Scientific Review
ISSN(e): 2412-2599, ISSN(p): 2413-8835
Vol. 6, Issue. 2, pp: 14-22, 2020
URL: https://arpgweb.com/journal/journal/10
DOI: https://doi.org/10.32861/sr.62.14.22

Academic Research Publishing Group

Original Research Open Access

Design of Quad-Wheeled Robot for Multi-Terrain Navigation

Mbadiwe S. Benyeogor (Corresponding Author)
Department of Electrical and Electronic Engineering, University of Ibadan, Ibadan, Nigeria
Department of Physics and Electrical Engineering, University of Bremen, Bremen, Germany
Email: samrexbenzil@gmail.com

Oladayo O. Olakanmi
Department of Electrical and Electronic Engineering, University of Ibadan, Ibadan, Nigeria

Sushant Kumar
Department of Physics and Electrical Engineering, University of Bremen, Bremen, Germany

Article History
Received: January 15, 2020
Revised: February 9, 2020
Accepted: February 17, 2020
Published: February 19, 2020

Abstract

Wheeled robots are often utilized for various remote sensing and telerobotic applications because of their ability to navigate through dynamic environments, mostly under the partial control of a human operator. To make these robots capable to traverse through terrains of rough and uneven topography, their driving mechanisms and controllers must be very efficient at producing and controlling large mechanical power with great precision in real-time, however small the robot may be. This paper discusses an approach for designing a quad-wheeled robot, which is wirelessly controlled with a personal computer (PC) by medium-range radio frequency *(RF)* transceiver, to navigate through unpaved paths with little or no difficulty. An efficient servo-controlled Ackerman steering mechanism and a high-torque driving power-train were developed. The robot's controller is programmed to receive and respond to *RF* control signals from the PC to perform the desired motions. The dynamics of the robot's drivetrain is modeled and analyzed on MATLAB to predict its performances. The robot was tested on various topographies to determine its physical capabilities. Results show that the robot is capable of non-holonomically constrained motions on rough and uneven terrains.

Keywords: Drive-train; Controllers; Servo; Teleoperation; Topography; Robot.

1. Introduction

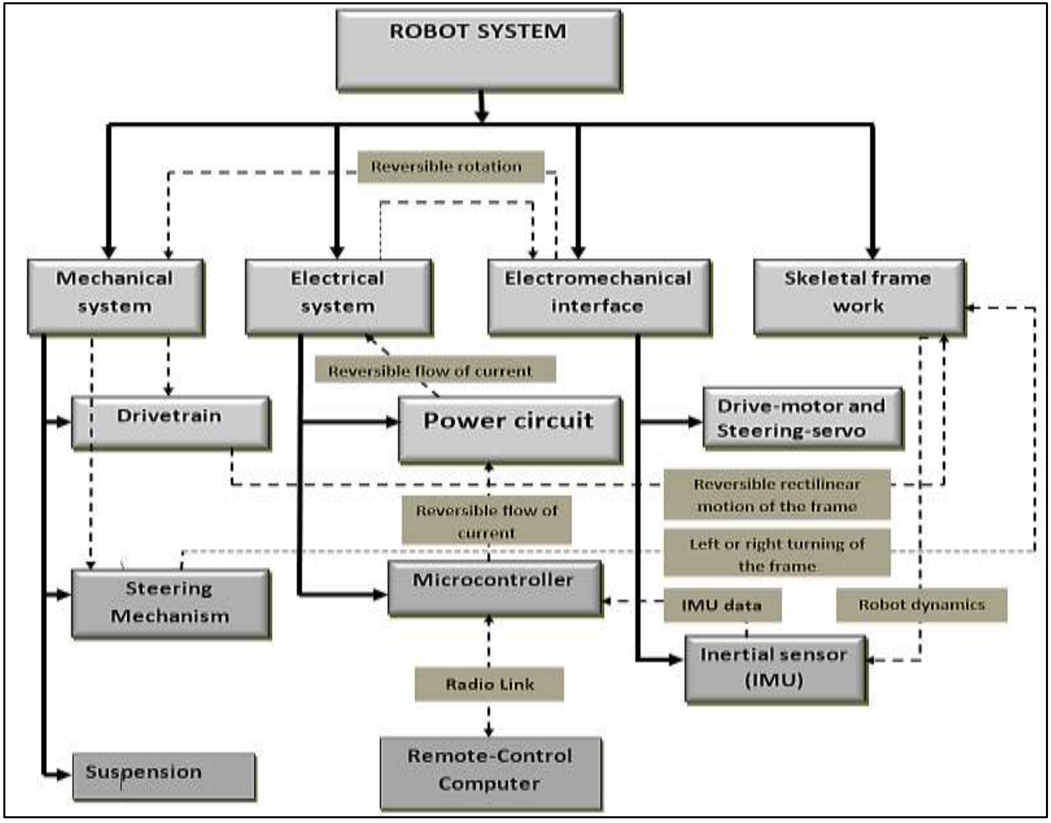
The current trend in robotics is the development of semi-autonomous rovers that can navigate around a geographical location. For long, this has been of great research interest. However, it is observed that many ground robots especially, those that navigate on wheels face physical challenges on entering terrains of uneven topography. These include unpaved roads, muddy grounds, stony paths, and hilly terranes. In the past, several researchers have made attempts at developing Wheeled Mobile Robots (WMR) that can climb stones and move in unstructured environments. Although the development of these robots has been inspired by planetary exploration [1]. For earth-based applications, WMR is developed/made to navigate around dynamic environments like homes and farms. The Rover-1 of the Robotics Institute at Carnegie-Mellon University (CMU) is an example [2]. This robot can effortlessly climb ridges by altering its center of gravity. The Shrimp which is an all-terrain six-wheeled robot is another notable obstacle climber that owes its passive climbing abilities to its geometrical structure and how its center of gravity is adjusted to its wheels over time [3]. To develop an efficient WMR with the capabilities to run on rough grounds, its power-train must be able to produce enough torque and traction at the driving wheels, its steering mechanism must accurately orient the robot in any given direction, while its controller must respond actively in accordance to software instructions to control the mentioned processes in real-time. In our work, a four-wheeled robot capable of driving through rough terrains is developed. This work is an improvement on its precursor – a simple locomotor with a primitive controller, which was developed by Mbadiwe and Bonde in Benyeogor and Danladi [4]; to demonstrate the deployment of geophysics instruments to remote locations.

