Journal of Applied Informatics and Computing (JAIC)
Vol.x, No.x, Juli 201x, pp. xx~xx
e-ISSN: 2548-6861

Adaptive Velocity Control Based on IMU Terrain Classification for Campus Delivery Robots

Yudhistira Prasetyo Adios [1]*, Anugerah Wibisana [2]*, Hendawan Soebakti [3]*, Ryan Satria Wijaya [4]*
* Department of Electrical Engineering, Robotics Engineering Technology Study Program, Politeknik Negeri Batam
yp.adios@gmail.com [1]

Article Info

Article history:

Received …
Revised …
Accepted …

Keyword:

Terrain Classification,
Adaptive Velocity Control,
IMU Sensor

ABSTRACT

The ability of autonomous robots to adapt to a wide range of terrains is crucial for operational safety. The study proposes a real-time terrain classification method for delivery robots using MPU6050 IMU sensors to lower costs. Kalman filters are applied as pre-processing to reduce sensor noise, which significantly improves signal quality. The Artificial Neural Network (ANN) model is trained separately for each speed level (0.1–0.5 m/s) to handle vibration variations. The results show models with Kalman filters consistently outperform the raw data with an accuracy above 90%. Although real-time accuracy on ambiguous terrain initially varies between 66%–90%, the implementation of validation three times in a row has significantly improved system performance. Transition testing proves that the system is able to identify terrain changes and adjust the speed smoothly, ensuring navigation stability and safety of the robot's load.

I. INTRODUCTION

The ability of autonomous delivery robots to adapt to the road surface is an important prerequisite for cargo safety and navigation efficiency. When operating in a campus environment, the robot encounters a variety of surfaces such as ceramic corridors, fine ceramics, paving blocks, and asphalt. These changes in terrain produce unique vibration characteristics that can interfere with mechanical stability, especially in differential drive systems without active suspension. Undynamically mitigated over-shocks may cause a risk of damage or spillage of the robot's payload. Therefore, an intelligent mechanism is needed that is able to classify terrain types in real-time to adjust the robot's speed strategy adaptively [1], [2].

Several previous studies have attempted to solve the challenge of vibration signal-based field classification. Sanusi et al. [1] applied the Artificial Neural Network (ANN) architecture utilizing the commercial fusion sensor BNO055 on disaster robots. However, the use of internal fusion sensors such as the BNO055 tends to be black-box, limiting the analysis of raw signal characteristics in depth, and has a higher cost burden. On the other hand, the use of low-cost inertial sensors (IMUs) such as MPU6050 is often constrained by high random noise magnitudes and linear self-vibration interference with increased robot speed [2], [3]. Conditioning of vibration signals from raw data is crucial before being fed into machine learning models [4], [5].

To overcome these limitations, this study proposes a dynamic field classification framework using an economical MPU6050 IMU sensor optimized with the Kalman filter algorithm at the microcontroller level [4]. The main contribution of this study lies in the development of an adaptive speed control strategy based on the ANN architecture, which is trained using a velocity-specific dataset to reduce the ambiguity of vibration features. The performance of the model was evaluated comparatively using two frequency spectrum data-based scenarios, namely raw data and Kalman filtered data. The results of this field detection directly become a controller parameter to execute the speed transition smoothly (accelerating on smooth surfaces and slowing down on rough surfaces such as paving blocks). Through a robust signal pre-processing approach, the system offers a practical and low-cost solution to improve reliability and extend the mechanical life of the transmitter robot in the mobility of dynamic campus environments.

II. METHOD

This research methodology that combines offline and online processes is systematically designed as illustrated in the workflow in Figure 1.

