Head-neck gesture classification for assistive wheelchair control

Student Details

Satyaprakash Rath
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
Telephone: +91 8260881544
E-mail: 122ei0621@nitrkl.ac.in

Student ID: 122EI0621

Supervisor Details

Dr.Alok Prakash
Assistant Professor
Department of Electronics and Communication Engineering
National Institute of Technology, Rourkela
E-mail:prakasha@nitrkl.ac.in

Supervisor Approval:

Signature: ____________________

Date: ____________

Abstract—**Quadriplegia causes complete or partial paralysis of all four limbs, severely restricting a person's mobility and independence. This work presents a head- and neck-gesture–controlled electric wheelchair that uses inertial motion sensing and wireless communication. The system incorporates an MPU6050 6-DOF IMU module combining a 3-axis accelerometer and 3-axis gyroscope to measure head orientation (pitch and roll). The IMU data are processed by an Arduino Nano (transmitter), where a Kalman filter performs sensor fusion to suppress drift and noise, resulting in stable orientation angles. Directional commands are generated when the user's head movement exceeds ±30° along the X-axis (roll) and Y-axis (pitch). These commands are transmitted via nRF24L01 2.4 GHz transceivers to an Arduino Uno (receiver), which actuates the wheelchair's DC motors and steering servos. The servo rotation is proportionally mapped to the user's tilt angle to ensure smooth and natural directional transitions. The wireless design enables low-latency, real-time response and improved user comfort. An ultrasonic sensor is integrated for obstacle detection; when an obstacle appears within a predefined range, the system automatically stops the wheelchair to ensure user safety. Experimental results show that the system provides stable, accurate, and continuous control suitable for quadriplegic users, while being significantly more cost-effective than EEG- or vision-based control approaches.**

Index Terms—**Head Gesture Control, Quadriplegia, Assistive Wheelchair, MPU6050, Arduino Nano, Arduino Uno, nRF24L01, Kalman Filter, Inertial Measurement Unit (IMU), Servo Motor Control, Wireless Communication, Threshold Detection,Ultrasonic Sensor, Embedded System, Human–Machine Interface (HMI).**

I. Introduction

Quadriplegic patients with cervical spinal cord injuries lose voluntary limb control and often depend on caregivers for movement. Traditional joystick-operated wheelchairs require upper-limb mobility and are therefore unsuitable. Alternatives such as voice, eye-tracking and EEG-based systems exist, but they are expensive, complex, need frequent calibration and are highly sensitive to lighting, noise and physiological variations.

This work presents a low-cost, head-gesture-controlled electric wheelchair using inertial motion sensing and wireless communication. An MPU6050 IMU measures head pitch and roll, while a Kalman filter on an Arduino Nano fuses accelerometer and gyroscope data to generate smooth, drift-free orientation estimates. Head tilts beyond ±30° are mapped to motion commands: forward pitch for forward movement, backward pitch for reverse and left/right roll for turning. Tilt magnitude is proportionally converted into PWM signals for smooth motor control.

Wireless communication is achieved using nRF24L01 2.4 GHz transceivers, providing low-latency, reliable control between the head-motion unit and the Arduino Uno-based wheelchair receiver. An ultrasonic sensor enhances safety by automatically stopping the wheelchair when an obstacle is detected.

Overall, the system is simpler, more robust and more affordable than voice-, eye- or EEG-based interfaces. With minimal calibration needs and stable Kalman-filtered IMU output, it offers a practical and reliable mobility solution for individuals with severe motor impairments.

II. Related Work

A variety of approaches have been developed for assistive wheelchairs to improve the mobility of people with severe motor impairments such as quadriplegia or tetraplegia. Conventional joystick-operated wheelchairs are unsuitable for users who cannot control their upper limbs, motivating research into head-, hand- and gesture-based control systems.

Notable developments include accelerometer-driven head motion control systems [2], which translate tilt angles into joystick-equivalent commands through embedded microcontrollers. These systems offer simplicity and low cost but depend on mechanical actuators that increase latency and reduce precision. Later advancements replaced mechanical actuation with direct electronic interfaces, employing MEMS-based gyroscopes and accelerometers to detect head gestures and convert them into motion commands [5].

Modern approaches further integrate IoT communication modules, wireless transceivers and health-monitoring sensors for safety and adaptability [4]. Additionally, vision-based systems using convolutional neural networks (CNNs) [6] enable contactless head pose estimation and multi-DOF orientation control. Although accurate, vision-based systems require substantial computational resources and controlled lighting.

Recent contributions extend into advanced computer vision pipelines combining Haar-like feature detectors, Kernelized Correlation Filters (KCF), and lightweight 2D-CNN gesture classifiers for real-time processing on embedded platforms [9]. These methods reduce dependence on specialized wearable sensors. Alternatively, near-field gesture sensors such as the MGC3130-based Skywriter module [10] offer precise 3D gesture detection without cameras but require the user's hand to be within a small sensing region.