2. Material and Methods

2.1. System Model

The system of our robot comprises some subsystems that are integrated to work together as a functional unit. The block diagram shown in Fig.1 is a simple model describing this robotic system.

Fig-1. Model of the robot system in block diagram



2.2. Design and Modelling of Drivetrain

The robot is mechanically propelled by a speed-reduction gear-train that is driven by an internally geared 12-volt 7.2-Amp DC motor that rotates at 120 rpm (no load) as shown in Fig.2. This transmits equal power to the two driving tires (i.e. wheels) at the rear side of the robot. This functional unit is fastened to the robot's chassis. Its Gear Ratio (G.R.) is evaluated as in eq.1.

$$G.R = \frac{n_F}{n_B} = \frac{24}{8} = 3 \qquad (1)$$

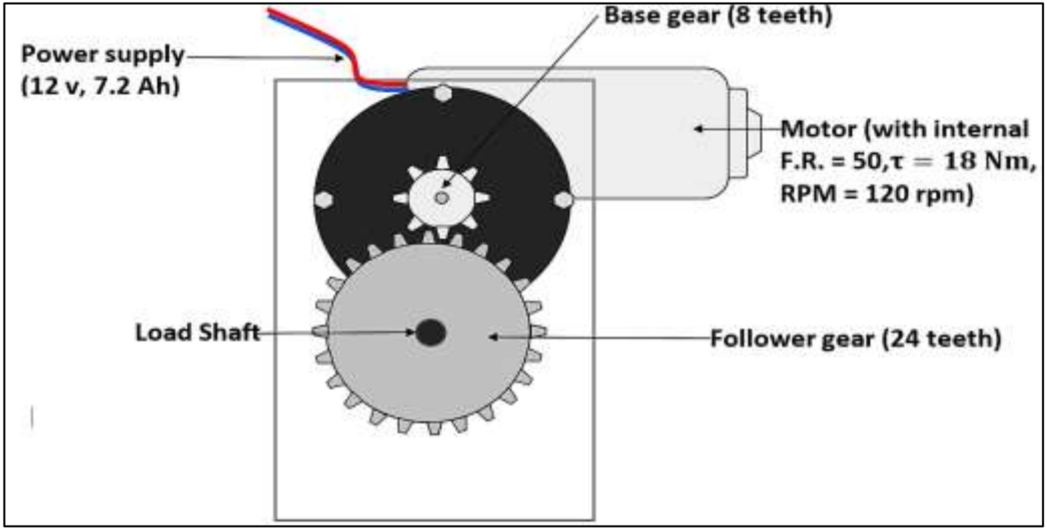
The variable, n_F denotes the number of teeth in the follower gear and n_B is the number of teeth of the base gear. The torque equation is expressed according to the following equation.

$$G.R(\tau_B) + \tau_F - \tau_{Losss} = 0 \qquad (2)$$

Where τ_B is the torque input from the motor, which has an average torque of 18.0 Nm; τ_F is the torque output of the gear-train, while τ_{Losss} is the torque loss due to friction. By ignoring torque losses (i.e. τ_{Losss} = 0); the Mechanical Advantage (M.A.) equals G.R. so that the ideal value of τ_F maybe evaluated as per in eq.3.