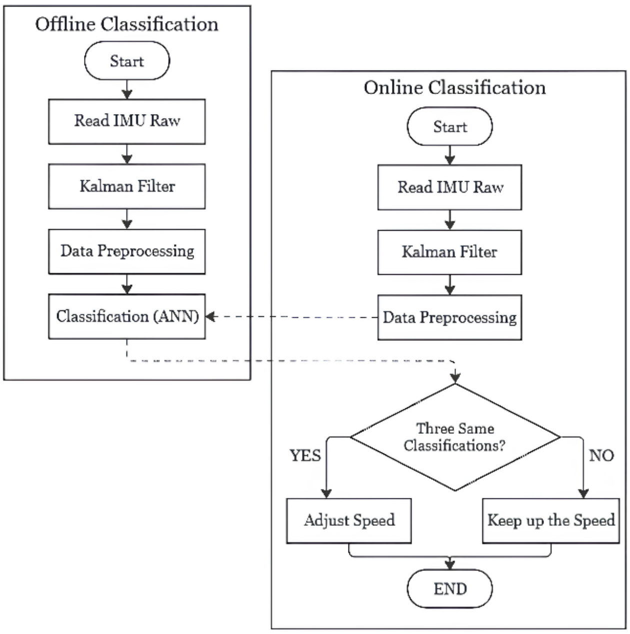


Figure 1. Flowchart

This study uses a two-phase approach to classify the types of terrain, namely offline classification and online classification (Figure 1). In the first phase, raw data of accelerometers (x,y,z) and gyroscopes (x,y,z) were collected from robots moving in various terrains. This raw data is then processed using the kalman filter to reduce noise and improve signal accuracy, a practice that is in line with the research of Setiawan et al. [4][5]. The raw data and filtered data are then used to train the Artificial Neural Network (ANN) classification model. Model evaluation showed that although models trained with raw data achieved good accuracy, models using kalman filter results showed significant performance improvements, consistent with findings in the Setiawan et al.

A. Design Robot and Specification

This delivery robot is designed by adopting a rectangular box shape to maximize the internal volume of the load while maintaining motion stability. The design of this structure bears a resemblance to the navigation platform developed by previous researchers [6], but is further optimized for campus mobility needs through the use of lightweight yet sturdy materials to support energy efficiency. To provide an in-depth visual understanding of the physical configuration and integration of its mechanical system, the details of the robot's structural design are presented in Figure 2.

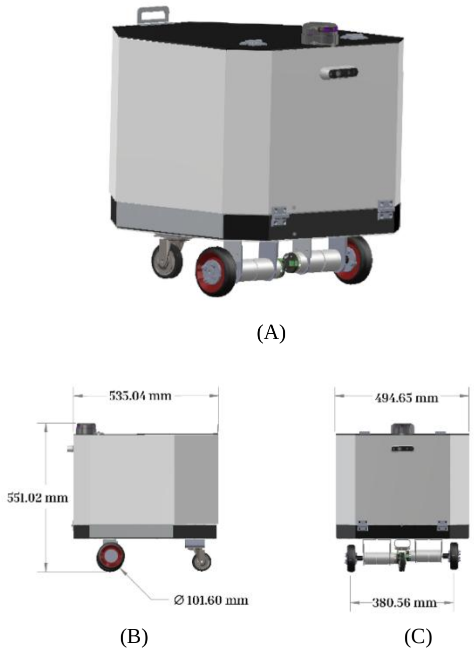


Figure 2. Design Robot. (A): Isometric view. (B): Front view. (C): Side View

A comprehensive visualization of the robot's mechanical structure in its entirety is shown in Figure 2A. This section presents an isometric view that serves to provide a clear three dimensional perspective of the layout of the placement of external components, such as the position of sensors and actuators, thus facilitating an understanding of the overall physical configuration of the system.

In more detail, Figure 2B shows that the robot has a maximum width of 494.65 mm with a wheelbase of 380.56 mm on the front display. Meanwhile, the dimensions of the robot's body as seen from the side view are shown in Figure 2C, which has a length of 535.04 mm and a total height of 551.02 mm. The movement system is supported by a 101.60 mm (4 inch) diameter rear wheel, where the specification of these dimensions is chosen to guarantee the safety of the load during the robot's operation.

B. Experimental Setup

System testing is conducted using a distributed hardware architecture to ensure that the Kalman Filter and Artificial Neural Network (ANN) algorithms can process high-frequency sensor data with minimal latency. This approach aims to validate the synchronization between hardware and software capabilities to meet real-time operational needs. High-level computing for classification and control is allocated to the Main Controller laptop (AMD Ryzen 7 PRO 3700U, 16 GB of RAM), while the data acquisition and low-level actuator control tasks are handled by the 600 MHz clocked Teensy 4.1 (Sub-Controller) microcontroller.

