digital I/O Pins: 14 (of which 6 can provide PWM output) UART: 1
SPPI: 1 Analog Input Pins: 6
DC Current per I/O Pin: 20 mA
DC Current for 3.3V Pin: 50 mA
Flash Memory: 32 KB of which 0.5 KB used by bootloader
SRAM: 2 KB
EEPROM: 1 KB
Clock Speed: 16 MHz

- **16-Channel 12-bit PWM/Servo Driver I2C interface PCA9685 for Arduino Raspberry Pi**

You can control 16 free-running PWM outputs with 16-Channel 12-Bit PWM/Servo Driver using only 2 pins. You can even chain up 62 breakouts to control up to 992 PWM outputs.
It's an I2C-controlled PWM driver with a built-in clock. That means you do not need to continuously send it to signal to tie up your microcontroller, it is completely free running!
It is 5V compliant, which means you can control it from a 3.3V microcontroller and still safely drive up to 6V outputs. This is good when you want to control white or blue LEDs with 3.4+ forward voltages. 6 address select pins so you can wire up to 62 of these on a single I2C bus, a total of 992 outputs.

- **MPU-6050**

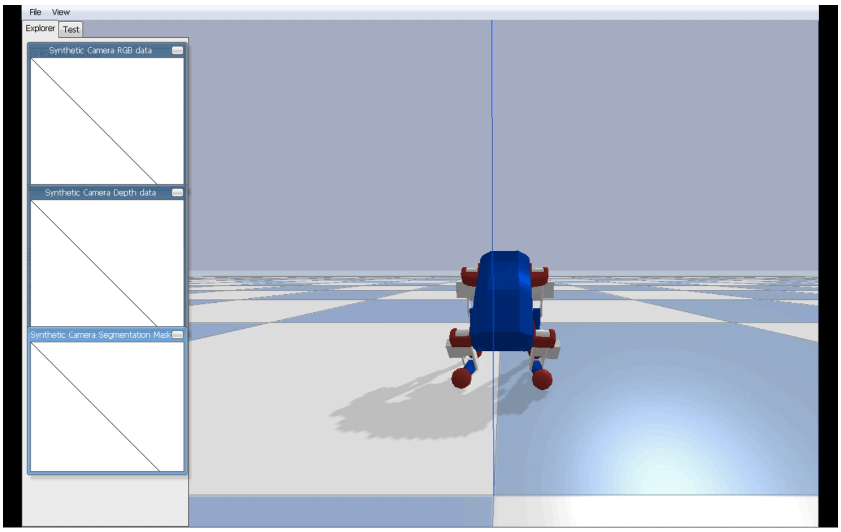
The MPU-6050 devices combine a 3-axis gyroscope and a 3-axis accelerometer on the same silicon die, together with an onboard Digital Motion Processor™ (DMP™), which processes complex 6-axis MotionFusion algorithms. The device can access external magnetometers or other sensors through an auxiliary master I^2C bus, allowing the devices to gather a full set of sensor data without intervention from the system processor. The devices are offered in a 4 mm x 4 mm x 0.9 mm QFN package.