Vision-based head gesture systems have also advanced significantly. Somawirata and Utaminingrum [8] used geometric relationships between facial landmarks (face and nose) to classify head movements, achieving reliable performance under controlled indoor conditions. Collectively, these works demonstrate a clear progression from mechanical interfaces to sensor-driven, computer-vision-based and hybrid smart wheelchair systems.

A. *Comparative Analysis of Techniques*

Different methodologies have been proposed to enable independent mobility for users with severe physical impairments. The comparative aspects primarily include sensor type, signal processing method, responsiveness, user adaptability and cost-effectiveness.

Accelerometer-based systems [2], [3] offer low cost and minimal power consumption. However, they require accurate calibration and suffer from drift or orientation dependence. Gyro–accelerometer combinations [5] improve gesture stability and reduce noise effects, enabling smoother navigation over uneven terrain.

IoT-integrated systems [4] add flexibility through remote monitoring and smartphone backup control but may suffer from network delays. Vision-based head pose estimation [6], or facial landmark–based gesture extraction [8], eliminates the need for physical sensors, providing intuitive and contactless control. However, these systems require high-quality cameras and robust processing hardware.

Hand-gesture-based wheelchairs appear in two major forms: (1) IMU-based wrist/finger gesture systems [7] and (2) vision- or sensor-based gesture recognition using CNNs, Haar cascades, and KCF trackers [9]. These systems are suitable for users with partial limb function. Near-field electrical sensing solutions [10] achieve accurate 3D gesture recognition without cameras but restrict motion to a small sensing region.

Major Findings from Previous Works:

1) Humanoid-Assisted Wheelchair Control: Work in [1] integrated humanoid robots with environment recognition and cooperative assistance, pioneering shared-autonomy mobility support.

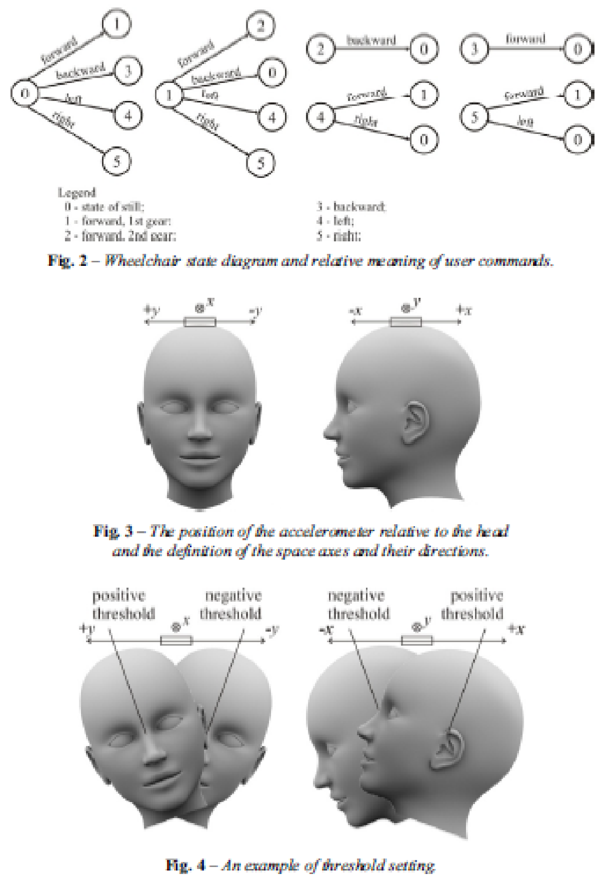

Fig. 1: Illustration of the thresholding concept used for head-gesture detection. Positive and negative angular thresholds are defined along the pitch and roll axes to distinguish intentional head movements from natural micro-motions. A head rotation exceeding these preset thresholds is interpreted as a valid control command, while smaller variations are rejected as noise.

2) Accelerometer-Based Head Motion Control: The study in [2] implemented a low-cost head tilt detection system as an alternative to joystick control, though sensitive to mechanical drift.

3) Multi-Motor Head Tilt Mechanism: The approach in [3] developed a high-torque multi-motor wheelchair using helmet-mounted sensors, improving outdoor stability on slopes.

4) IoT-Integrated Neck Movement Control: The work in [4] introduced Blynk-enabled neck motion detection with integrated health monitoring, suitable for indoor use but affected by network delays.

5) Gyro–Accelerometer Gesture Control: The method in [5] utilized IMU-based gesture detection to achieve reliable control on both smooth and rough terrains.

6) Vision-Based Touchless Control: The system in [6] proposed a CNN-based 3-DOF head pose estimator for touchless control, offering high precision at increased computational cost.

