Signal and Image Processing Letters
Vol. 3., No. 3, 2021, pp. 20-25
ISSN 2714-6677
https://simple.ascee.org/

Control of Electric Car Wheel Rotation Speed Using an Accelerometer Sensor (MPU-6050)

Fadjar Nur Faalah [a,1], Ahmad Zyusrotul Hanna [a,2], Alfian Ma'arif [a,3,*]

[a] Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] fadjar1800022067@webmail.uad.ac.id; [2] ahmad1900022065@webmai.uad.ac.id; [3] alfian.maarif@te.uad.ac.id
* Corresponding author

ARTICLE INFO

Keywords
MPU6050
Controller
Accelerometer

ABSTRACT

In the modern era, the implementation of electrification technology is one of the efforts that needs to be applied to create fuel-efficient and environmentally friendly vehicles. This technology encompasses various types, ranging from hybrid to full electric vehicles. There are four types of electric vehicle drivetrains and power sources used, namely: Battery Electric Vehicle (BEV), Hybrid Electric Vehicle (HEV), Plug-in Hybrid Electric Vehicle (PHEV), and Fuel Cell Electric Vehicle (FCEV). The prototype system of an electric car aims to create an Electrical Control Unit (ECU) with a feature that enhances acceleration in a multifunctional electric car, which can be used in both conventional and unconventional vehicles, such as Formula-E racing cars. The participation in creating an electric car prototype is used for testing the speed control of electric cars using an accelerometer sensor (MPU-6050) placed on the steering wheel, similar to electrification technology. The drivetrain technology aims to improve acceleration (the change in speed within a specific time unit) in electric cars. The components used in creating the electric car prototype include the main components such as the Arduino Uno R3 microcontroller, MPU-6050 sensor, and potentiometer.

1. Introduction

In the modern era, the implementation of electrification technology is one of the efforts that needs to be applied to create fuel-efficient and environmentally friendly vehicles. This technology encompasses various types, ranging from hybrid to full electric vehicles [1]. There are four types of electric vehicle drivetrains and power sources used, namely: Battery Electric Vehicle (BEV), Hybrid Electric Vehicle (HEV), Plug-in Hybrid Electric Vehicle (PHEV), and Fuel Cell Electric Vehicle (FCEV) [2].

The participation in creating an electric car prototype is used for testing the speed control of electric cars using an accelerometer sensor (MPU-6050) placed on the steering wheel, similar to electrification technology [3]. The drivetrain technology aims to improve acceleration (the change in speed within a specific time unit) in electric cars [4]. The components used in creating the electric car prototype include the main components such as the Arduino Uno R3 microcontroller, MPU-6050 sensor, and potentiometer [5]. Additionally, a DC motor equipped with an encoder sensor will be used as an actuator and for collecting RPM (Rotations Per Minute) data [6]. This data will be displayed through a 16x2 I2C LCD [7]. The L298N motor driver is used as a medium to transmit PWM (Pulse Width Modulation) input values from the microcontroller to the DC motor, enabling it to operate at a voltage of 5V-12V in the form of a DAC (Digital to Analog Converter) signal [8].

The prototype system of an electric car aims to create an Electrical Control Unit (ECU) with a feature that enhances acceleration in a multifunctional electric car, which can be used in both conventional and unconventional vehicles, such as Formula-E racing cars [9]-[12].

2. Method

2.1. Software Design

The software design process involves the creation of program code, which begins with the initialization of ports on the components and the use of program libraries for proper functioning [10]. The program code creation process is done using the Arduino IDE software and will be compiled into the Arduino Uno R3 microcontroller [13]-[15], which serves as the system for reading the designed sensor, as shown in Fig. 1.

2.2. System and Hardware Design

From the flowchart in Fig. 1, the system flow can be interpreted as follows: the input consists of the MPU6050 sensor and the potentiometer [16]-[17]. These inputs are processed by the microcontroller to read the input values. If the outputValue aY > 0, then both the right and left motors will rotate clockwise. If the value of aY > 0.01, then the speed of the right motor will decrease according to the output value. Similarly, if the value of aY < 0.01, the speed of the left motor will decrease. If none of these conditions are met, the process will loop back for further processing repeatedly.