TABEL I
SPECIFICATION HARDWARE

Component	Specification	Function
Main Controller	AMD Ryzen 7 PRO 3700U @ 2.3 GHz, 16GB RAM, 239GB NVMe SSD	ANN classification & main control algorithms
Slave Controller	Teensy 4.1 (ARM Cortex-M7 @ 600 MHz)	Real-time sensor sampling & motor control
IMU Sensor	MPU6050 (3-axis Gyroscope + 3-axis Accelerometer)	Robot motion & orientation tracking
Motor Driver	BTS7960	PWM power drive for actuators

In its workflow, Teensy 4.1 continuously samples orientation data from MPU6050 IMU sensors via the I^2C interface, and then sends the telemetry to the laptop via high-speed USB Serial communication. After the classification algorithm on the laptop processes the data, the laptop sends the robot's speed target command in m/s. Teensy 4.1 then receives the m/s data and performs calculations to convert it into a PWM control signal to drive the PG45 DC motor through an H-bridge BTS7960 driver supplied by a 24 VDC battery. Full details of the hardware specifications are summarized in Table I.

C. Dataset Collection

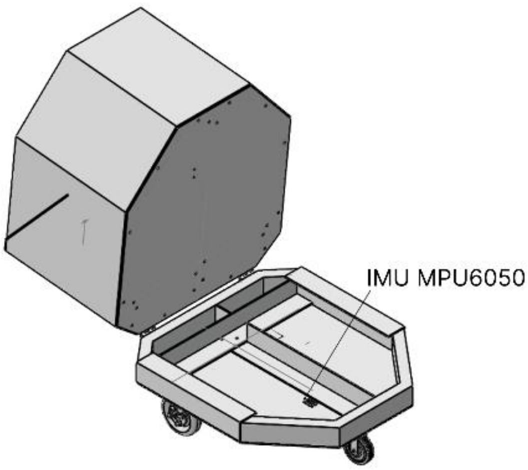


Figure 3. IMU Sensor Placement

The data acquisition process is carried out with an MPU6050 IMU sensor mounted in the center of the robot's body, strategically positioned to minimize the influence of asymmetric vibrations from the differential drive motor as seen in Figure 3. This sensor measures accelerometer (3 axis) and gyroscope (3 axis) data with a sampling frequency of 100 Hz.

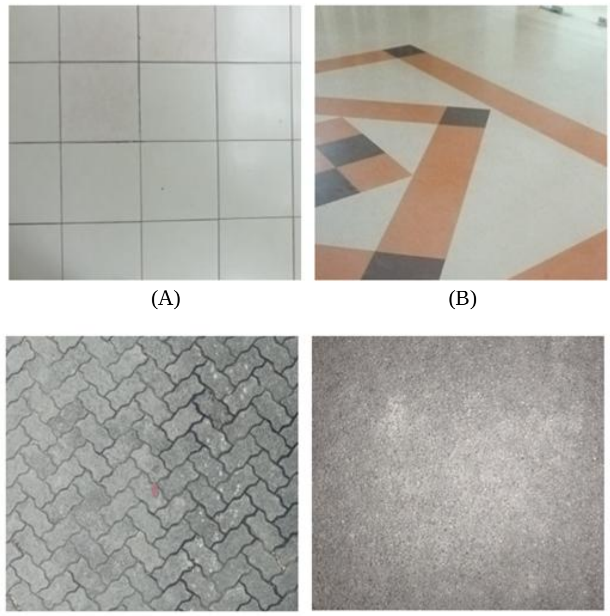


Figure 4. Terrain Type. (A): Corridor ceramic. (B): Smooth Ceramic. (C): Paving Blocks. (D): Asphalt.

The data collection process was carried out on four types of terrain in the campus area as well as one stationary condition (not moving), as illustrated in Figure 4. Specifically, the characteristics of the terrain surface tested include corridor ceramics in Figure 4A, smooth ceramics in Figure 4B, paving blocks in Figure 4C, and asphalt in Figure 4D. In this study, the robot was operated at five different speed levels, ranging from 0.1 m/s to 0.5 m/s, with a total of 9,600 samples recorded for each combination of parameters [2].