Detached Upper Cover

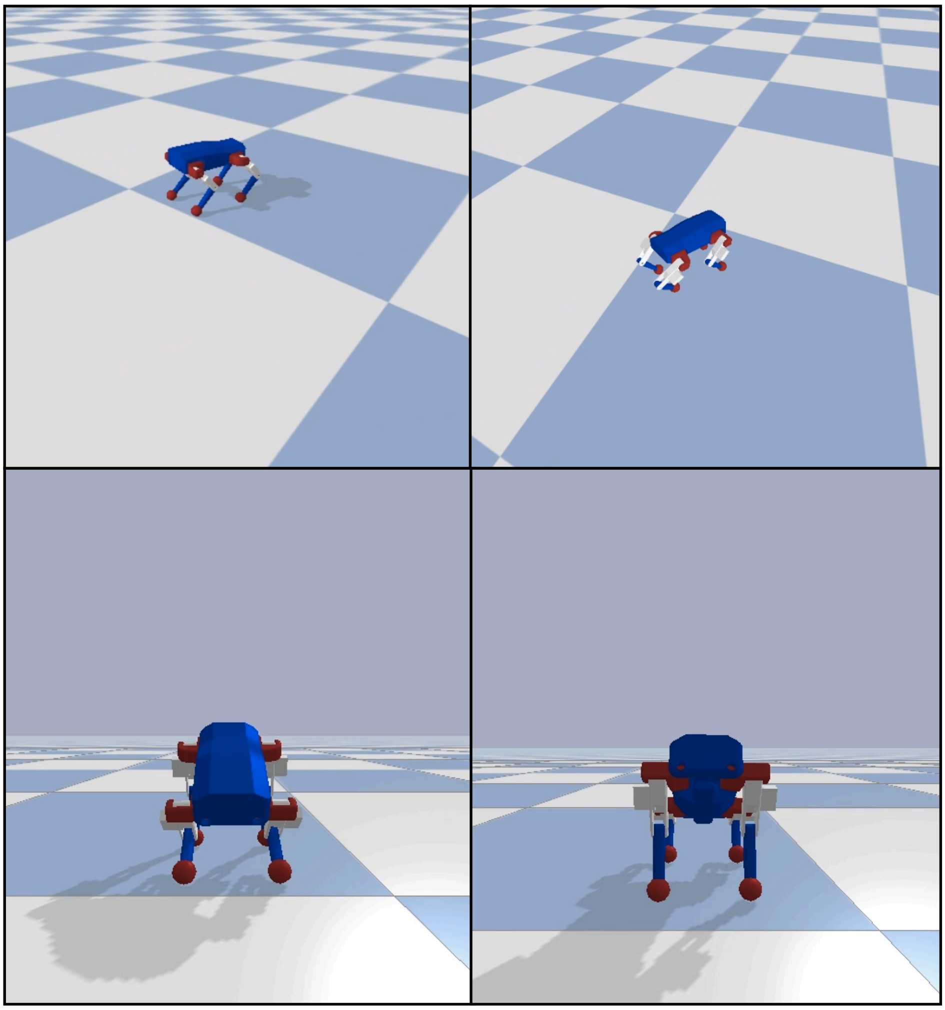
SIMULATIONS USING PYBULLET

The Python wrapper of the Bullet3 SDK- will be used as the default Physics engine in the entire project to run all the simulations and to train the quadruped in a virtual environment.This would ensure that all the parameters regarding the working of the robot are checked however many times we want before implementing them on the real robot.We have performed gait training and simulations on this software.



PyBullet User Interface

Simulations are done in PyBullet software. For this, we prepared a '*URDF*' file(i.e. Universal Robot Description Format) of the robot by allocating its joints and angle by which they can be rotated.Then the robot is imported into the PyBullet physic engine and necessary calibrations are done.After calibration, different movements are tested.For example Movement in X, Y, Z direction, pitch, roll, yaw, trotting, and walking.Then, the most desired movement, for example walking is iterated till the perfect movements are achieved.Once the perfect motions are achieved the data of the angle variance with respect to the required motion are extracted into a file and the data is added to the microcontroller's code.This data helps robot to run with perfect motions in real world and it saves robot from accidents thus saving time and money from wasted due to failing of the components.

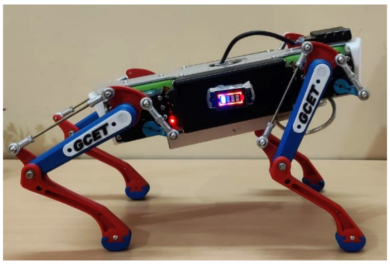


Simulations in PyBullet

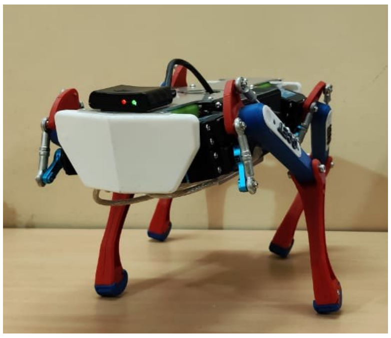
Result

There are total 6 different types of gestures and total 11 types of movements our robot can perform with ease:

