Design and Development of an IoT-Based Drive-By-Wire System for an Electric All-Terrain Vehicle

Yogesh Katare[1], Ravi Gujrati[2], Imran Khan[3], Prof.Hemant Marmat[4]

[1]*M. tech student, Department of Internet of Things, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India*
[2]*M. tech student, Department of Internet of Things, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India*
[3]*Project In-charge, Department of Mechanical Engineering, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India*
[4]*Associate Professor, Department of Mechanical Engineering, Acropolis Institute of Technology and Research, Indore, Madhya Pradesh, India*

ABSTRACT

This research paper outlines the design, development, and testing of an IoT-based Drive-By-Wire (DBW) system for an electric All-Terrain Vehicle (ATV). The system replaces traditional mechanical throttle, braking, and steering linkages with electronic controls, enhancing control precision, adaptability, and safety in dynamic off-road conditions. A Manual/Automatic switching feature allows seamless transition between manual and electronic control modes. The DBW system employs Jetson Nano as the central processing unit and integrates Socket.IO for real-time communication. Four Arduino microcontrollers are utilized: one for throttle control, one for brake control, one for steering control, and a dedicated Mega Arduino for data acquisition and feedback from various sensors. Comprehensive validation tests demonstrate the system's functionality, reliability, and significant potential for improving off-road electric vehicle performance.

Keywords: Drive-by-wire, Throttle-by-wire, Brake-by-wire, Steer-by-wire, Electric vehicle, IoT-based system, Electric ATV, Vehicle automation, electronic control systems.

I. INTRODUCTION

The increasing demand for off-road mobility, coupled with advancements in automation and electrification, has driven significant progress in automotive technology. One of the most exciting developments is the transition towards Drive-By-Wire (DBW) systems, particularly for off-road All-Terrain Vehicles (ATVs). Traditional mechanical linkages for steering, throttle, and braking can introduce lag and imprecision in control, making it challenging to achieve optimal performance and safety in dynamic off-road environments. DBW systems address these limitations by replacing mechanical linkages with electronic controls, offering significant improvements in precision, control, and adaptability.

This research focuses on the design, implementation, and testing of a comprehensive DBW system for an electric ATV, incorporating Throttle-By-Wire, Brake-By-Wire, and Steer-By-Wire mechanisms. The system utilizes Jetson Nano as the central processing unit and integrates Socket.IO for real-time communication. Four Arduino microcontrollers are employed: one for each of the throttle, brake, and steering subsystems, and a dedicated Mega Arduino for data acquisition and feedback from various sensors. This paper details the system architecture, hardware design, communication protocols, and validation test results, along with a discussion of the system's potential impact, including improved safety through automation features and enhanced driver experience.

This paper presents the design and development of a robust IoT-based Drive-By-Wire (DBW) system for an electric All-Terrain Vehicle (ATV). This project aimed to design and implement Throttle-By-Wire, Brake-By-Wire and Steer-By-Wire mechanisms systems, replacing traditional mechanical systems with electronic controls to enhance vehicle response, precision, and adaptability in dynamic terrain conditions. The system enables precise vehicle speed and braking control via electronic commands transmitted through a web interface hosted on a Jetson Nano server, with real-time feedback for adjustments.

A manual/automatic switching system has been incorporated, allowing the driver to switch seamlessly between manual and electronic control modes. The hypothesis is that integrating these systems will significantly improve the vehicle's overall performance, stability, and control, making it better suited for off-road navigation. The system's integration with other vehicle control modules, including object detection and terrain analysis, is discussed to provide context for its role in the overall vehicle architecture.

Comprehensive validation tests were performed to assess the functionality and reliability of the Drive-By-Wire system. The results demonstrate that the system meets the requirements for automating the control of an electric ATV, with high accuracy and reliability. After extensive testing, the system proved to be a robust and reliable solution for the control of an electric ATV. This work lays the foundation for future advancements in IoT-based vehicle control technologies.

II. Related Works

The integration of electronics into vehicles began in the mid-20th century, marking a pivotal shift in the automotive industry. One of the earliest milestones was the introduction of the first transistorized ignition system in vehicles by Lucas Industries in 1955. This advancement paved the way for a series of innovations that would later lead to the widespread adoption of electronics in automotive systems. Shortly thereafter, Advanced Driver Assistance Systems (ADAS) emerged, incorporating a variety of electronic devices aimed at enhancing driver and passenger safety and comfort [4].