$$\tau_F = 3 \cdot \tau_B = 3(18.0) = 54 \qquad (3)$$

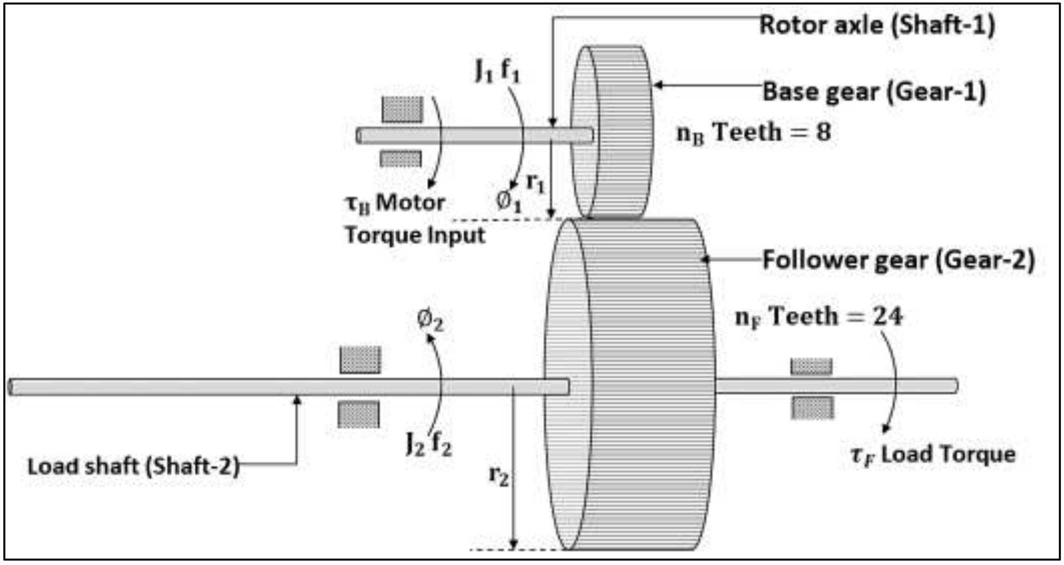
Fig-2. Power-train of the driving system



The τ_F is applied equally to the two rear wheels. On rough grounds, traction is generated, due to the inherent friction between the wheels and the ground, to propel the robot.

The 3D free-body diagram of the driving system's gear train is shown in Fig.3; this shows the motor driving the robot's wheel (i.e. the load) through the gear train. The angular displacement of shaft-1 and -2 are denoted by $\emptyset_1$ and $\emptyset_2$ respectively. The moment of inertial and viscous friction of the motor and gear-1 are denoted by J_1 and f_1 and those of gear-2 and the load denoted by J_2 and f_2. While r_1 and r_2 are the radii of gear-1 and -2 respectively.

Fig-3. Three-D drawing of the gear-train system



Based on the concept of Nagrath and Gopal [5], the describing equation for this gear-train is given as follows:

For shaft-1, the dynamic equation is:

$$J_1\frac{d^2\emptyset_1}{dt^2} + f_1\frac{d\emptyset_1}{dt} + \tau_1 = \tau_B \qquad (4)$$

Where: τ_1 → is the load torque on gear 1.

For shaft-2, the dynamic equation is:

$$J_2\frac{d^2\emptyset_2}{dt^2} + f_1\frac{d\emptyset_2}{dt} + \tau_F = \tau_2 \qquad (5)$$

Where: τ_2 → torque transferred to gear 2 .

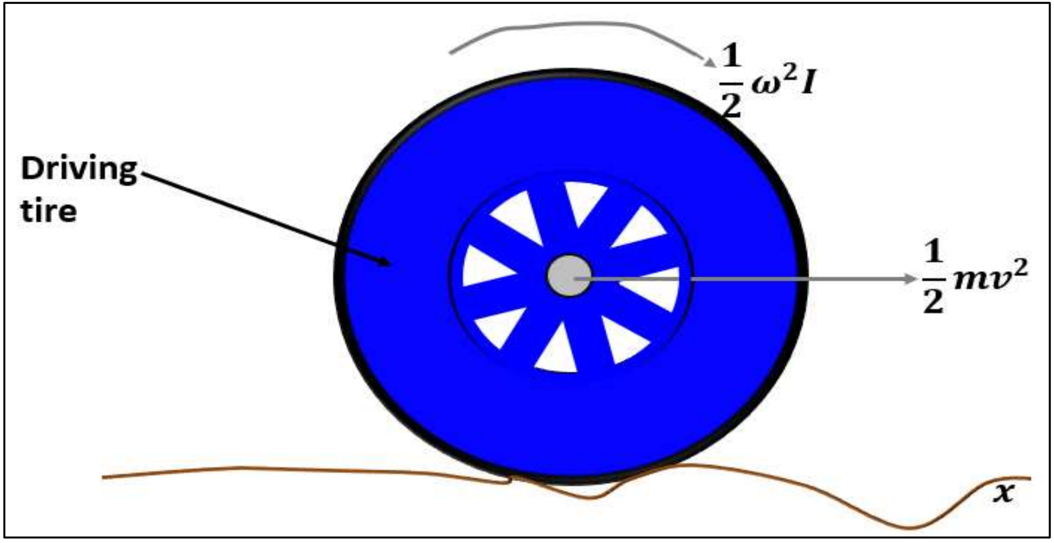
Given that; $\frac{\tau_1}{\tau_2} = \frac{\emptyset_2}{\emptyset_1} = \frac{n_B}{n_F} = \frac{1}{G.R}$, we have:

$$J_{2eq}\frac{d^2\emptyset_2}{dt^2} + f_{2eq}\frac{d^2\emptyset_2}{dt^2} + \tau_F = (G.R)\tau_B \qquad (6)$$

Where: $J_{2eq} = J_2 + (G.R)^2 J_1$ and $f_{2eq} = f_2 + (G.R)^2 f_1$.