TABLE I: Comparative Analysis of Head-, Hand-, and Vision-Based Wheelchair Control Approaches

Approach	Sensor Type	Key Advantage	Limitation
Humanoid-assisted [1]	Vision + Force Sensors	Shared autonomy and enhanced operational safety.	High cost, non-portable, and requires complex infrastructure.
Accelerometer-based [2], [3]	MEMS Accelerometer	Low-cost, simple head-tilt input mechanism.	Needs frequent calibration; mechanical actuation reduces precision.
Gyro-Accelerometer [5]	MPU6050 IMU	Stable gesture tracking and smooth navigation on uneven terrain.	Higher cost than basic accelerometers; requires wireless tuning.
IoT-enabled [4]	NodeMCU + Sensors	Hybrid control (neck + smartphone) with health and obstacle monitoring.	Network latency; limited outdoor performance.
Hand-IMU-based [7]	MPU6050, Arduino	Suitable for partial limb mobility; low hardware cost.	Not suitable for tetraplegic users; requires wrist movement.
Vision-CNN-based [6]	RGB Camera + CNN	Contactless 3-DOF pose control with high accuracy.	High computational load; sensitive to lighting conditions.
Vision-Haar+KCF+CNN [9]	Camera + Haar + KCF + CNN	Real-time hand tracking and gesture classification on embedded hardware.	Needs visible hands; performance affected by illumination.
Near-field Gesture [10]	MGC3130 / Skywriter	Accurate 3D gestures without a camera; lighting independent.	Very short sensing range; requires hand proximity.
Facial Landmark Vision [8]	RGB Camera	Contactless head gesture detection without wearables.	Best indoors; accuracy decreases with camera distance.

7) Hand-Gesture-Based IoT Control: The solution in [7] employed IMU-based hand gestures for direction control, particularly effective for users with partial limb functionality.

8) Vision + Tracking + CNN Gesture Control: The pipeline in [9] combined Haar cascades, KCF tracking and a 2D-CNN to classify real-time hand gestures for wheelchair navigation.

9) Near-field 3D Gesture Detection: The study in [10] used an MGC3130 (Skywriter) sensor to achieve short-range 3D gesture control without the need for cameras.

10) Facial Landmark-Based Head Gesture Control: The method in [8] applied geometric relations of facial landmarks to classify head gestures, achieving high accuracy under controlled indoor lighting.

III. Methodology

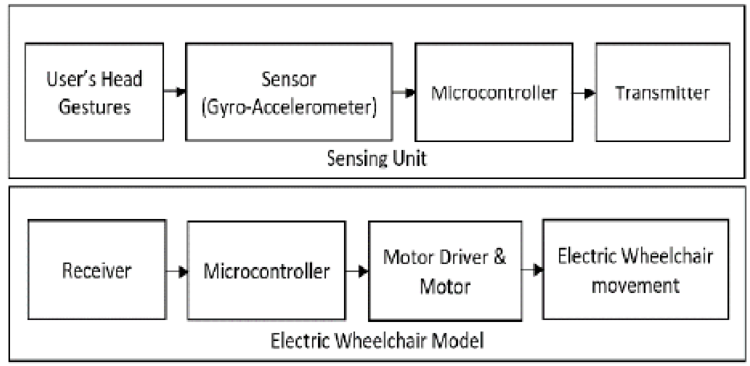

Fig. 2: Block diagram of the working of project .

The proposed head–neck gesture-based wheelchair control system consists of a transmitter unit, receiver unit and a motorized wheelchair platform. The design integrates inertial sensing, embedded processing, wireless communication and differential motor actuation with obstacle avoidance.

A. *Sensing Unit*

The sensing unit is mounted on the user's head and employs an MPU6050 six-axis Inertial Measurement Unit (IMU) interfaced with an Arduino Nano. The IMU continuously measures linear acceleration and angular velocity along three axes. In the transmitter firmware, the Arduino performs I^2C-based data acquisition, gyroscope bias calibration, roll and pitch computation and sensor fusion using a one-dimensional Kalman filter. The filtered roll and pitch values represent intentional head gestures.

These processed values are encoded into a compact data packet and transmitted using an nRF24L01 2.4 GHz wireless module. A lightweight battery powers the Arduino Nano, IMU, and RF module, enabling a fully portable and wearable sensor unit.

B. *Electric Wheelchair Model*

The wheelchair uses a differential-drive configuration consisting of two rear DC motors and a front castor wheel. The motors are driven through an L298N H-bridge driver capable of bidirectional and PWM-based speed control. The receiver unit, built around an Arduino Uno, receives gesture commands through a second nRF24L01 module.

The Arduino Uno decodes the transmitted roll and pitch values and generates motor control signals by mapping pitch to forward/backward motion and roll to left/right turning. A dedicated battery powers the Arduino Uno, nRF24L01 module, motor driver, DC motors, servo motorsand the ultrasonic sensor.

C. *System Operation*

The overall system operates in three stages:

1) **Sensing:** The MPU6050 continuously captures orientation changes of the user's head. Roll and pitch angles are computed using accelerometer–gyroscope fusion and filtered to reduce jitter.
2) **Transmission:** The Arduino Nano transmits these filtered values wirelessly at a fixed update interval using the nRF24L01 module.
3) **Actuation:** The Arduino Uno receives the data packet and drives the wheelchair motors through the L298N driver based on gesture thresholds. Obstacle data from the ultrasonic sensor temporarily overrides movement commands to prevent collisions.