By the 1960s, significant advancements in automotive electronics began to surface. Ford’s Mercury Park Lane model, introduced in 1965, was the first to feature Electro-Hydraulic Powered Steering (EHPS). Just a few years later, Volkswagen launched the first vehicle equipped with electronic fuel injection in 1968, further demonstrating the shift toward electronic control. The 1970s saw the arrival of safety features like the Antilock Braking System (ABS) by Mercedes-Benz, and by 1990, Honda’s NSX became the first vehicle to feature Electric Powered Steering (EPS) [5]. Initially, such technologies were mostly confined to high-end vehicles due to their cost, but as research in automotive electronics progressed, these systems began to trickle down to more mainstream vehicles.

As the automotive industry continued to evolve, the drive-by-wire concept emerged, representing a significant departure from traditional mechanical linkages in favor of electronic control systems. This concept aims to replace mechanical control mechanisms with electronic interfaces, allowing for more precise and flexible vehicle control. Within the broader Drive-By-Wire framework, there are three primary subsystems: throttle-by-wire, Brake-By-Wire, and Steer-By-Wire.

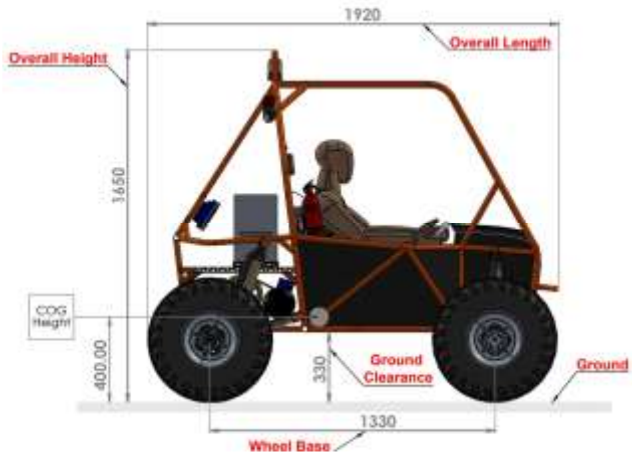
Several studies have explored the individual components of Drive-By-Wire (DBW) systems, including throttle-by-wire for precise engine control (J.L. Anderson, 2019), brake-by-wire for enhanced safety and faster response times in autonomous vehicles (M.B. Lee, 2021), and steer-by-wire for improved maneuverability (referenced earlier). Additionally, research has addressed the importance of real-time control and feedback systems using platforms like Arduino and Jetson Nano for efficient sensor data processing and actuator control (A.N. Singh, 2020). Furthermore, studies have explored the use of communication protocols like Socket.IO to facilitate reliable data exchange between control units and microcontrollers in DBW systems (P. Kumar, 2023). These findings highlight the growing body of research on DBW technology, focusing on both individual components and system-level integration for improved performance and safety in autonomous and electric vehicles.

II. System Architecture

The electric ATV used in this project features a [motor type, e.g., brushless DC] motor powered by a [battery voltage] V Lithium-ion battery pack. The chassis is constructed from [chassis material, e.g., high-strength steel tubing] to ensure durability in off-road conditions.

Table4.1. Technical specifications of the ATV vehicle.

Parameter	**Value**	**Parameter**	**Value**
Track Width	*1230mm*	Motor type	PMSM
Wheelbase	1330mm	Rated Power	5kw
Ground Clearance	330mm	Maximum Speed	4500 RPM
Overall, Height	1650 mm	Rated Torque	39 nm
Weight (with driver)	240 kg	Voltage controller	48 - 72v
Weight Distribution(R)	64 %	Battery voltage	58.8
Weight Distribution(F)	36%	Top Speed	55 Km/h
		Max torque	59 nm



(a)Side View
Figure 1. ATV Vehicle Specification

The core architecture of our Drive-By-Wire system, illustrated in Figure2, is designed to electronically control the vehicle's throttle, brake, and steering systems, eliminating the need for traditional mechanical linkages. The system operates through a series of interconnected modules that communicate via a local network. This network facilitates real-time data exchange between the throttle control system, brake control system, steering control system, feedback sensors, and the Jetson Nano for centralized control.