The driving rear wheels are articulated to the propelling load-shaft to convert its rotary motion into the translational motion of the robot. Coupled with the cushioning rubber used in lining the wheels and its large area in contact with the ground, enough traction is generated to give about 80% skid-free driving motion. The masses of the wheels are concentrated at the rims as shown in Fig. 4; so that at constant friction and approximately uniform motion, a moment of inertia could build-up at each wheel. This enhances the optimization of the wheels' rotational energy as expressed in eq. 7.

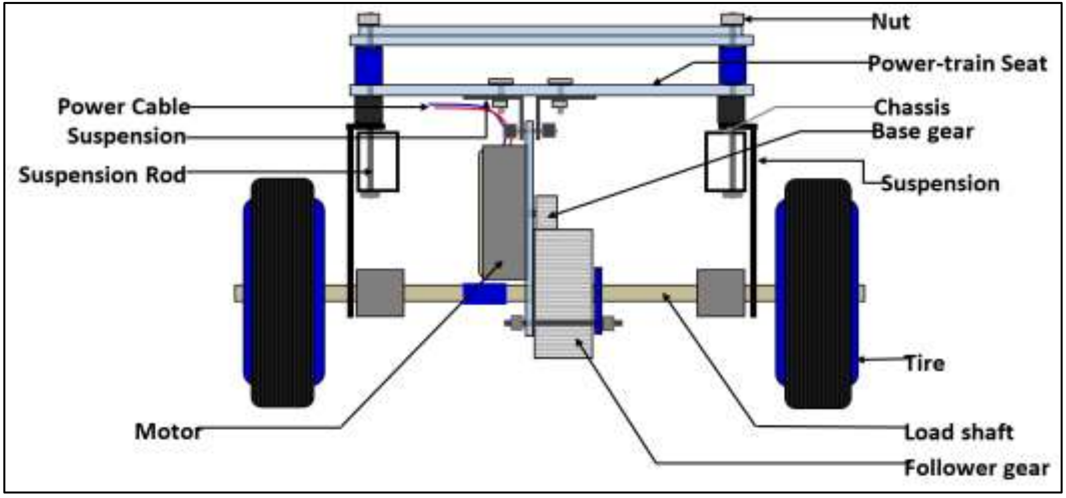
Fig-4. Schematic showing the effective rotational energy of each driving wheels



$$E = \frac{1}{2}mv^2 + \frac{1}{2}\omega^2 I = \frac{1}{2}mv^2 + \frac{1}{2}\omega^2 mr^2 = \frac{1}{2}v^2\left[m + \frac{I}{r^2}\right] \quad (7)$$

The Eq.7 represents the effective energy of each driving wheel. The physical-layout of this driving system is shown in Fig. 5. This depicts the actual design of the drivetrain.

Fig-5. Design of the robotic drivetrain



2.3. Steering Mechanism and Control

For the steering mechanism, a system similar in design to that of Olakanmi and Benyeogor [6], is developed. This uses a servo to steer the robot while it is translating as shown in Fig.6. To control this servo, the controller is instructed to generate Pulsed Width Modulated (PWM) signals, which are serially wired to the servo's in-built control circuit to specify the required steering angles. The timing diagram for the steering PWM is shown in Fig. 7.

Fig-6. Design of the robotic servo-steering system

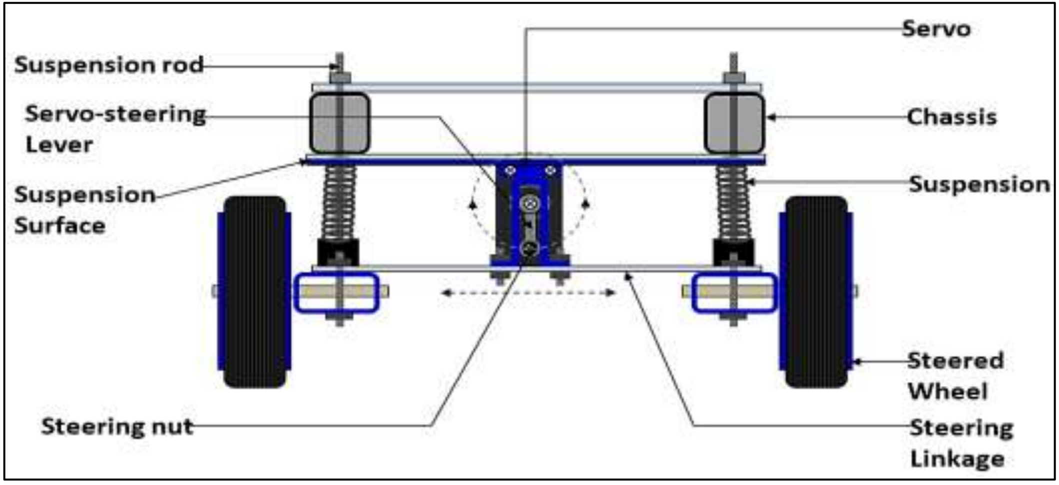
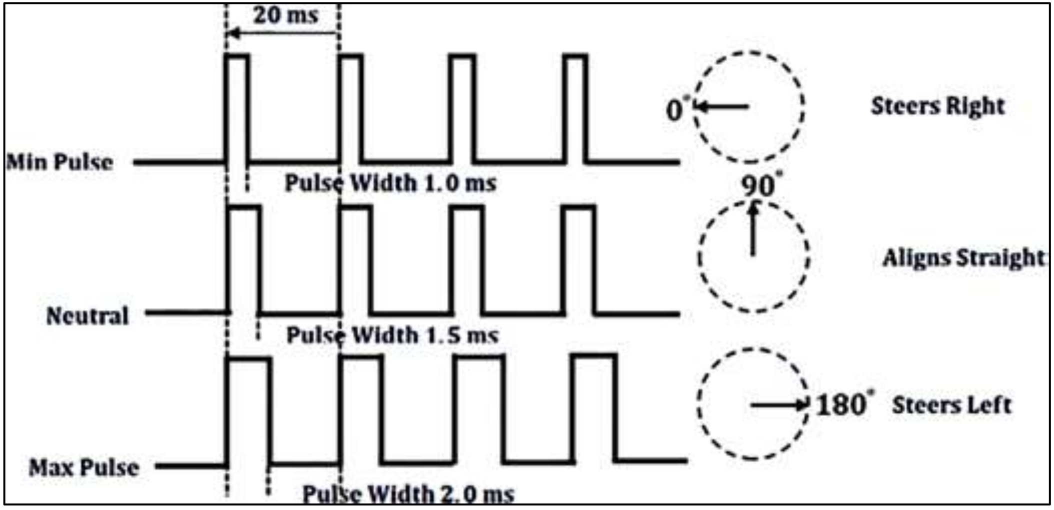