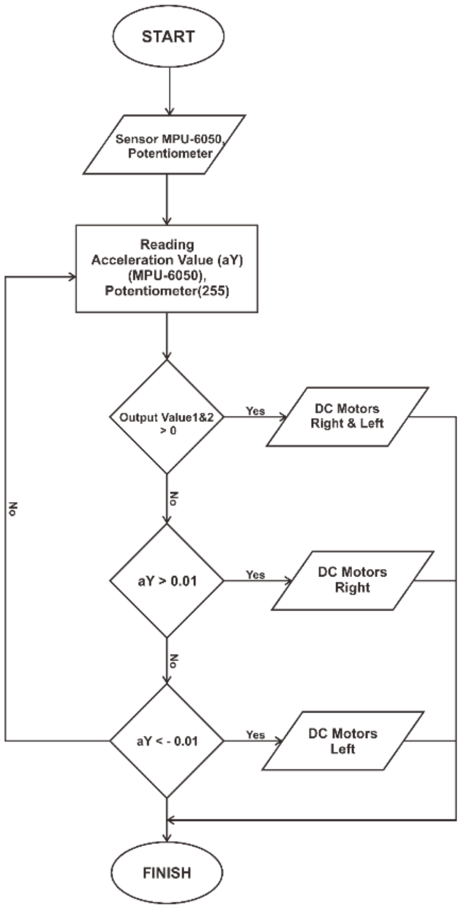


Fig. 1. Flowchart

The block diagram in Fig. 2 illustrates the design of a water flow monitoring system, which utilizes the NodeMCU ESP8266 microcontroller as the system controller [18]-[20]. The wiring diagram in Fig. 3 shows the design of the connection paths for each device. The NodeMCU is used as the microcontroller in this setup.

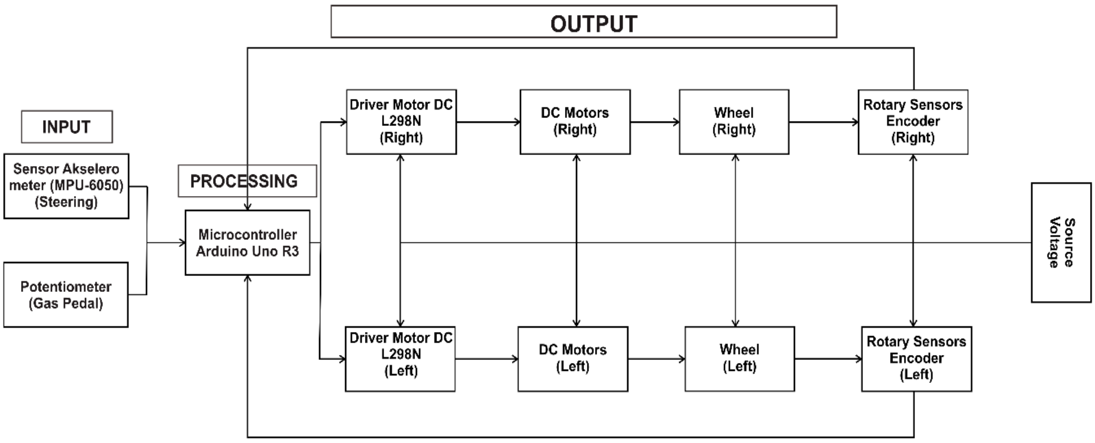
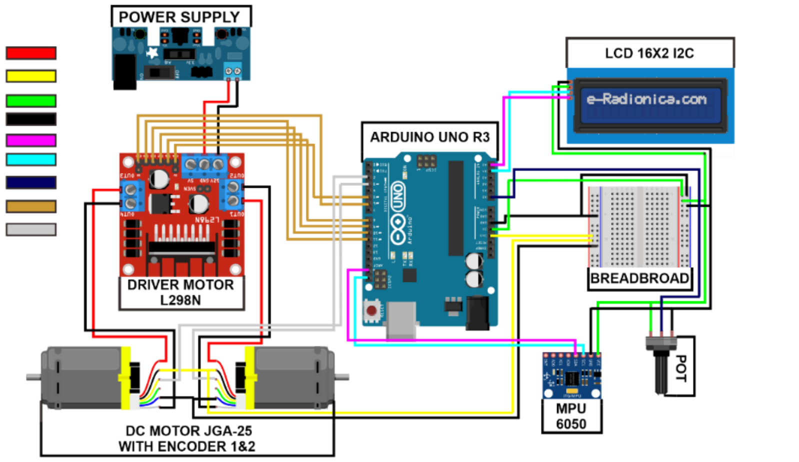