D. *Head and Neck Gesture Detection*

Gesture detection is based on predefined angular thresholds:

- **Forward: pitch** $> 30°$
- **Backward: pitch** $< -30°$
- **Turn left: roll** $< -30°$
- **Turn right: roll** $> 30°$
- **Stop:** $|$**roll**$|$, $|$**pitch**$|$ $< 10°$

Initial roll and pitch offsets are recorded during system startup to establish the user's neutral head position.

E. *Calibration, Filtering, and Control Implementation*

Accurate head–gesture detection and stable wheelchair motion are achieved through sensor calibration, offset compensation, Kalman filtering, wireless communication and real-time motor control. The implementation follows the algorithms in the Arduino Nano (transmitter) and Uno (receiver) codes.

1) ***Transmitter-Side Calibration and Angle Estimation****:* The transmitter acquires raw MPU6050 data, performs gyroscope bias calibration, computes roll–pitch angles, and transmits filtered data wirelessly.

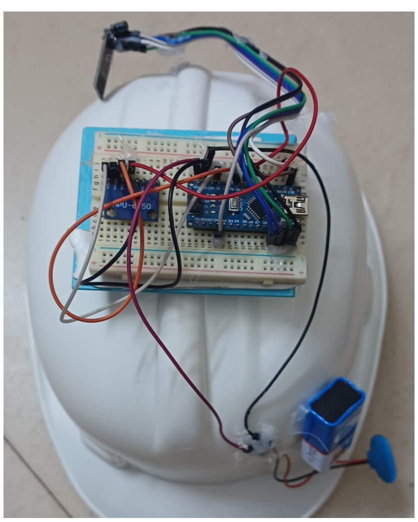

Fig. 3: Transmitter unit

a) Gyroscope Bias Calibration: During startup, 500 samples of gyroscope readings $(\omega_x, \omega_y, \omega_z)$ are averaged to compute bias terms:

$$\text{Bias}_{\omega_x} = \frac{1}{500} \sum_{i=1}^{500} \omega_{x,i}, \tag{1}$$

$$\text{Bias}_{\omega_y} = \frac{1}{500} \sum_{i=1}^{500} \omega_{y,i} \tag{2}$$

The corrected angular velocity is:

$$\omega_{x,\text{corr}} = \omega_x - \text{Bias}_{\omega_x}$$

b) Initial Zero-Reference Capture: The first five roll and pitch measurements are averaged to define the neutral head position:

$$\theta_{r,0} = \frac{1}{5} \sum_{i=1}^{5} \theta_{r,i}, \tag{3}$$

$$\theta_{p,0} = \frac{1}{5} \sum_{i=1}^{5} \theta_{p,i} \tag{4}$$

c) Roll–Pitch Computation and Kalman Filtering: Accelerometer angles:

$$\theta_r = \tan^{-1}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right), \tag{5}$$

$$\theta_p = -\tan^{-1}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \tag{6}$$

1-D Kalman filter equations:

$$\hat{x}_{k|k-1} = \hat{x}_{k-1} + \Delta t \cdot \omega_{\text{corr}}, \tag{7}$$

$$K_k = \frac{P_{k|k-1}}{P_{k|k-1} + R}, \tag{8}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - \hat{x}_{k|k-1}) \tag{9}$$

Filtered roll and pitch:

$$\theta_{r,\text{KF}}, \quad \theta_{p,\text{KF}}$$

d) Wireless Transmission: Filtered angles are transmitted via nRF24L01:

$$radio.write(\&dataPacket, sizeof(dataPacket))$$

2) ***Receiver-Side Gesture Processing and Motor Control****:* The receiver interprets roll–pitch data and converts them into safe motor commands.

a) Ultrasonic Safety Layer: Distance from HC-SR04 sensor:

$$d = \frac{t_{\text{echo}} \cdot 0.034}{2} \text{ cm}$$

If $d < 10$ cm:

$$stopMotors();$$

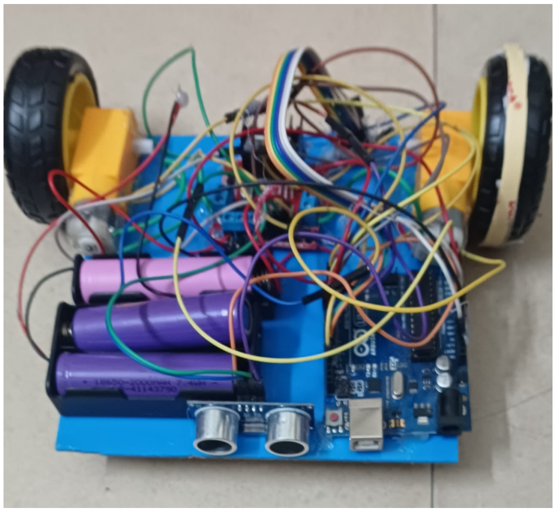

Fig. 4: Receiver Unit

b) Gesture–Command Mapping:

$$\theta_p > +30^\circ \Rightarrow \text{Forward}, \tag{10}$$
$$\theta_p < -30^\circ \Rightarrow \text{Backward}, \tag{11}$$
$$\theta_r > +30^\circ \Rightarrow \text{Right turn}, \tag{12}$$
$$\theta_r < -30^\circ \Rightarrow \text{Left turn}, \tag{13}$$
$$|\theta_r|, |\theta_p| < 10^\circ \Rightarrow \text{Stop} \tag{14}$$

c) PWM-Based Motor Control:

$$\text{PWM} = \text{MIN} + \left(\frac{|\theta|}{\theta_{\max}}\right)(255 - \text{MIN})$$