The main reason for collecting data at different speeds is to overcome significant vibration variations in the same terrain due to speed differences. These differences can make the classification model too complex which makes the model confused, as different patterns of one type of terrain can have many variations. By collecting data at different speeds, we can train one classification model for one type of speed this will be more powerful and able to adapt to changes in the speed of the robot. Idle conditions are also included as an important class label, ensuring the model can identify when the robot is stationary.

The analysis below comparing accelerometer data and the gyroscope of the corridor ceramic field was chosen as an example because of its relatively stable charac-teristics, so that the difference in impact of each speed increase can be seen more clearly. The visualization results show that the increase in robot speed directly af-fects the frequency and amplitude of data spikes caused by ceramic connectors.

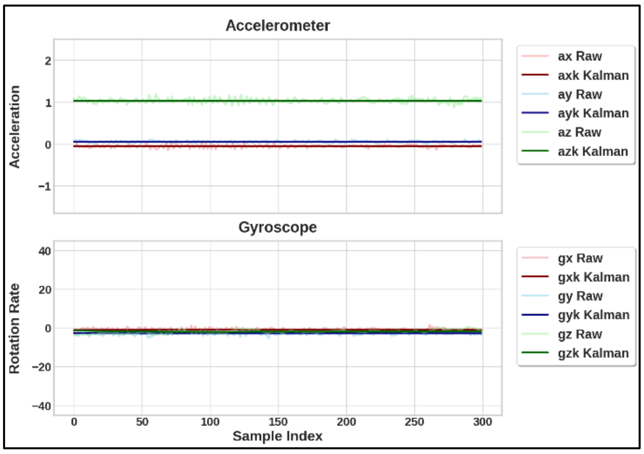


Figure 5. Corridor Ceramic 0.1 m/s

Figure 5 illustrates the accelerometer and gyroscope data at a speed of 0.1 m/s which shows relatively stable conditions with minimal fluctuations. In most time windows, the data almost resembles a straight line, confirming that at low speeds, the influence of ceramic surface imperfections is almost insignificant. However, in certain windows, there is a significant spike that is characteristic of the corridor ceramic where there is a barrier between ceramic tiles. These occasional fluctuations provide important information about the texture of the terrain the robot is traversing.

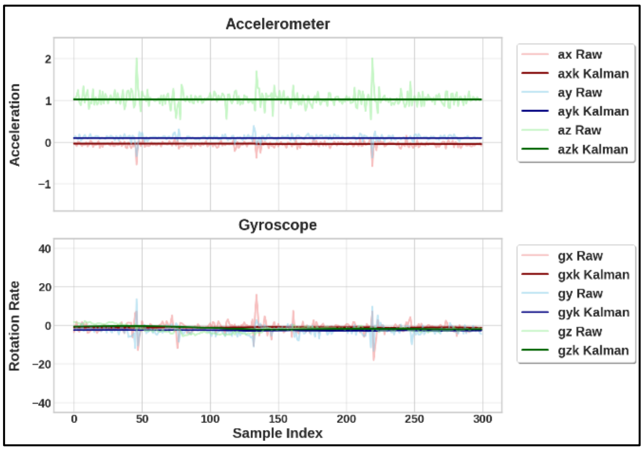


Figure 6. Corridor Ceramic 0.5 m/s

Figure 6 shows that at the highest speed of 0.5 m/s, the spike that occurs becomes more and more dense. This happens because the higher speed of the robot shortens the time interval between the meeting of the wheels and the barrier between tiles (tile connector). The faster the robot moves, the denser the surge pattern, which reflects the higher vibration frequency.

TABEL II
NUMBER OF DATA SAMPLE

Speed	Details per Field	Amount of data
0 m/s	9.600 (Not Moving)	9,600
0.1 m/s	9.600 sample per terrain	48,000
0.2 m/s	9.600 sample per terrain	48,000
0.3 m/s	9.600 sample per terrain	48,000
0.4 m/s	9.600 sample per terrain	48,000
0.5 m/s	9.600 sample per terrain	48,000
Total		249,600

Table 2 presents a summary of the datasets used to train a single classification model. Each combination of terrain type and speed has a balanced total of 9,600 data samples except for the 'Not Moving' condition that only has one data set with 9,600 samples because this condition is not affected by speed. Thus, the training and testing dataset has a total of 249,600 samples, which are evenly distributed among the five terrain classes. This balanced dataset structure is critical to preventing model bias and ensuring that models can learn effectively to identify each terrain class.