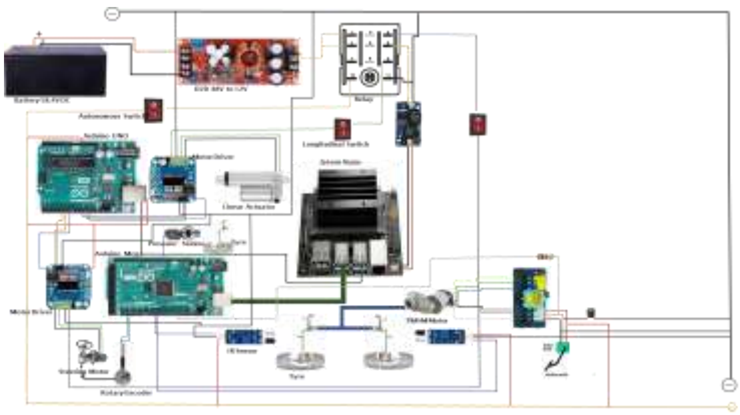


Figure 2: Block diagram of the DBW system architecture

Central Processing Unit

Jetson Nano: This serves as the brain of the system, a powerful central processing unit (CPU) responsible for several critical tasks. It receives commands from the web interface, processes them, and generates control signals for the throttle, brake, and steering systems. Additionally, the Jetson Nano manages communication with all other components within the DBW system.

Throttle Control System

Arduino Uno (Throttle): This microcontroller acts as the dedicated control unit for the throttle system. It receives control signals transmitted by the Jetson Nano via USB communication. Based on these signals, the Arduino Uno generates Pulse Width Modulation (PWM) signals to regulate the throttle motor, precisely controlling the vehicle's speed.

Throttle Motor**:** This electric motor directly controls the throttle position within the ATV's drivetrain. By adjusting the speed and rotation of the throttle motor based on PWM signals, the system can achieve precise control over vehicle acceleration.

RC Circuit**:** This electronic circuit plays a vital role in ensuring smooth and accurate throttle control. It filters out any unwanted high-frequency components present in the PWM signals from the Arduino Uno, resulting in a steadier and more precise actuation of the throttle motor.

IR Sensor**:** This sensor continuously measures the vehicle's speed. This data is then fed back to the Arduino Mega for monitoring purposes and potential use in control algorithms.

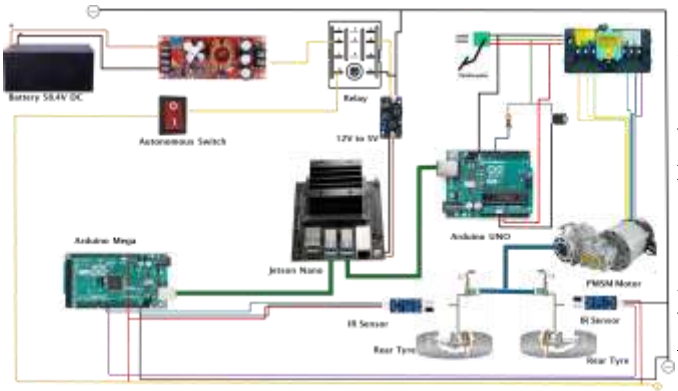


Fig 3: Hardware Architecture of Throttle-By-Wire

Brake Control System

Arduino Uno (Brake): Similar to the throttle control system, a dedicated Arduino Uno (Brake) is responsible for managing the braking operation. It receives control signals from the Jetson Nano and translates them into PWM signals for the brake actuator.

Linear Actuator: This electromechanical device directly controls the braking force applied to the wheels. By varying the position of the actuator based on PWM signals, the system can regulate the braking force and achieve controlled deceleration of the ATV.

Pressure Sensor: Mounted within the brake line, this sensor continuously monitors the brake line pressure. This data is transmitted to the Arduino Mega, providing valuable feedback for monitoring and potential use in control algorithms.

Motor Driver: This electronic circuit serves as the intermediary between the Arduino Uno (Brake) and the linear actuator. It receives PWM control signals from the Arduino and translates them into the appropriate voltage and current levels required to drive the linear actuator, ensuring effective brake control.

Fig 4: Real time position of our brake actuator and pressure sensor.

Steering Control System

Arduino Uno (Steering): Dedicated to steering control, this Arduino Uno receives control signals from the Jetson Nano and generates PWM signals to regulate the steering motor.