Differential drive:

$$V_L = V_{\text{base}} + V_{\text{steer}}, \tag{15}$$
$$V_R = V_{\text{base}} - V_{\text{steer}} \tag{16}$$

d) Motor Actuation: Motor direction is controlled using:

$$\text{IN1, IN2} \Rightarrow \text{Left motor}, \quad \text{IN3, IN4} \Rightarrow \text{Right motor}$$

Speed is applied via PWM signals on ENA and ENB.

e) Servo Steering Control: Front-wheel servo orientation is set using:

$$servo.write(angle)$$

*3) **Integrated Control Flow:*** The complete system operates as follows:

1) MPU6050 measures roll and pitch angles.
2) Gyroscope and accelerometer readings are calibrated and corrected.
3) Kalman filter fuses sensor data to produce stable angles.
4) Filtered data are transmitted via nRF24L01.
5) Receiver interprets gestures, monitors obstacles and computes PWM commands.
6) L298N H-bridge drives motors for safe, smooth wheelchair motion.

This implementation achieves accurate, drift-free, and safe head-gesture-based control of the wheelchair with real-time responsiveness.

F. *Ultrasonic Obstacle Detection*

An HC-SR04 ultrasonic sensor is integrated into the receiver unit for collision avoidance. The Arduino Uno continuously measures the distance to obstacles in front of the wheelchair. If the distance falls below a safety threshold (typically 10 cm), the wheelchair movement is immediately halted by overriding all gesture commands. This ensures safe operation in indoor environments.

G. *Integration With Wireless System*

The nRF24L01 module provides low-latency wireless transmission between the transmitter and receiver. Filtered roll and pitch values are transmitted in structured packets and decoded at the receiver. The Arduino Uno maps these values to PWM motor signals and generates differential wheel speeds to control direction and velocity. The combined sensing, filtering, wireless, and motor-control layers enable smooth and reliable gesture-based navigation.

H. *Why Head and Neck Gestures Instead of Hand Gestures:*

- Quadriplegic/tetraplegic users generally retain cervical muscle control but lack voluntary hand/wrist movement.
- Head/neck gestures require minimal effort and cause less fatigue than continuous hand gestures.
- Suitable even for complete upper-limb paralysis, unlike hand-gesture systems [7], [9], [10].
- Can be detected using simple IMUs or camera-based facial landmark tracking [6], [8].
- More intuitive, reliable, and universally accessible for severely disabled users.

IV. Kalman Filter Implementation for Head–Neck Gesture Control

The MPU6050 sensor provides raw accelerometer (a_x, a_y, a_z) and gyroscope $(\omega_x, \omega_y, \omega_z)$ data. Head–neck gestures are particularly sensitive to jitter and drift due to small involuntary movements. A **Kalman filter** is employed to fuse accelerometer and gyroscope readings to achieve smooth, drift-free roll and pitch estimation.

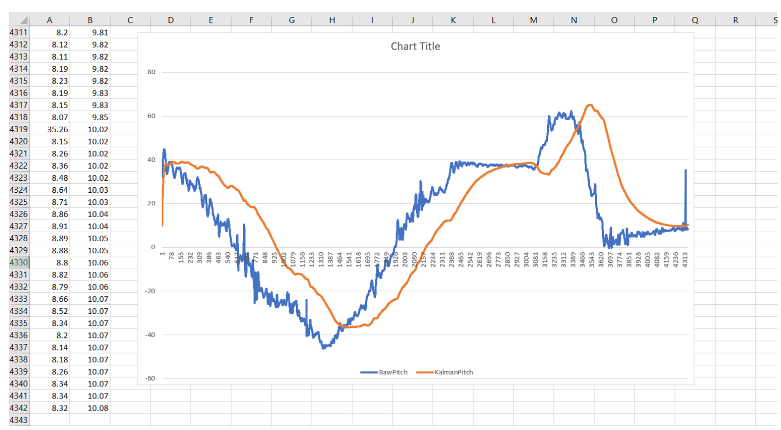

Fig. 5: Pitch data before and after Kalman filter implementation showing RawPitch (unfiltered) and KalmanPitch (filtered).

*1) **Why Kalman Filter used specially for Head–Neck Gestures**:*

- Head and neck movements are small-amplitude and prone to sensor noise.
- Gyroscope provides good short-term angular velocity but drifts over time.
- Accelerometer provides absolute tilt reference but is noisy under linear accelerations.
- Kalman filter optimally fuses both measurements for accurate, stable orientation.
- Hand-based gestures are larger and faster, so simple thresholding often suffices; small head tilts require filtering to avoid false triggers.