Fig. 2. Block Diagram



No	Description of cable color	Explanation
1	POWER SUPPLY VOLTAGE 12V(+)	Motor Pole JGA-25, Drive Motor L298N .
2	ARDUINO VOLTAGE 3.3V(+)	Encoder Pole Motor JGA-25 1&2.
3	ARDUINO VOLTAGE 5V(+)	LCD Pole 16x2 I2C, POT, MPU-6050.
4	GROUND(-)	LCD Pole 16x2 I2C, MPU-6050, POT, Encoder Motor, JGA-25 1&2, Power Supply, Arduino, Drive Motor L298N.
5	SDA	LCD 16x2 I2C, MPU-6050(A4).
6	SCL	LCD 16x2 I2C, MPU-6051(A5).
7		POT(Potensio Meter, A0).
8	PIN DRIVER MOTOR L298N	ENA(D5), ENB(D6), IN1(D8), IN2(D9), IN3(D10), IN4(D11).
9		Pin Interupt Encoder Motor JGA-25 1(D2)&2(D3).

Fig. 3. Wiring Diagram

3. Results and Discussion

Fig. 4 illustrates the hardware design result used for the final testing of the built system. It shows the detailed circuitry of the controller, which consists of Arduino Uno R3, Motor Driver L298N, DC Motor, Potentiometer, MPU-6050 sensor, and LCD module, along with the ports that connect the controller to the DC motor and MPU-6050 sensor.

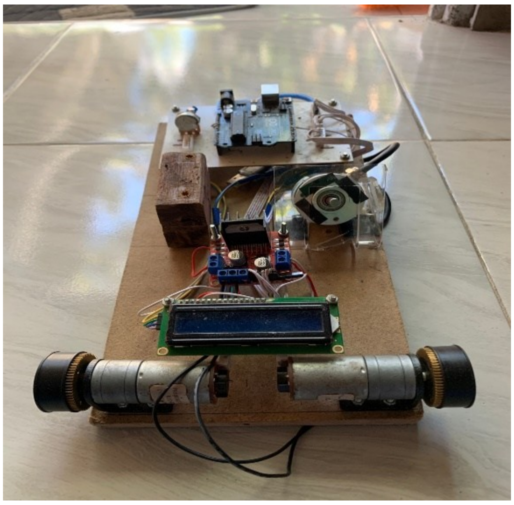


Fig. 4. Hardware Design: (Controller and Sensor Circuitry)

3.1. Testing of MPU-6050 Sensor

The testing of the MPU6050 sensor aims to determine whether the sensor can accurately read the angle of inclination. The test is performed three hundred times. After conducting the tests, a comparison value between the MPU6050 sensor and a degree measuring instrument (protractor) is obtained. The values from the MPU6050 sensor testing can be seen in Table 1.

In Table 1, the test results of the MPU6050 sensor are shown. A standard measuring instrument used for comparison is a 180-degree protractor, which serves as a reference for sensor calibration and comparison. It can be concluded that there are differences in degree values due to uneven placement of the instrument, hence the need for calibration based on the reference value. The graph illustrating the comparison of MPU6050 sensor values is shown in Fig. 1.

Here is the conversion of raw data values outputted by the sensor to degrees in equation (1).

$$\mathrm{aY} = (\mathrm{aY}/16384 * 100)/2 * 2 \quad (1)$$

Explanation:

aY = Raw data value read by the sensor.

16384 = Conversion factor from g-force to m/s2 (9.8 m/s2 is equivalent to 1 g).

3.2. Testing of DC Motor Encoder

The testing of the DC motor encoder aims to determine the RPM (Rotations Per Minute) speed produced by the DC motor. The results can be seen in Table 1. From Table 2, it can be concluded that there is a difference in RPM readings between the encoder and the tachometer. This significant difference is due to the variation in the interval value used in the RPM conversion formula between the system and the tachometer. The RPM conversion formula for the system uses an interval of 0.01/ms, while the tachometer uses 0.08. This indicates the need for calibration based on the reference value. Table 3 shows the results of the DC motor encoder sensor testing. A graphical image of a DC motor encoder is shown in Fig. 5.