Steering Motor: This electric motor is responsible for rotating the steering column, ultimately controlling the direction of the ATV's wheels. By varying the speed and direction of the motor's rotation based on PWM signals, the system can achieve precise steering control.

Rotary Encoder: This sensor plays a crucial role in the steering system by measuring the steering wheel's angular position. This feedback data is transmitted to the Arduino Mega, providing information about the steering angle for monitoring and potential use in control algorithms.

Communication System Wi-Fi Module: This module enables wireless cmmunication between the Jetson Nano and the web interface. Users can send control commands to the DBW system from a remote device connected to the web interface through a Wi-Fi network.

Socket.IO: This real-time communication library facilitates bidirectional communication between the web interface and the Jetson Nano. It allows for the seamless exchange of control commands and sensor data between these two components in real-time.

USB Communication: Each Arduino Uno (Throttle, Brake, Steering) is connected to the Jetson Nano via a USB cable. This enables serial communication, allowing Jetson Nano to transmit control signals and receive sensor data from the individual Arduino microcontrollers.

1) Driving Functions: The Drive-By-Wire (DBW) system in the ATV replaces traditional mechanical linkages with electronic controls for throttle, brake, and steering functions.

Throttle-by-Wire: The system uses an Arduino Uno to generate a Pulse Width Modulation (PWM) signal, passed through a low-pass filter to produce a smooth voltage range of 0-5V. This voltage range controls the throttle motor, regulating speed from 0 to 30 km/h. Speed control is proportional to the voltage, with 0V corresponding to 0 km/h and 5V corresponding to 30 km/h. The formula for voltage-speed mapping is:

Voltage=Speed (km/h)30×5 (V)\text{Voltage} = \frac{\text{Speed} \, (km/h)}{30} \times 5 \, (V)Voltage=30Speed(km/h)×5(V)

An IR sensor, mounted near the rear axle, measures the rotational speed of the shaft and provides real-time feedback for closed-loop throttle control.

Brake-by-Wire: A linear actuator rated at 24V with a speed of 120 mm/sec is connected to the brake pedal. The actuation force is calculated using the formula:

$F = m \cdot a$

Where:

Braking force
Vehicle mass (e.g., 300 kg)
Deceleration (e.g., -5 m/s² for emergency stops)

For a 300 kg vehicle:

$F = 300\ kg \times (-5\ m/s^2) = -1500\ N$

A pressure sensor monitors brake line pressure, providing real-time feedback to the Arduino Uno and Jetson Nano for precise braking control.

Steer-by-Wire: The steering system utilizes a Hyundai Electric Power Steering Motor and a rotary encoder for feedback. The encoder measures the steering angle, which is adjusted based on input commands. The Jetson Nano receives steering commands and transmits them to an Arduino Uno, which generates the PWM signal to drive the motor. The motor rotates the steering column, achieving a road wheel angle of ±30° for left and right turns.

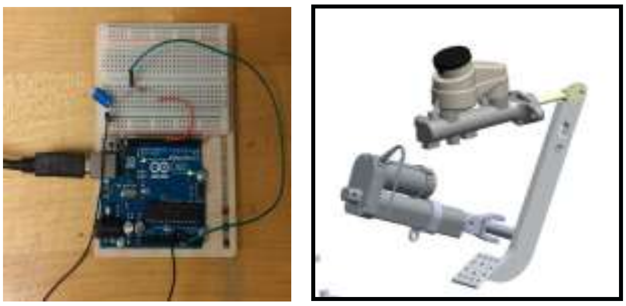


Fig 5: Brake actuator setup with RC circuit Arduino connection

2) Control Functions: The control of the DBW system is achieved through a combination of high-level processing and low-level actuation. The Jetson Nano serves as the core processing unit, interfacing with a user interface to receive throttle and brake commands. These commands are transmitted to dedicated Arduino Unos for throttle and brake control. The Arduino Uno (Throttle) processes the speed command and generates the required PWM signal for the throttle motor, while the Arduino Uno (Brake) generates control signals for the motor driver of the linear actuator. Real-time feedback is incorporated into the control loop: the IR sensor measures the vehicle's speed and sends data to the Jetson Nano for throttle adjustments, while the pressure sensor continuously monitors brake line pressure to ensure accurate braking force. This closed-loop control ensures consistent performance and enhanced safety.