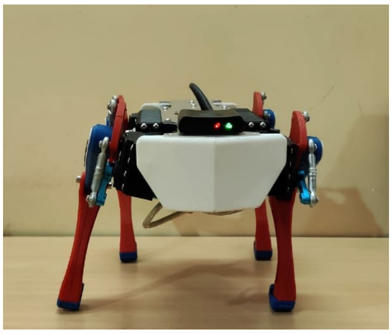
Seek



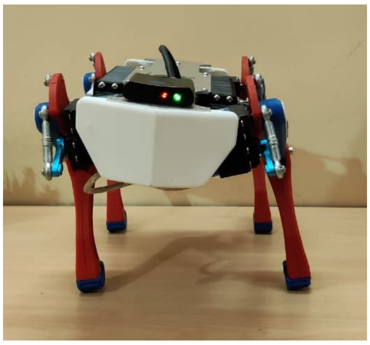
Pitch



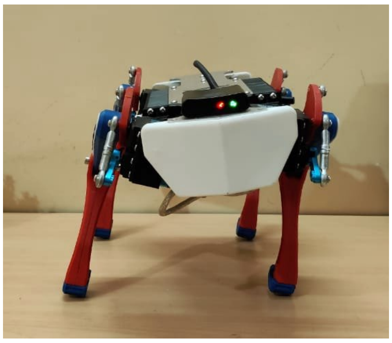
Yaw-Left



Yaw-Right



Roll-Left



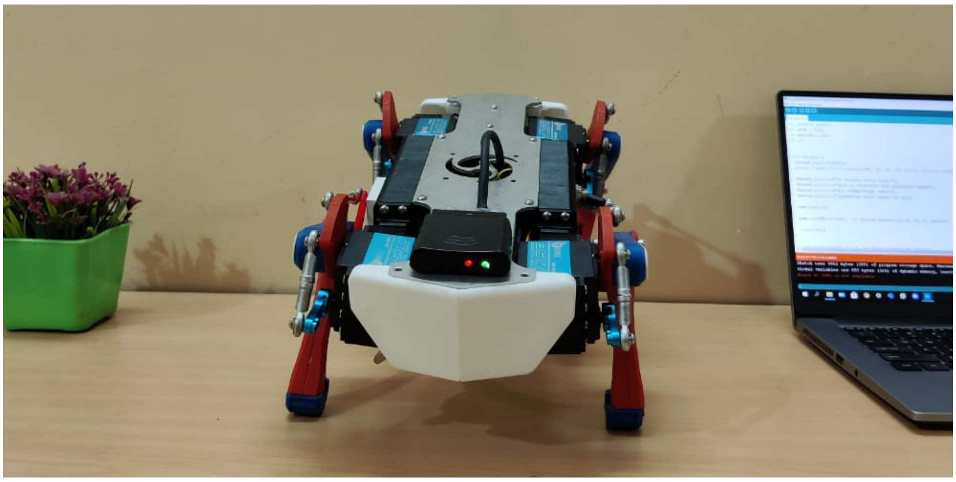
Roll-Right

Conclusion & Road Map

In this stage, we worked upon using optimum resources, so that we can diversify the use of funds.We used different material for different body parts according to the requirement.We learned to implement PID control and MPU 6050 for balancing and calibration of the robot.We used forward Kinematics and inverse kinematics for training the robot with different gaits for motion.

In the near future, we plan advance this concept through the use of Deep Learning and by using better hardware. The Primary upgradation on our Quadruped Robot would be its actuators; In the newer version, we've planned to substitute Servo Motors with BLDC motors attached with a custom-designed gear box in combination with an optical encoder that will help in the position and speed control. The custom-designed gearbox for the BLDC will enhance the control and manage the stiffness of the movement of the joint, thus achieving higher accuracy. Also this will give the quadruped the function to control the damping of the joints and links, resulting in more stability. It also would provide the capability to resist the shocks and jerks faced while working in unforeseen environments.

All over this journey till this stage, we have been through enormous learn and grow odyssey and we are very delightful to be a part of "*GUJCOST Robofest 2.0*".Surely we are looking forward to enhance our product and make it worth winning the competition.



THANK YOU