D. *Simple Kalman Filtering*

To improve the quality of signal data from IMU (Inertial Measurement Unit) sensors used in terrain type classification, this study implements the kalman filter algorithm [4]. This process is very important because the raw data from the sensor is often interrupted by noise or noise. This noise mainly comes from real world variations, such as imperfections of the road surface, as well as the condition of the robot itself that is not equipped with suspension, which directly generates high vibrations and noise that can affect the accuracy and stability of the system.

By filtering the raw data, we can derive more reliable value for real-time analysis and decision making by robots. The implementation of kalman filter in this project uses optimized parameter configurations to balance system responsiveness and stability. The parameter used is SimpleKalmanFilter (2, 2, 0.01), where a process noise value of 2 and a measurement noise of 2 are specifically selected to filter out high-frequency interference without causing significant lag, which is vital for real-time applications [4]. Meanwhile, an estimatedError of 0.01 ensures that the estimates generated by the filter have a high level of accuracy.

Mathematically, the kalman filter algorithm works by predicting the state of the system and updating the forecast based on new measurements. The renewal process involves three primary steps: calculating the Kalman Gain K_k, updating the current forecast \widehat{X_k}, and updating the estimated current error P_k.The implementation of these steps is defined by the following formulas:

$$K_k = \frac{P_{k-1} + Q}{P_{k-1} + Q + R} \tag{1}$$

On the Equation (1), K_k\ (Kalman Gain) is the weight given to the new measurement data. The higher the value, the greater the trust in the sensor data. P_{k-1}\ (Predicted Past Error) is the degree of uncertainty or error from previous predictions. Q\ (Process Noise) is uncertainty or noise originating from the system model itself. R (Measurement Noise) is Noise that comes directly from the sensor.

$$x_k = x_{k-1} + K_k(z_k - x_{k-1}) \tag{2}$$

Equation (2) above is used to calculate $\widehat{X_k}$ (Current Estimate) is the final value that has been updated and more accurate. $\mathrm{x}_{\mathrm{k}-1}$(Previous Estimate) is Prediction of the value of the previous time step.z_{k} (New Measurement): Newly received raw data from the sensor.

$$P_k = (1 - K_k)P_{k-1} \tag{3}$$

In Equation (3), a calculation is made to update P_k (Estimated Current Error) i.e. the value of the covariance of the error that has been updated. This update gradually reduces system uncertainty so that the estimated results are more accurate.

The following analysis compares the raw signal data from the accelerometer and gyroscope sensors with the data that has been processed using the kalman filter. I took this comparison as an example of the paving blocks field at a speed of 0.5 m/s where the comparison is clearly visible. This visualization clearly shows the effectiveness of the algorithm in reducing noise in the signals coming from each field.

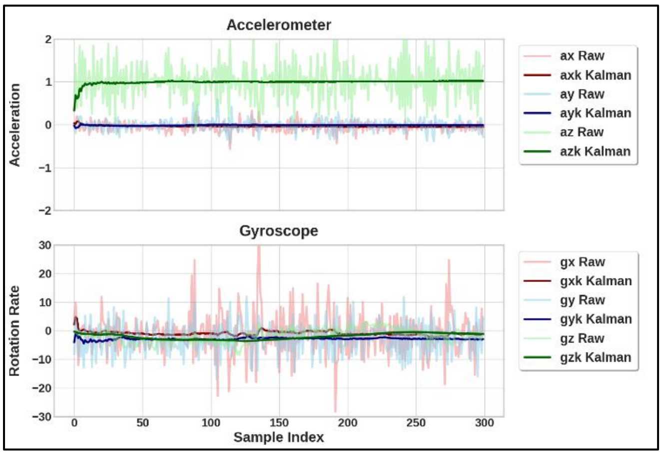


Figure 7. Paving Blocks Type

Figure 7 data from the paving blocks terrain shows more frequent and varied fluctuations. This reflects the uneven nature of the terrain, where each paving blocks can have a different height and cause changes in the direction of facing (gyroscope). Despite strong fluctuations in the raw data, the kalman filter manages to reduce noise and create smoother trend lines. These results show that the filter is able to isolate fundamental movements from random vibrations, which is crucial for detecting actual field changes. These subtle signals help the neural network extract the unique features of the paving block field, resulting in a more reliable classification.