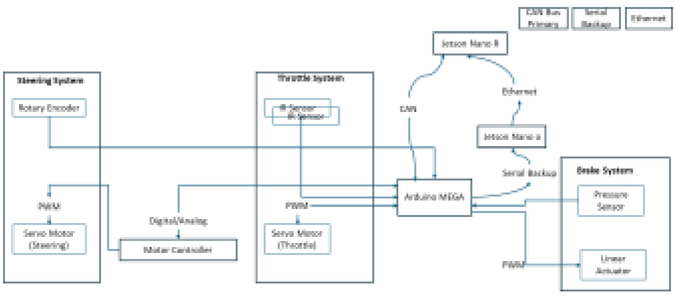


Fig 6 : Ccommunication Protocol of Drive-By-Wire System

3) Sensor Connection: The DBW system integrates sensors to provide real-time feedback for precise control. An IR sensor, positioned near the rear axle, monitors the rotational speed of the shaft. The sensor data is processed by the Arduino Uno and sent to the Jetson Nano to calculate vehicle speed. This feedback loop enables the system to dynamically adjust throttle inputs for maintaining desired speed. For the braking system, a pressure sensor is integrated into the hydraulic brake line to monitor real-time brake pressure. The sensor transmits this data to the Arduino Uno, which relays

it to the Jetson Nano for closed-loop control adjustments. Both sensors ensure smooth operation and efficient performance of the ATV in diverse operating conditions.

The system integrates sensors for real-time feedback:

IR Sensor: Positioned near the rear axle, it monitors shaft rotations to calculate speed. For a shaft with 60 teeth, speed is calculated as:

Speed (km/h)=Rotational Speed (RPM)×Tire Circumference (m)60 ×1000

Example: For 600 RPM and a tire circumference of 2 m:

Speed=600×260×1000=20 km/h

Pressure Sensor: Monitors hydraulic brake line pressure, with a selected range of 0-2500 psi based on a maximum system pressure of 2000 psi.

Rotary Encoder**:** Measures steering column angle in real time, providing precise feedback to the Jetson Nano for maintaining desired steering behavior.

III. SOFTWARE SYSTEM

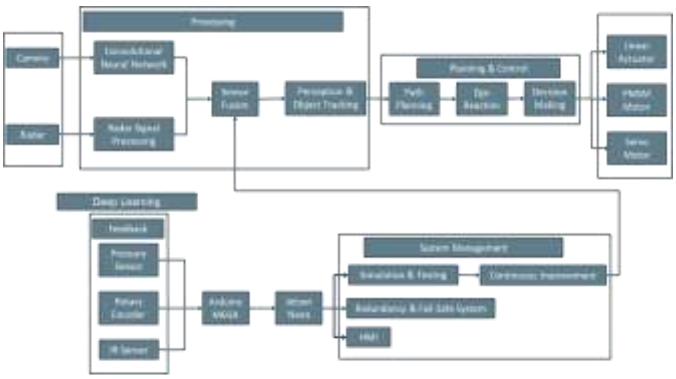


Fig 7 : Complete Vehicle Software Architecture

The software architecture for the Drive-By-Wire system integrates various modules to ensure precise control, efficient communication, and real-time feedback. The system is divided into three primary components: the Perception and Control System, Central Processing Unit, and User Interface.

Perception and Control System: This subsystem is implemented on Arduino microcontrollers, each dedicated to specific tasks such as throttle, brake, and steering control. These microcontrollers use PWM (Pulse Width Modulation) signals for actuator control and employ PID (Proportional-Integral-Derivative) algorithms for fine-tuned adjustments based on real-time sensor feedback. The Arduino for throttle control uses feedback from an IR sensor to maintain the desired speed, while the brake control Arduino uses pressure sensor data to modulate braking force dynamically. The steering subsystem, equipped with a rotary encoder, ensures precise angular adjustments based on user commands. Interrupt-based programming on the Arduino facilitates prompt responses to real-time changes, enhancing system performance.

Central Processing Unit**:** The Jetson Nano serves as the central hub for managing high-level commands, data aggregation, and communication with other components. It runs a Python Flask application, which interfaces with the website via Socket.IO, enabling seamless communication between the user and the vehicle systems. The Jetson Nano processes inputs from the website and transmits control commands to the Arduinos for throttle, brake, and steering adjustments. Simultaneously, it collects real-time feedback from the microcontrollers, including speed, throttle position, brake pressure, and steering angle, ensuring continuous monitoring and optimization.