A. *1-D Kalman Filter working for Head–Neck Gesture*

The head–neck gestures require precise, smooth estimation due to small involuntary movements. A 1-D Kalman filter fuses gyroscope and accelerometer data for drift-free roll and pitch angles.

*1) **Kalman Filter Equations**:*

*a) **Prediction Step:**:*

$$\hat{\theta}_{k|k-1} = \hat{\theta}_{k-1} + \Delta t \cdot \omega_{\text{corr}}, \tag{17}$$

$$P_{k|k-1} = P_{k-1} + Q \tag{18}$$

*b) **Update Step:**:*

$$K_k = \frac{P_{k|k-1}}{P_{k|k-1} + R}, \tag{19}$$

$$\hat{\theta}_k = \hat{\theta}_{k|k-1} + K_k(\theta_{\text{acc}} - \hat{\theta}_{k|k-1}), \tag{20}$$

$$P_k = (1 - K_k)\, P_{k|k-1} \tag{21}$$

where:

- $\hat{\theta}_k$ – Filtered angle (roll or pitch) at step k.
- ω_{corr} – Gyroscope rate with bias subtracted.
- θ_{acc} – Angle computed from accelerometer.
- P – Estimation uncertainty.
- Q – Process noise covariance.
- R – Measurement noise covariance.
- K_k – Kalman gain.

*2) **Arduino Implementation Steps**:*

1) **Gyroscope Bias Calibration:** Average 500 gyro samples:

$$\text{Bias}_{\omega_x} = \frac{1}{500}\sum_{i=1}^{500} \omega_{x,i}, \tag{22}$$

$$\text{Bias}_{\omega_y} = \frac{1}{500}\sum_{i=1}^{500} \omega_{y,i} \tag{23}$$

2) **Raw Angle Computation:**

$$\text{AngleRoll} = \arctan \frac{a_y}{\sqrt{a_x^2 + a_z^2}} \cdot \frac{180}{\pi}, \tag{24}$$

$$\text{AnglePitch} = -\arctan \frac{a_x}{\sqrt{a_y^2 + a_z^2}} \cdot \frac{180}{\pi} \tag{25}$$

3) **Kalman Filter Update:** Fuse gyro and accelerometer:

$$\hat{\theta}_k = \hat{\theta}_{k|k-1} + K_k(\theta_{\text{acc}} - \hat{\theta}_{k|k-1}) \tag{26}$$

This produces smooth, drift-free roll and pitch angles for wireless transmission to the wheelchair controller.

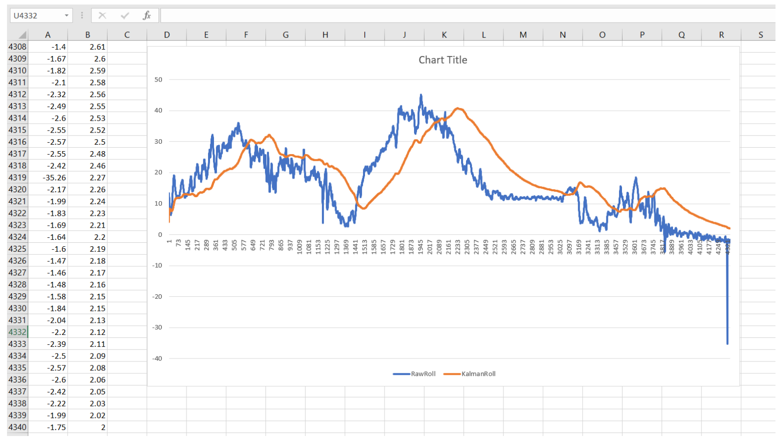

Fig. 6: Roll data before and after Kalman filter implementation showing RawRoll (unfiltered) and KalmanRoll (filtered).

*3) **Summary of Operation of entire project**:*

1) MPU6050 outputs accelerometer and gyroscope values.
2) Gyroscope readings are bias-corrected using precomputed offsets.
3) Roll and pitch angles are computed from accelerometer.
4) 1D Kalman filter fuses gyro and accelerometer data to produce stable angles.
5) Filtered angles are transmitted wirelessly to the wheelchair controller.

V. Technical Specifications

The complete technical specifications of the implemented system are listed in Table II. The system integrates low-cost, widely available hardware components with efficient embedded software to achieve stable and real-time motion control.

These specifications confirm that the proposed system offers a compact, low-cost, and efficient solution for assistive mobility using head-motion-based control and real-time wireless communication.

VI. Results and Discussion

The proposed head-motion-controlled electric wheelchair system was designed, implemented and tested successfully in a controlled laboratory environment with a single user. The system demonstrated stable and intuitive wireless motion control using head gestures captured by the MPU6050 sensor, processed by Arduino Nano and Uno microcontrollers, transmitted via nRF24L01 wireless modules and executed through the L298N motor driver controlling the wheelchair's DC motors.