E. Data Preprocessing

Before being used to train the model, the IMU sensor data undergoes pre-processing to optimize signal representation. First, the raw data is standardized using the StandardScaler so that all features are uniformly scaled, preventing large-range features from dominating training. Furthermore, the signal is segmented into a fixed 150-sample window to capture the complete vibration cycle [1][2][3]. Most importantly, Fast Fourier Transform (FFT) converts signals from time to frequency domains, where the field vibration patterns differ [1]. In addition, the Root Mean Square (RMS) is calculated to represent the signal energy. Finally, the dataset was divided using a balanced 80:20 train test ratio for effective validation.

Based on the reference in previous research [1], the feature extraction process is carried out using the Fast Fourier Transform (FFT) algorithm to convert signals from time domains to frequency domains. The general formula of FFT can be written as:

$$X(f) = \sum x_k \cdot e^{-il\Delta\omega k} \tag{4}$$

$$= \sum x_k \cdot e^{-il2\pi fk} \tag{5}$$

$$= \sum x_k \cdot e^{-ilk\frac{2\pi}{N}} \tag{6}$$

Equations (4), (5), and (6) represent the process of transforming a signal from the time domain to the frequency domain to extract the characteristics of the sensor data spectrum. Where X is the value of the transformation results in the frequency domain, x_k is sampling values to (-k) of the time domain, *N* is the number of FFT points, l is discrete frequency index, k is the time index and *f* is frequency. In this study, the value of, n_{\mathrm{fft}}=\ 256 was used so that every channel produces 256/2 =\ 128 FFT magnitude components representing half of the frequency spectrum (due to the symmetrical nature of FFT for real signals). With six channels (3 accelerometer axes and 3 gyroscope axes), the number of features per window remains at 768 because both types of data have the same six channels. This configuration ensures compatibility with the ANN model architecture in,

e-ISSN: 2548-6861

facilitates performance comparison, and calculates overall accuracy directly at the model evaluation stage.

Spectrum analysis via Fourier transform is essential to identify specific vibration patterns of wheel and road surface interactions. By extracting frequency features from Kalman's filtered sensor data, the system is able to distinguish random noise from field texture signals (such as paving blocks or asphalt). This synergy captures spectral characteristics that are invisible in the time domain, prevents overfitting, and ensures ANN produces consistent and stable classifications during real-time operation.

F. ANN Model Architecture

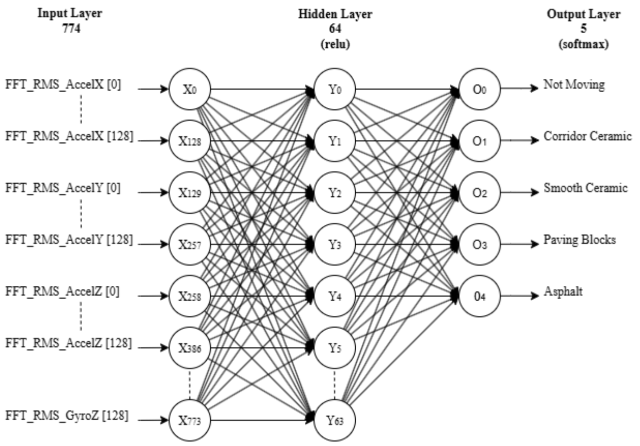


Figure 8. Neural network architecture for accelerometer-gyroscope input

Figure 8 illustrates the Artificial Neural Network (ANN) architecture designed in this study to process features in the frequency domain, similar to the approach described in [1].Total feature = 6 x 128(FFT) + 6 x 1 (RMS) = 774 per window. The input layer receives a total of 774 features for each of the six sensor channels (three accelerometer axes and three gyroscope axes). Based on reference [1], the early architecture used only a hidden layer with 32 neurons. However, these configurations proved to be less accurate when tested at various speed variations, requiring an increase in model representation capacity.

On the core layer of the architecture, a single hidden layer of 64 neurons equipped with the Rectified Linear Unit (ReLU) activation function is used to better capture nonlinear patterns. The output layer uses a softmax activation function with the number of neurons corresponding to the number of field classes, resulting in a multiclass classification probability. The optimized model uses the Adam algorithm with a learning rate of 0.001 and a loss categorical crossentropy function, which has been shown to provide more stable and accurate classification results than the 32 neuron architecture.