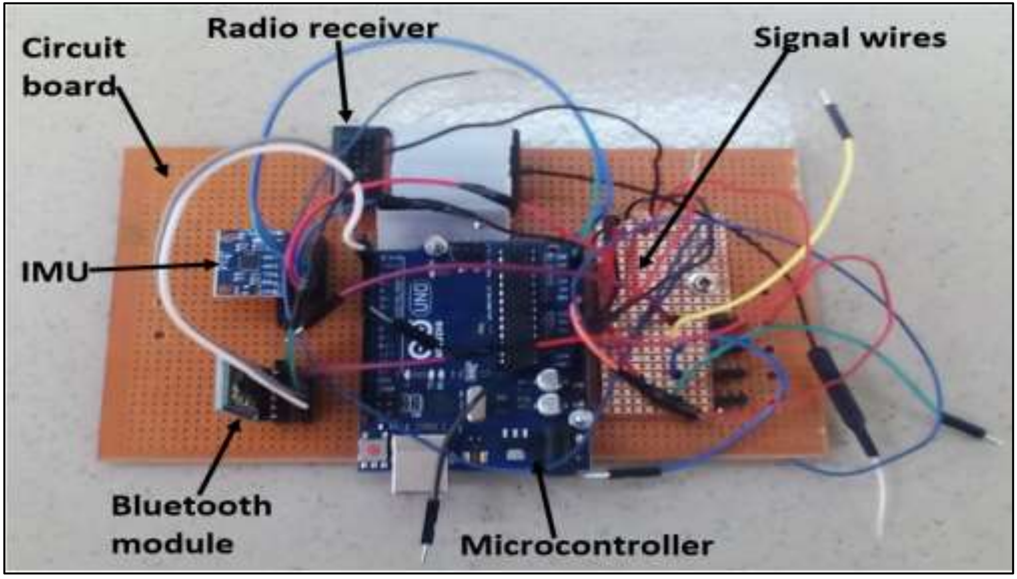
Fig-7. Pulse widths and corresponding steering angle



2.4. Control, Signaling and Software System

For control, we used the Personal Computer (PC). This PC is wirelessly interfaced with the robot's controller, which is developed with the Arduino microcontroller, in a Master-Slave architecture, using the Bluetooth communications protocol. The robot's controller also contains a radio receiver (for long-distance teleoperation) and an MPU6050 Inertial Measurement Unit (IMU), as shown in Fig. 8. The PC primarily acts as a gate-way that enables the human operator to serially transmit selected ASCII characters to the controller, at a baud-rate of 9600 bits-per-second, through a Teraterm communication interface. Each of these characters is mapped to the various robot control functions that switch actuations like drive, reverse, steering towards right or left and stop. The mapping of these functions is encrypted in the robot's control software. The major aspects of this embedded software is outlined in Listing 1. Also included in this software are program lines for reading attitude measurements from the IMU.

Fig-8. The constructed robot controller as a system of inter-connected microchips



Listing 1 Main robot control codes – mapping the various actuations to selected ASCII characters