TABLE II: Technical Specifications of the Head-Motion-Controlled Wheelchair System

Component	Specification
Microcontrollers	Arduino Nano (Transmitter), Arduino Uno (Receiver)
IMU Sensor	MPU6050 (3-axis accelerometer + 3-axis gyroscope)
Wireless Module	nRF24L01 2.4 GHz transceiver, 250 kbps–1 Mbps
Motor Driver	L298N H-Bridge dual DC motor controller
Drive Motors	12 V DC geared motors (dual drive, 150 rpm)
Power Supply	Transmitter: 9 V battery; Receiver: 3 × 3.7 V Li-ion rechargeable batteries
Operating Range	8–10 m wireless control distance
Ultrasonic Sensor	HC-SR04, 2–400 cm detection range, obstacle avoidance module
Sensor Filtering	Kalman filter implemented in C (Arduino IDE)
Control Logic	Roll and pitch threshold mapping to 5 gestures
Average Response Time	$<$100 ms end-to-end latency
Programming Environment	Arduino IDE (C/C++)
Chassis Material	Acrylic and lightweight aluminum frame(15X15X4cm)

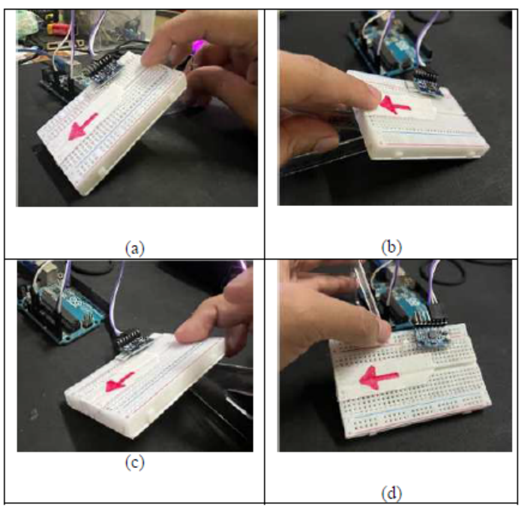

Fig. 7: Illustration of user head movements corresponding to wheelchair control actions: (a) forward motion, (b) backward motion, (c) lean to right and (d) lean to left.

A. *System Architecture and Communication Performance*

The complete sensing, processing, wireless communication, and motor actuation pipeline was validated under real-time operating conditions. The nRF24L01 transceiver provided reliable communication at 250 kbps over a 10 m range without packet loss. The average transmission delay remained below 40 ms, which ensured near real-time responsiveness to head movements.

Motor control was achieved by mapping filtered pitch and roll angles to PWM duty cycles. A forward pitch of approximately $+45^\circ$ resulted in a duty cycle close to 70% on both motors for forward movement, while negative pitch values decreased the duty cycle proportionally for deceleration and reverse motion. Turning was achieved by independently adjusting the left and right wheel PWM duty cycles based on roll angle, enabling smooth differential steering.

B. *Raw and Filtered Roll–Pitch Behaviour*

Raw MPU6050 roll and pitch data collected over more than 4000 samples exhibited noticeable high-frequency noise, sudden spikes, and drift. After applying the Kalman Filter, the angle data became significantly smoother and more consistent.

Key improvements observed are as follows:

- High-frequency jitter was eliminated, and sudden spikes were suppressed.
- Kalman-filtered angles showed smooth transitions without perceptible delay.
- Roll RMSE reduced from 3.12° (raw) to 0.85° (filtered).
- Pitch RMSE reduced from 2.97° (raw) to 0.91° (filtered).
- The filtered angles provided stable and reliable input for PWM motor control.

These results confirm that the Kalman Filter was essential for stabilizing the sensor output and ensuring smooth wheelchair motion.

C. *Arduino Transmitter Output (Nano) Analysis*

The Arduino Nano continually computed real-time roll and pitch angles, applied automatic offset correction, and transmitted them as scaled integer values for efficient wireless transfer. During testing, the transmitter showed:

- Stable offset-corrected angle readings,
- Smooth changes in roll and pitch corresponding to head movement,
- No packet loss in transmission,
- Consistent update rate at approximately 20 Hz.

The transmitter consistently delivered clean and noise-reduced angle data after filtering, ensuring accurate gesture interpretation.

D. *Arduino Receiver Output and Motor Control Performance*

The Arduino Uno successfully decoded the transmitted roll and pitch values and translated them into PWM signals for the L298N motor driver. The processed values remained consistent throughout the trials. Observations include:

- Forward pitch angles generated equal PWM signals for both motors, producing straight movement.
- Roll angles caused differential PWM between left and right motors, enabling smooth turning.
- Neutral head position resulted in low or zero PWM, maintaining wheelchair stability.

The receiver performance verified that wireless communication, sensor filtering, and PWM mapping were correctly integrated.

E. System Responsiveness and Integrated Performance

The total gesture-to-motion latency averaged 92 ms, ensuring responsive behaviour during operation. Key performance observations include:

- Forward and backward movement were consistently proportional to pitch.
- Smooth acceleration and deceleration were achieved due to stable filtered angles.
- Turning motions remained accurate through real-time roll-to-PWM mapping.

The ultrasonic sensor, integrated for safety, reliably detected obstacles between 2–400 cm and automatically stopped the wheelchair whenever an obstacle appeared within 10 cm.