User Interface: The web-based interface provides an intuitive platform for users to interact with the system. It allows remote control of the vehicle's throttle, brake, and steering systems while displaying live feedback data. Features such as interactive controls and real-time data visualization enhance user engagement. Historical performance metrics, such as throttle response, braking force, and speed trends, are displayed dynamically through streaming charts, offering insights into system behavior.

This modular software design ensures a robust, responsive, and user-friendly Drive-By-Wire system that meets the demands of electric All-Terrain Vehicles in dynamic off-road environments.

IV. RESULTS AND DISCUSSION

The Drive-By-Wire (DBW) system in the ATV was rigorously tested and validated as part of the overall navigation architecture during its development. These tests were conducted in various controlled environments around the University Campus, focusing exclusively on the Throttle-By-Wire and Brake-By-Wire functionalities, without the integration of autonomous modules such as perception and decision-making. The validation process included the following key tests.

The Throttle-By-Wire system underwent a step response test to assess its accuracy in translating

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

throttle input voltage to the expected engine RPM under various operating conditions. The results, as detailed in Table 4 and the corresponding graphs, revealed that the system has fast response times, ranging from 0.3 to 0.6 seconds, depending on the input voltage. It also demonstrated a quick settling time of 0.6 to 1 second, indicating that the system stabilizes promptly without significant delays. The maximum error in RPM was minimal, approximately 2%, highlighting the system's high accuracy during both acceleration and deceleration. Throughout all tests, the system exhibited smooth acceleration with minimal fluctuations in RPM. Even during deceleration, when the throttle was released, the system responded immediately and returned to idle speed (0 RPM) without delay.

Table 4: Experimental Data for Throttle-By-Wire Step Response

Ex. No.	Throttle Input (V)	Expected RPM (Rev/min)	Actual RPM (Rev/min)	Response Time (s)	Settling Time (s)	Max Error (%)	Observations
1	0.5V	1000	980	0.3	0.7	2.0	Smooth response, minimal delay.
2	2.0V	2500	2450	0.4	0.8	2.0	Linear acceleration, quick stabilization.
3	3.5V	3200	3080	0.5	0.9	0.5	Response consistent, small fluctuation.
4	4.9V	4800	4300	0.6	1.0	1.5	Slight overshoot, but stable after settling.
5	0V (deceleration)	0	0	0.4	0.6	0.0	Immediate response to throttle release.

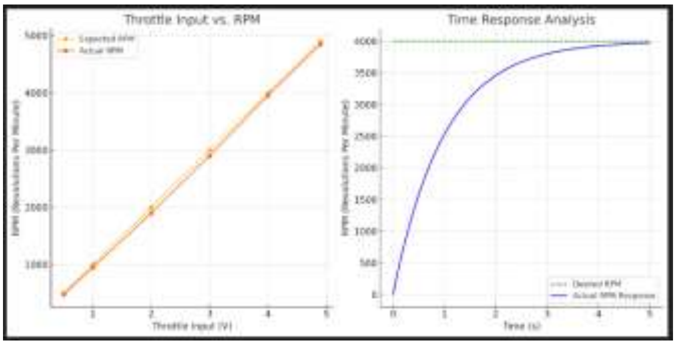


Fig 8. The graphs represent the performance analysis of the Throttle-By-Wire system during the step

The Brake-By-Wire system was subjected to a comprehensive series of tests, including a step response test, controlled deceleration, emergency braking, and dynamic performance assessments. These tests, as presented in Table 5 and accompanying graphs, showcase the system's ability to apply braking force effectively in response to inputs. The response time of the system was minimal, with a negligible delay between brake input and actuator response, ensuring rapid and reliable performance. The braking force applied was linear with respect to the brake input percentage, as seen in the test data. For instance, at a 0.60s input time, the braking force reached 5.45% while the input percentage was 6.06%. This linearity ensures that the system provides proportional braking, offering precise control over deceleration.