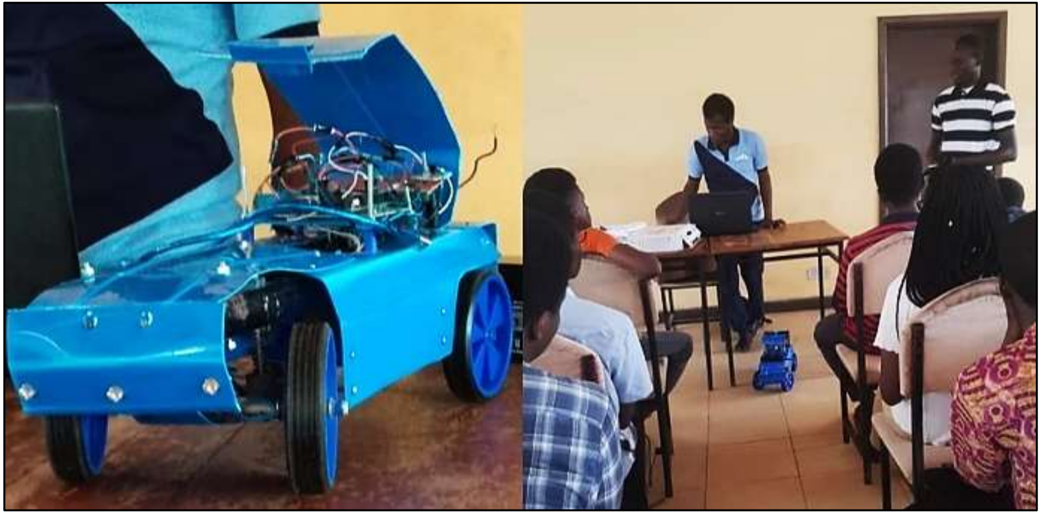
```
//START POINT
#include <Servo.h> //PREPROCESSOR DIRECTIVES AND LIBRARY FUNCTIONS
#include<Wire.h>
const int MPU6050_addr=0x68;
int16_t AccX, AccY, AccZ, Temp, GyroX, GyroY, GyroZ;
int servo_steering = 3; //ASSIGN PIN TO STEERING SERVO
Servo Servo1; // CREATE SERVO OBJECT
int buzzer = 4; //ASSIGN PIN TO BUZZER
int drive = 5; //ASSIGN PIN TO FORWARD-DRIVE RELAY-PIN
int reverse =6; //ASSIGN PIN TO REVERSE-DRIVE RELAY-PIN
int LED = 7; //ASSIGN PIN TO LED
int controlSignal = 0; //ASSIGN PIN FOR PC SERIAL CONTROL

void setup () // FINITE SETUP_SEQUENCE
{
 Wire.begin();
 Wire.beginTransmission(MPU6050_addr);
 Wire.write(0x6B);
 Wire.write(0);
 Wire.endTransmission(true);
 Servo1.attach(servo_steering);
 pinMode (buzzer, OUTPUT);
 pinMode (drive, OUTPUT);
 pinMode (reverse, OUTPUT);
 pinMode (LED, OUTPUT);
 digitalWrite (buzzer, HIGH);
 delay (200);
 digitalWrite (buzzer, LOW);
 digitalWrite (LED, HIGH);
 delay (200);
 digitalWrite (LED, LOW);
 Serial.begin(9600);
 }

void loop () //INFINITE CONTROL-LOOP
{
  //IDLE MODE
  Servo1.write(90); //Returns steering to straight alignment
  Wire.beginTransmission(MPU6050_addr); //IMU measurements
  Wire.write(0x3B);
  Wire.endTransmission(false);
  Wire.requestFrom(MPU6050_addr,14, true);
  AccX=Wire.read() << 8|Wire.read();
  AccY=Wire.read() << 8|Wire.read();
  AccZ=Wire.read() << 8|Wire.read();
  Temp=Wire.read() << 8|Wire.read();
  GyroX=Wire.read() << 8|Wire.read();
  GyroY=Wire.read() << 8|Wire.read();
  GyroZ=Wire.read() << 8|Wire.read();
  Serial.print("AccX = "); Serial.print(AccX);
  Serial.print(" || AccY = "); Serial.print(AccY);
  Serial.print(" || AccZ = "); Serial.print(AccZ);
  Serial.print(" || Temp = "); Serial.print(Temp/340.00+36.53);
  Serial.print(" || GyroX = "); Serial.print(GyroX);
  Serial.print(" || GyroY = "); Serial.print(GyroY);
  Serial.print(" || GyroZ = "); Serial.println(GyroZ);
  delay (100);

if (Serial.available() > 0)
     {
controlSignal = Serial.read();
     }

switch(controlSignal) //RADIO-CONTROL MODE
        {
   case 'q':
   Servo1.write(180);
   delay (500);
   Serial.println("Steering leftward");
   digitalWrite (LED, HIGH);
   delay (500);
   digitalWrite (LED, LOW);
   controlSignal='*';
   break;

   case 'g':
   Servo1.write(90);
   delay (500);
   Serial.println("Steering straight");
   digitalWrite (LED, HIGH);
   delay (500);
   digitalWrite (LED, LOW);
   controlSignal='*';
   break;

   case 'p':
   Servo1.write(0);
   delay (500);
   Serial.println("Steering rightward");
   digitalWrite (LED, HIGH);
   delay (500);
   digitalWrite (LED, LOW);
   controlSignal='*';
   break;

   case 'd':
   digitalWrite (drive, HIGH);
   delay (2000);
   Serial.println("Drives forward");
   Serial.println("press key 's' for braking");
   digitalWrite (drive, LOW);
   controlSignal='*';
   break;

   case 'b':
   digitalWrite (reverse, HIGH);
   delay (2000);
   Serial.println("reverses Backward");
   digitalWrite (reverse, LOW);
   controlSignal='*';
   break;

   case 's':
   digitalWrite (drive, LOW);
   controlSignal='*';
   Serial.println("Robot halts!");
   break;

   case 'l':
   digitalWrite (LED, HIGH);
   Serial.println("LED switches on");
   controlSignal='*';
   break;

   case 'o':
   digitalWrite (LED, LOW);
   Serial.println("LED switches off");
   controlSignal='*';
   break;
    }
} //END POINT
```