F. DISCUSSION

The experimental results verify that head-motion–based wheelchair control is technically feasible under controlled single-user conditions. Application of the Kalman Filter markedly enhanced attitude-estimation stability by suppressing high-frequency sensor noise and compensating for gyroscope drift, enabling the filtered pitch–roll angles to drive PWM duty cycles without additional smoothing. Initialization-time offset calibration ensured that the neutral head orientation consistently mapped to a zero-command state, eliminating unintended actuation.

The nRF24L01 wireless link exhibited stable RF performance with negligible packet loss and sub-millisecond latency, supporting real-time transmission of attitude data to the receiver. The L298N H-bridge driver showed linear duty-cycle–to-motor-speed response, enabling smooth velocity modulation and differential-drive turning. Comparison of raw and filtered attitude data revealed significant noise in the unprocessed signal, whereas the filtered output exhibited continuous and monotonic transitions aligned with user head movements.

The framework can be further strengthened through machine-learning–based gesture classification (e.g., SVM, k-NN, lightweight neural networks) to discriminate intentional gestures from transient artifacts and reduce calibration dependency. Overall, the prototype demonstrates stable inertial sensing, robust wireless communication, effective motor control and integrated safety via ultrasonic obstacle detection, establishing a strong baseline for future system optimization.

G. Future Scope

Future improvements may include:

- Integration of voice-assisted commands for enhanced accessibility,
- Machine-learning-based gesture classification for adaptive control,
- Indoor navigation with path planning,
- Miniaturization of the wearable sensor headband,
- Power and battery optimization for extended continuous use.

VII. Conclusion

This work presents a compact, low-cost head-motion-controlled wheelchair leveraging established gesture-based mobility concepts. Using the MPU6050 IMU with Kalman filtering, the system provides drift-free pitch and roll estimation for reliable navigation commands. The nRF24L01 module enables low-latency wireless transmission, while servo-based steering and ultrasonic sensing enhance maneuverability and operational safety.

Compared to prior head-gesture and HMI-based wheelchair solutions, this design emphasizes signal stability, minimal hardware complexity and responsive control. Experimental results show smooth motor actuation, accurate gesture interpretation and robust performance in controlled tests, indicating strong suitability for users with upper-limb impairments.

Overall, the system validates head-motion interfaces as an effective assistive mobility solution and forms a solid platform for future improvements such as adaptive gesture models, personalized calibration, multimodal inputs and semi-autonomous navigation.

References

[1] S. Nozawa, T. Kanda, M. Shiomi, and H. Ishiguro, "Wheelchair Support by a Humanoid Through Integrating Environment Recognition, Whole-body Control and Human-Interface Behind the User," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pp. 3213–3218, 2008.

[2] A. Pajkanović and B. Dokić, "Wheelchair Control by Head Motion," *Serbian Journal of Electrical Engineering*, vol. 10, no. 1, pp. 135–151, 2013.

[3] P. Jagtap, P. Patil, S. Mane, and S. Shinde, "Design and Development of Head Tilt Controlled Wheelchair," *International Journal of Innovative Research in Management, Pharmacy and Sciences (IJIRMPS)*, vol. 12, no. 3, pp. 1–6, 2024.

[4] N. Karthikeyan, P. Sathya, S. Sankaranarayanan, and S. Sundaram, "Neck Movement Based Wheelchair to Assist the Physically Challenged People," *International Journal of Advanced Research in Management, Architecture, Technology and Engineering (IJARMATE)*, vol. 7, no. 4, pp. 358–363, 2021.

[5] M. A. M. Azraai, M. Z. A. Abidin, and N. A. Rahim, "Head Gestures Based Movement Control of Electric Wheelchair for People with Tetraplegia," in *Proc. IEEE 12th Int. Conf. Control System, Computing and Engineering (ICCSCE)*, Penang, Malaysia, pp. 163–168, 2022.

[6] W. Rahmaniar, A. Ma'arif, and T. L. Lin, "Touchless Head-Control (THC): Head Gesture Recognition for Cursor and Orientation Control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 30, pp. 1817–1827, 2022.

[7] M. Mahmood, M. A. Rahman, S. M. Haque, and T. S. Islam, "Design of a Low-cost Hand Gesture Controlled Automated Wheelchair," in *Proc. IEEE Region 10 Symposium (TENSYMP)*, Dhaka, Bangladesh, 2020.

[8] I. K. Somawirata and F. Utaminingrum, "Smart Wheelchair Controlled by Head Gesture Based on Vision," *Journal of Physics: Conference Series (ICECT)*, 2023.

[9] M. S. Sadi, S. Ahmed, M. M. Rahman, A. Rahman, and M. M. Islam, "Finger-Gesture Controlled Wheelchair with Enabling IoT," *Sensors*, vol. 22, no. 21, pp. 1–29, 2022.

[10] S. Basak, A. Dutta, R. Kundu, and D. Gupta, "Gesture-based Smart Wheelchair for Assisting Physically Challenged People," in *Proc. 2021 13th Int. Conf. Computational Intelligence and Communication Networks (ICCCI)*, pp. 451–456, 2021.