Table 5: Experimental results of the Brake-By-Wire system step response

Time (s)	Brake Input (%)	Braking Force (%)
0.00	0.00	0.00
0.10	1.01	0.91
0.20	2.02	1.82
0.30	3.03	2.73
0.40	4.04	3.64
0.50	5.05	4.55
0.60	6.06	5.45
0.70	7.07	6.36
0.80	8.08	7.27
9	4	+5

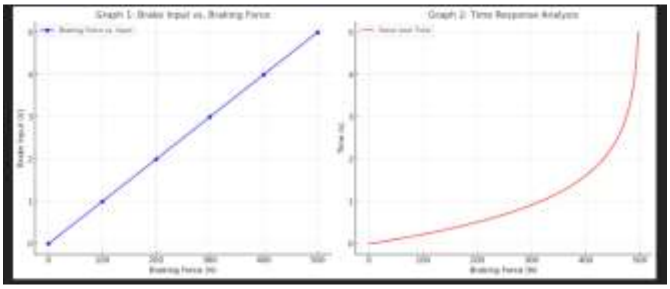


Fig 9. The graphs represent the performance analysis of the Break-By-Wire system during the step

In emergency braking tests, the system was able to handle maximum brake inputs effectively, bringing the vehicle to a complete stop in critical scenarios. This demonstrated the system's capability to respond to emergency situations with full stopping power, which is essential for safety. During the dynamic performance tests, the system maintained consistent braking performance, even under varying speeds and payloads, ensuring robustness in real-world conditions. The tests confirmed the system’s reliability, with stable braking force throughout, which was validated by the data showing that the braking force increased proportionally with brake input, from 0% to over 7%. This consistency and responsiveness make the Brake-By-Wire system an effective and reliable solution for safe vehicle operation.

The Steer-By-Wire system was rigorously tested to assess its accuracy in translating manual steering inputs to the corresponding validation angle of the wheel. The results, as seen in Table 6, show that the system performed exceptionally well, with minimal deviation between the manual input and the validation angle. The difference in angles ranged from 1 to 2 degrees, demonstrating the system’s high precision. For instance, when the manual angle was 10 degrees, the validation angle was 9 degrees, resulting in a difference of just 1 degree. Similarly, when the manual input was 30 degrees, the system achieved a validation angle of 28 degrees, with a 2-degree difference. These results indicate that the system maintains a consistent and reliable linear relationship between the manual steering input and the corresponding validation angle. The system demonstrated effective performance across a range of steering angles, proving its capability to provide accurate and smooth steering control, which is crucial for ensuring safe vehicle operation.

Table 6: Experimental results of Table Steering-by-wire Validation Angle

S.NO	Manual Angle	Validation Angle	Difference of Angle
1	10	9	1
2	20	18	2
3	30	28	2
4	35	33	2

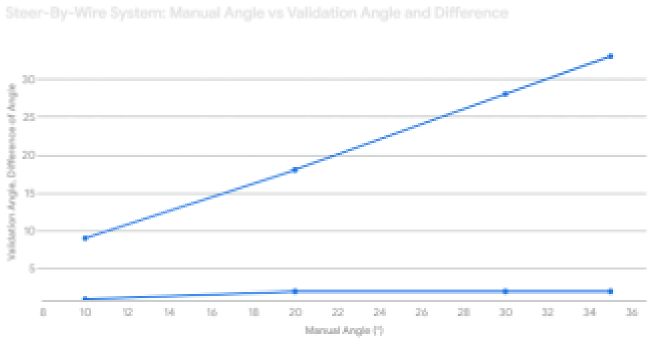


Fig 10. The graphs represent the performance analysis of the steer by wire system during the step

The Reliability and Failure Analysis was conducted by collecting data from a series of tests performed on the Throttle-By-Wire and Brake-By-Wire systems. Throughout the testing, no significant failures were observed, which indicates that both systems are highly reliable and robust. The analysis suggests that both the Throttle-By-Wire and Brake-By-Wire systems demonstrated consistent and dependable performance, even under varying operational conditions, and without the integration of higher-level autonomous modules such as perception and decision-making. This reinforces the systems' ability to operate effectively in real-world scenarios, ensuring stability and safety under diverse driving conditions. The absence of failure points further supports the systems' durability and readiness for future integrations with autonomous functionalities.