3. Testing Phase and Results

3.1. The Complete System

Our work culminates in a robot in the form of a small-scale vehicular system as shown in Fig.9. This has proven to be effective for exploring different topographical locations. A demonstration of this system can be viewed in [7]. Similar tests were conducted at different topographical locations to ascertain the dynamical behaviors and performances of the developed robot.

Fig-9. The complete robot at the testing phase



3.2. Test Results

Using an Inertial Measurement Unit (IMU), we measured the dynamics (i.e. the variation of the pitch- and roll-angle) of the robot while it's in motion, as a function of time, in different topographies of constant distance. 3D models of these dynamics were visualized on MATLAB. These gave analytical insights into how effectively our robot could traverse terrains of different topographies. The results are presented in Fig.-10, -11, and -12.

Fig-10. Plot of robot dynamics in 3D scattering – describing moderate-speed and wobbling motion on rough topography

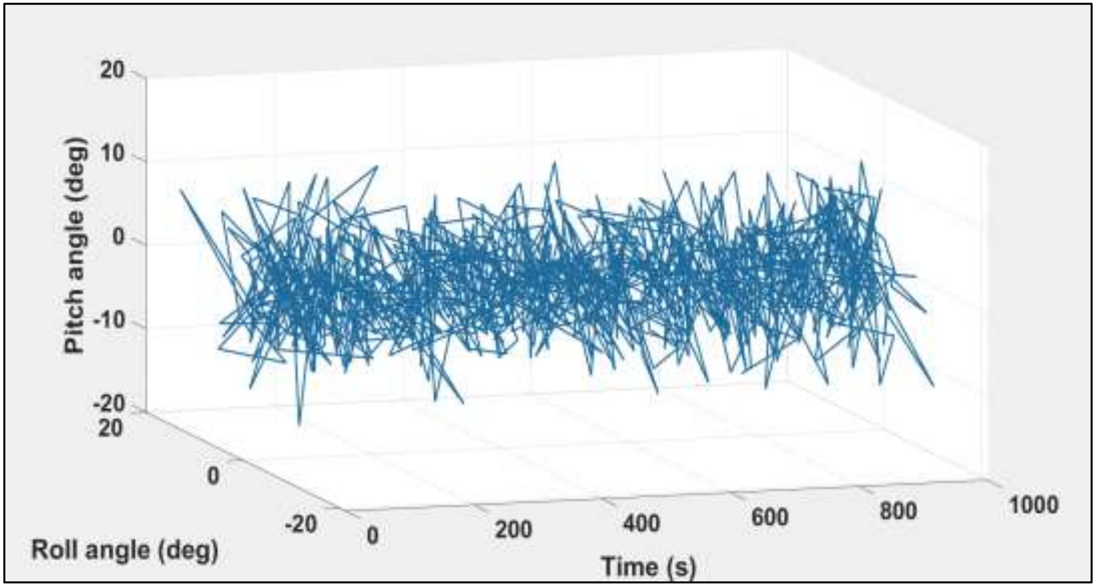


Fig-11. Plot of robot dynamics in 3D scattering – describing fast and quasi-stable motion on a paved pathway