G. Adaptive Velocity Strategy for Robot Navigation

The robot's operational speed is adjusted to the characteristics of the terrain surface to maintain a balance between efficiency and load safety. In fine ceramic fields with minimal vibration, the robot operates at a maximum speed of 0.5 m/s. In ceramic and asphalt corridors with light vibrations, the speed was lowered to 0.4 m/s to dampen the shock. Meanwhile, in uneven and high-vibration paving blocks, the speed is limited to 0.3 m/s to prioritize stability and operational safety.

To ensure a smooth transition of speed between terrain types, the system applies gradual adjustment. For example, when fine ceramics are detected from an initial speed of 0.2 m/s, the speed is gradually increased by 0.1 m/s until it reaches the target of 0.5 m/s. During the transition, the system continuously verifies the type of terrain to prevent switching errors and maintain the consistency of the robot's movement.

III. RESULT

This section presents the results of testing the model performance in three stages: comparison of raw data accuracy vs Kalman filters, validation of real-time detections, and analysis of the stability of model predictions as terrain type transitions occur at various speed levels.

A. Training Results

TABEL III
CONFUSION MATRIX RESULT TRAINING RAW AND KALMAN 0.1 M/S

V = 0.1 m/s		**Predict terrain**				
Actual Terrain		Not Moving	Corridor Ceramic	Smooth Ceramic	Paving Block	Asphalt
	Not Moving	100% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Corridor Ceramic	7.69% **(0%)**	92.31% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Smooth Ceramic	0% **(15.38%)**	0% **(0%)**	100% **(84.62%)**	0% **(0%)**	0% **(0%)**
	Paving Block	7.69% **(0%)**	15.38% **(0%)**	0% **(0%)**	76.92% **(100%)**	0% **(0%)**
	Asphalt	0% **(0%)**	0% **(0%)**	7.69% **(0%)**	0% **(0%)**	92.31% **(100%)**
Acc Raw		: 92.19%				
Acc Kalman		: 96.88%				

Table 3 shows that at a speed of 0.1 m/s, the model using the Kalman filter achieved an accuracy of 96.88%, higher than the raw data model which only reached 92.19%.

TABEL IV
CONFUSION MATRIX RESULT TRAINING RAW AND KALMAN 0.2 M/S

V = 0.2 m/s		**Predict terrain**				
		Not Moving	Corridor Ceramic	Smooth Ceramic	Paving Block	Asphalt
Actual Terrain	Not Moving	100% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Corridor Ceramic	0% **(0%)**	100% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Smooth Ceramic	0% **(7.69%)**	0% **(0%)**	92.31% **(92.31%)**	0% **(0%)**	7.69% **(0%)**
	Paving Block	7.69% **(0%)**	0% **(0%)**	0% **(0%)**	92.31% **(100%)**	0% **(0%)**
	Asphalt	0% **(0%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**	100% **(100%)**
Acc Raw	: 96.88%					
Acc Kalman	**: 98.44%**					

Based on Table 4, the Kalman filter increases the accuracy from 96.88% to 98.44%, with a 100% success rate on the recognition of ceramic and fine asphalt surfaces.

TABEL V
CONFUSION MATRIX RESULT TRAINING RAW AND KALMAN 0.3 M/S

V = 0.3 m/s		**Predict terrain**				
		Not Moving	Corridor Ceramic	Smooth Ceramic	Paving Block	Asphalt
Actual Terrain	Not Moving	100% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Corridor Ceramic	0% **(0%)**	76.92% **(92.31%)**	23.08% **(7.69%)**	0% **(0%)**	0% **(0%)**
	Smooth Ceramic	0% **(0%)**	7.69% **(0%)**	92.31% **(100%)**	0% **(0%)**	0% **(0%)**
	Paving Block	0% **(0%)**	23.08% **(0%)**	7.69% **(7.69%)**	69.23% **(92.31%)**	0% **(0%)**
	Asphalt	0% **(0%)**	0% **(0%)**	0% **(15.38%/)**	0% **(0%)**	100% **(84.62%)**
Acc Raw	: 87.50%					
Acc Kalman	**: 93.75%**					

Table 5 shows the Kalman filter increases the accuracy from 87.50% to 93.75% by addressing the vibration ambiguity between the ceramic corridor and the paving block.