The conversion equation for converting encoder readings to RPM can be defined in equation (2) and equation (3).

$$Rps = (Pulse\ Count\ /\ 360.00/0.01) \quad (2)$$

$$Rpm = Rps \times 60 \quad (3)$$

Explanation:

Rps = Motor revolutions per second /ms.

360 = Unit for a full revolution.

Rpm = Motor speed per minute/m.

Table 1. Testing of MPU-6050 Sensor

No	Reading Raw Data from MPU-6050 Sensor			
	Turn Left		Turn Right	
	Curve	Sensors	Curve	Sensors
1	-30	-491.42	30	491.42
2	-60	-983.040	60	983.040
3	-90	-1.474.5	90	1.474.5

Table 2. Testing of MPU6050 sensor data conversion formulas

No	Reading of MPU-6050 Sensor data					
	Turn Left			Turn Right		
	Curve	Sensors	Curve	Sensors	Curve	Error Value
1	-30	-31.32	-1.32	30	31.13	1.13
2	-60	-63.09	-3.09	60	61.38	1.38
3	-90	-92.07	-2.07	90	87.77	2.33

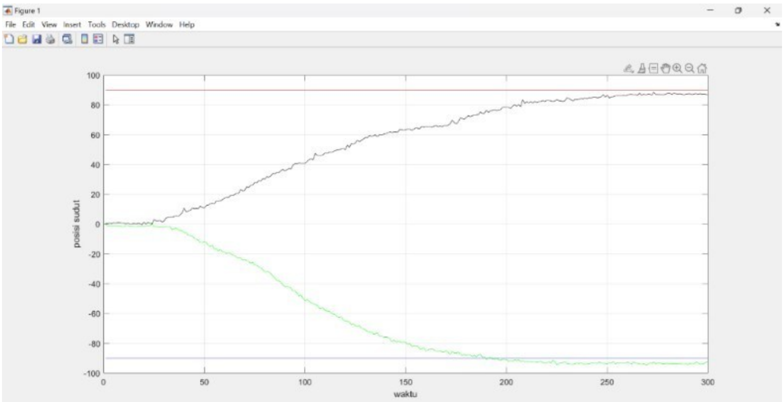


Fig. 5. Graph of readings from MPU-6050 sensor on the Y-axis

Table 3. Testing of DC motor encoder

	Left DC Motor		
No	**ADC Value**	**RPM Value**	**Tachometer RPM Value**
1	50	360	152.4
2	100	720	307.4
3	150	1020	408.9
4	200	1200	477.7
5	250	1320	548.1
	Right DC Motor		
No	**ADC Value**	**RPM Value**	**Tachometer RPM Value**
1	50	360	155.2
2	100	780	319.1
3	150	960	409.9
4	200	1200	490.1
5	250	1380	571.2

4. Conclusion

Based on the results of system design and testing by conducting system and sensor experiments, several conclusions are obtained. The error values of the accelerometer sensor readings compared to the calibrated reference tool in the form of a degree arc resulted in the following data for left turns: error values of (-1.32. -3.09, -2.07) at 30, 60, and 90 degrees, respectively. For right turns, the error values were (1.13, 1.38, 2.33) at 30, 60, and 90 degrees, respectively. PWM values ranging from 50 to 250 resulted in the following motor speeds for the left and right DC motors: RPM values ranging from 360 to 1320 (left DC motor) and 360 to 1380 (right DC motor). The tachometer RPM values were recorded as 152.4 to 548.1 (left DC motor) and 155.2 to 571.2 (right DC motor). From these results, it can be observed that the motor response is not stable, indicating instability in the system.

References

[1] R. Rocha Ribeiro, E. Bauer, and R. Lameiras, "HIGROTERM: An Open-Source and Low-Cost Temperature and Humidity Monitoring System for Laboratory Applications," *Inventions*, vol. 6, no. 4, p. 84, 2021.