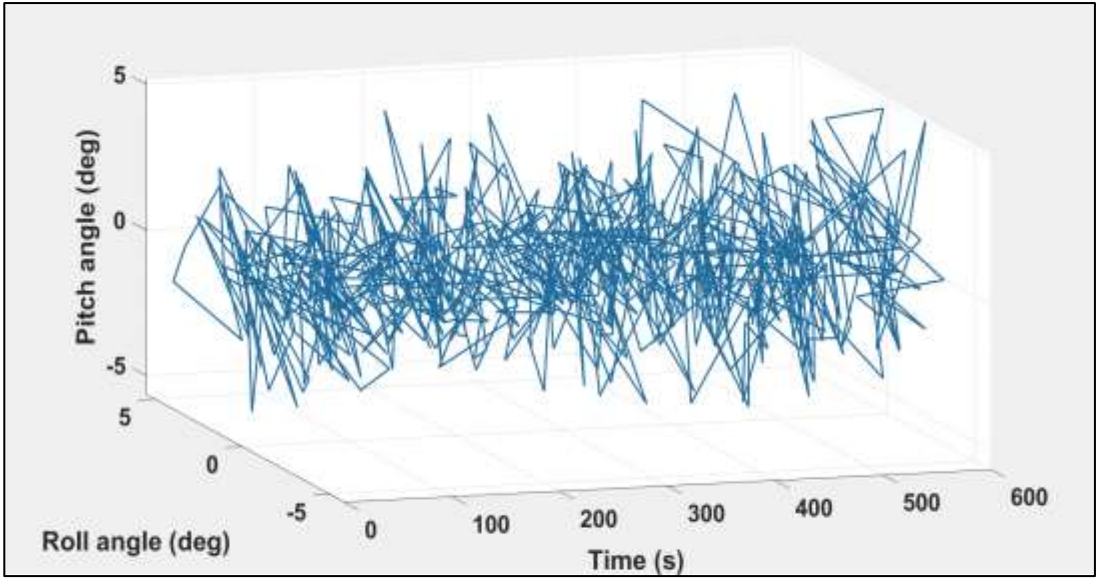
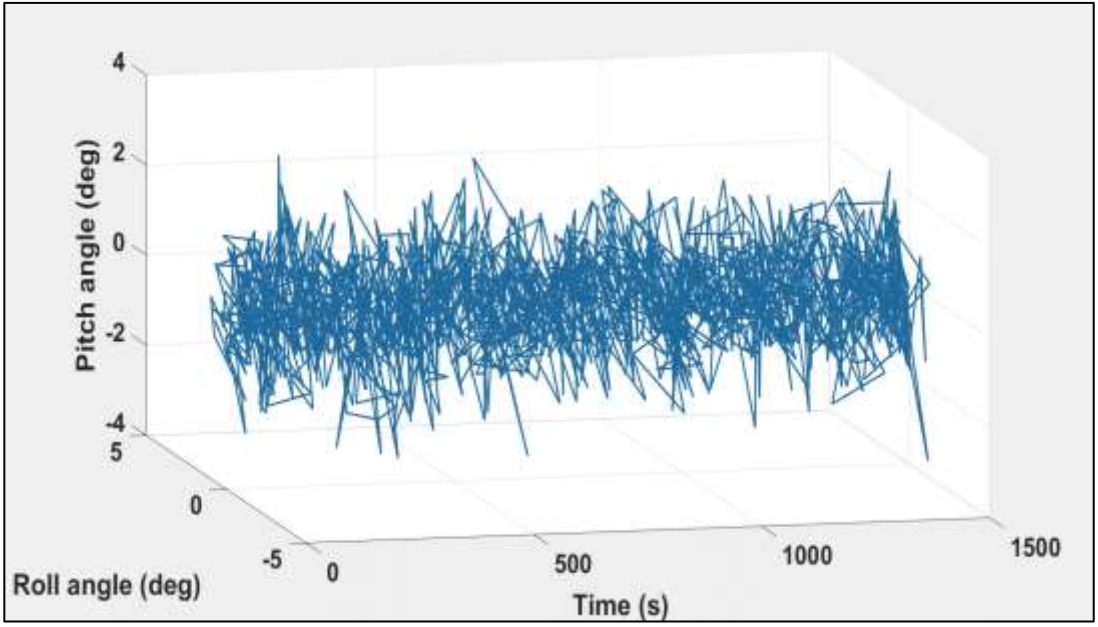


Fig-12. Plot of robot dynamics in 3D scattering – describing very slow and stable motion on a smooth floor



The analysis shows that on rough ground, the robot wobbles vigorously but traverse a given distance very quickly. On the contrary, the robot skids too often, while running on smooth surfaces due to poor traction, in spite of high dynamic stability. Consequently, the robot takes a long time to cover the same distance as when it moved on a smooth surface.

4. Conclusion

Based on the results, it is evident that our robot could be deployed to unmotorable terrains for various applications. This has also given insights on how to measure and visualize the dynamics of a robot moving on rough grounds. In light of this, it is recommended that future works on the similar system may focus on the development of active-suspension and usage of tires with chunky threading, to enhance high traction navigation on both rough and smooth topographies.

Acknowledgment

Our sincere gratitude goes to Office of Curriculum Development (OCD), Good-news Group of Schools, Ibadan for supporting our project with grants and equipment. We also appreciate the management of the International College of Aeronautics (ICA), Lagos for providing us with a platform for testing and exhibition.

References

[1] Iagnemma, K., Burner, R., Wilhelm, E., and Dubowsky, S., 1999. "Experimental validation of physics-based planning and control algorithm for planetary robotic rover." In *Proceeding of the Sixth International Symposium on Experimental Robotics, ISER'99.*

[2] Rajat, K., Ashok, S., Ishan, G., Vipin, K., Anuj, K., Tanrun, W., and Monika, C., 2007. *Robotics: Appin knowledge solution*. Hingham: Infinity Science Press LLC. p. 14.

[3] Siegwart, R. and Nourbakhsh, R. I., 2004. *Introduction to autonomous mobile robot*. Massachusetts: The MIT Press. p. 59.

[4] Benyeogor and Danladi, B., 2013. *Principle, design, and construction of prototype locomotive, department of physics*. Aliero: Unpublished: Kebbi State University of Science and Technology.

[5] Nagrath, I. J. and Gopal, M., 2007. *Mathematical models of physical systems, in control systems engineering.* 5th ed. New Delhi: New Age International Publishers. pp. 32-34.

[6] Olakanmi, O. O. and Benyeogor, M. S., 2019. "Internet-based tele-autonomous vehicle system with beyond-line-of-sight capability for remote sensing and monitoring." *Internet of Things,* vol. 5, pp. 97–11.

[7] Benyeogor, 2018. *Building model robotic vehicle*. Aeroclub: Unibadan.