V. CONCLUSION

The Drive-By-Wire system for the ATV, which integrates the Throttle-By-Wire, Brake-By-Wire, and Steer-By-Wire systems, has been rigorously tested and validated to ensure its performance under controlled conditions. The Throttle-By-Wire system exhibited excellent responsiveness with fast response times ranging from 0.3 to 0.6 seconds, minimal errors (around 2%), and smooth stabilization during both acceleration and deceleration. The Brake-By-Wire system demonstrated high precision in braking control, with a linear relationship between brake input and applied force, quick response times, and reliable emergency braking performance. Similarly, the Steer-By-Wire system was validated with an accuracy of up to 2 degrees in angle response, ensuring smooth and controlled steering inputs. Although not fully detailed, the steer-by-wire system has shown promising results that align with the performance metrics observed in the throttle and brake systems. These findings confirm the reliability and accuracy of the Drive-By-Wire system, establishing a robust foundation for future integration with higher-level autonomous functions.

VI. REFERENCES

1] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," Advances in neural information processing systems, vol. 1, 1988.

[2] A. Broggi, Automatic vehicle guidance: the experience of the ARGO autonomous vehicle. World Scientific, 1999.

[3] M. Buehler, K. Iagnemma, and S. Singh, The 2005 DARPA grand challenge: the great robot race. Springer, 2007, vol. 36.

[4] M. Buehler, K. Iagnemma, and S. Singh, The DARPA urban challenge: autonomous vehicles in city traffic. springer, 2009, vol. 56.

[5] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann et al., "Stanley: The robot that won the darpa grand challenge," Journal of field Robotics, vol. 23, no. 9, pp. 661–692, 2006.

[6] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, "Junior: The stanford entry in the urban challenge," in Springer Tracts in Advanced Robotics, vol. 56, 2009, pp. 91–123.

[7] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer et al., "Autonomous driving in urban environments: Boss and the urban challenge," Journal of field Robotics, vol. 25, no. 8, pp. 425–466, 2008.

[8] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada, "An open approach to autonomous vehicles," IEEE Micro, vol. 35, no. 6, pp. 60–68, 2015.

[9] Baidu, "Apollo," https://github.com/ApolloAuto/apollo.

[10] E. Prassler, D. Schwammkrug, B. Rohrmoser, and G. Schmidl, "A robotic road sweeper," in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 3. IEEE, 2000, pp. 2364–2369.

[11] J. Jeon, B. Jung, J. C. Koo, H. R. Choi, H. Moon, A. Pintado, and P. Oh, "Autonomous robotic street sweeping: Initial attempt for curbside sweeping," in 2017 IEEE International Conference on Consumer Electronics (ICCE). IEEE, 2017, pp. 72–73.

[12] S. Zhiwei, H. Weiwei, W. Ning, W. Xiaojun, W. C. Y. Anthony, V. B. Saputra, B. C. H. Quan, C. J. Simon, Z. Qun, Y. Susu et al., "Map free lane following based on low-cost laser scanner for near future autonomous service vehicle," in 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015, pp. 706–711.

[13] J. Yu and Z. Yu, "Mono-vision based lateral localization system of low-cost autonomous vehicles using deep learning curb detection," in Actuators, vol. 10, no. 3. MDPI, 2021, p. 57.

[14] H. Andersen, Y. H. Eng, W. K. Leong, C. Zhang, H. X. Kong, S. Pendleton, M. H. Ang, and D. Rus, "Autonomous personal mobility scooter for multi-class mobility-on-demand service," in 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2016, pp. 1753–1760.

[15] S. Belakaria, M. Ammous, L. Smith, S. Sorour, and A. Abdel-Rahim, "Multi-class management with sub-class service for autonomous elec- tric mobility on-demand systems," IEEE Transactions on Vehicular Technology, vol. 68, no. 7, pp. 7155–7159, 2019.

[16] S. Pendleton, T. Uthaicharoenpong, Z. J. Chong, G. M. J. Fu, B. Qin, W. Liu, X. Shen, Z. Weng, C. Kamin, M. A. Ang et al., "Autonomous golf cars for public trial of mobility-on-demand service," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 1164–1171.

[17] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in ICRA Workshop on Open Source Software, 2009.

[18] S. Sun, "Lidar obstacle detector," 12 2021. [Online]. Available: https://github.com/SS47816/lidar obstacle detector

[19] I. Kurniawan, "Yolov4 tensorrt ros object detector," 8 2021. [Online]. Available: https://github.com/indra4837/yolov4 trt ros

[20] P. Biber, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in IEEE International Conference on Intelligent Robots and Systems, vol. 3, 2003, pp. 2743–2748.