[2] A. Latif, P. Megantoro, "The Prototype of Automatic Water Sprinkle with Soil Moisture Sensor Based on ATMega 8535," *Journal of Physics: Conference Series*, vol. 1464, no. 1, p. 012035, 2020.

[3] S. Zhang, M. Yu, Y. Lu, and Q. Fan, "Research on control strategy of handling stability for formula SAE (FSAE) pure electric racing car," *Aust. J. Mech. Eng.*, vol. 16, no. sup1, pp. 61–67, 2018.

[4] H. E. Næss and A. Tjønndal, "Innovation, Sustainability and Management in Motorsports," *Innov. Sustain. Manag. Mot.*, pp. 1–16, 2021.

[5] S. Fatimah Anggraini, A. Ma, and R. Dwi Puriyanto, "PID Controller on DC Motors and Tuning Using the Differential Evolution Method PID Controllers on DC Motors and Tuning Using the Differential Evolution Method," *Telka*, vol. 6, no. 2, pp. 147–159, 2020.

[6] R. I. Alfian, A. Ma'arif, S. Sunardi, "Noise reduction in the accelerometer and gyroscope sensor with the Kalman filter algorithm," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, pp. 180-189, 2021.

[7] P. G. Anselma, "Optimal adaptive race strategy for a Formula-E car," *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.*, vol. 236, no. 9, pp. 2185–2199, 2022.

[8] I. Hudati, A. P. Aji, and S. Nurrahma, "Control of DC Motor Position Using PID Control," *J. List. Instrumentation and Electrons. Terap.*, vol. 2, no. 2, pp. 1–6, 2021.

[9] M. A. Basthomi *et al*., "Walking Balance Control for Humanoid Soccer Robot on Synthetic Grass," *2020 International Electronics Symposium (IES)*, pp. 213-218, 2020.

[10] S. Ur Rehman, S. A. Khan, A. Arif and U. S. Khan, "IoT-based Accident Detection and Emergency Alert System for Motorbikes," *2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, pp. 1-5, 2021.

[11] I. Asy Syamsbeta and I. Asy Syamsbeta, "IoT-Based Hydroponic Treatment Automation System for Kale Plants," *J. Electron. and Automation Ind.*, vol. 8, no. 3, p. 279, 2021.

[12] R. A. Amrullah, H. Herwandi, and A. Pracoyo, "Design and Manufacture of a 150WATT Pure Sine Wave Inverter With 220/50Hz AC Feedback Based on Arduino Microcontroller," *J. Electron. and Automation Ind.*, vol. 8, no. 2, p. 96, 2021.

[13] F. Loucif, S. Kechida, A. Sebbagh, "Whale optimizer algorithm to tune PID controller for the trajectory tracking control of robot manipulator," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 42, no. 1, p. 1, 2020.

[14] J. Sardi and A. Basrah Pulungan, "Bioelectrical Impedance As Control Commands for Speed Control for Wheel Chairs Using the PID Controller Method," *J. Nas. Tech. Electro*, vol. 3, no. 2, p. 125, 2014.

[15] S. Chaouch *et al*., "DC-Motor Control Using Arduino-Uno Board for Wire-Feed System," *2018 International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*, pp. 1-6, 2018.

[16] J. Jumiyatun, "Dc Motor Speed Control Using Encoder Sensors With Pi Control," J. ECOTYPE, vol. 4, no. 1, pp. 23–27, 2017.

[17] X. Zhang, "Design and implementation of fuzzy PID DC motor control system based on STM32," *2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT)*, pp. 1129-1131, 2023.

[18] A. Latif, A. Z. Arfianto, H. A. Widodo, R. Rahim, E. T. Helmy, "Motor DC PID system regulator for mini conveyor drive based-on MATLAB," *Journal of Robotics and Control (JRC)*, vol. 1, no. 6, pp. 185-190, 2020.

[19] M. Alessandrini, G. Biagetti, P. Crippa, L. Falaschetti, and C. Turchetti, "Recurrent Neural Network for Human Activity Recognition in Embedded Systems Using PPG and Accelerometer Data," *Electronics*, vol. 10, no. 14, p. 1715, 2021.

[20] A. Yudhana *et al.*, "Flex sensors and MPU6050 sensors responses on smart glove for sign language translation," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 1, no. 1, pp. 71–77, 2019.