TABEL VI
CONFUSION MATRIX RESULT TRAINING RAW AND KALMAN 0.4 M/S

V = 0.4 m/s		**Predict terrain**				
		Not Moving	Corridor Ceramic	Smooth Ceramic	Paving Block	Asphalt
Actual Terrain	Not Moving	100% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Corridor Ceramic	0% **(0%)**	76.92% **(84.62%)**	15.38% **(15.38%)**	7.69% **(0%)**	0% **(0%)**
	Smooth Ceramic	0% **(0%)**	0% **(0%)**	100% **(100%)**	0% **(0%)**	0% **(0%)**
	Paving Block	0% **(0%)**	23.08% **(0%)**	7.69% **(7.69%)**	69.23% **(92.31%)**	0% **(0%)**
	Asphalt	0% **(0%)**	0% **(7.69%)**	0% **(0%)**	0% **(0%)**	100% **(92.31%)**
Acc Raw	: 89.06%					
Acc Kalman	**: 93.75%**					

As summarized in Table 6, the accuracy of the raw data model of 89.06% was successfully increased to 93.75% through the implementation of the Kalman filter.

TABEL VII
CONFUSION MATRIX RESULT TRAINING RAW AND KALMAN 0.5 M/S

V = 0.5 m/s		**Predict terrain**				
		Not Moving	Corridor Ceramic	Smooth Ceramic	Paving Block	Asphalt
Actual Terrain	Not Moving	100% **(100%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**	0% **(0%)**
	Corridor Ceramic	0% **(0%)**	100% **(92.31%)**	0% **(7.69%)**	0% **(0%)**	0% **(0%)**
	Smooth Ceramic	0% **(0%)**	7.69% **(0%)**	92.31% **(100%)**	0% **(0%)**	0% **(0%)**
	Paving Block	0% **(0%)**	15.38% **(0%)**	0% **(0%)**	84.62% **(100%)**	0% **(0%)**
	Asphalt	0% **(0%)**	7.69% **(7.69%)**	0% **(0%)**	0% **(0%)**	92.31% **(92.31%)**
Acc Raw	: 93.75%					
Acc Kalman	**: 96.88%**					

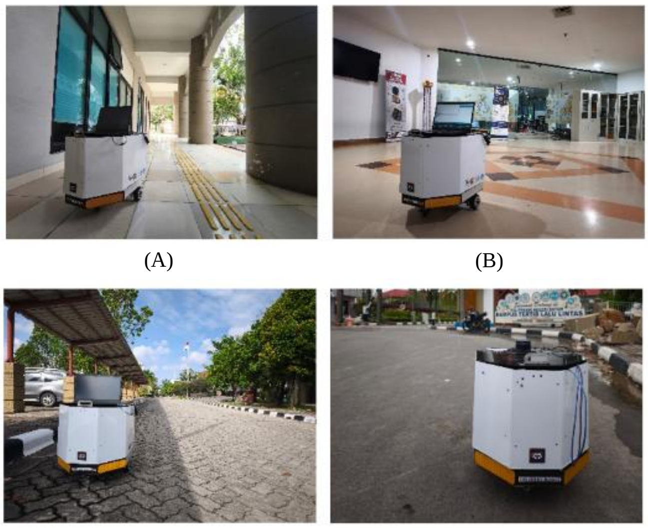
At a maximum speed of 0.5 m/s, Table 7 shows an increase in accuracy from 93.75% to 96.88% with the Kalman data. Even though the sensor's noise intensity is at its maximum,

e-ISSN: 2548-6861

the model is still able to classify the ceramic corridor and smooth ceramic fields with a high degree of accuracy.

Overall, these data reinforce the conclusion that pre-processing with Kalman filters is highly recommended to maintain the stability of the terrain classification system. This approach ensures the model remains resilient to sensor interference and provides reliable predictions at various operational speeds.

B. Real-time Terrain Classification Performance



(A) (B) (C) (D)

Figure 9. Online terrain classification testing on several terrains. (A): Corridor Ceramic. (B): Smooth Ceramic. (C): Paving Blocks. (D): Asphalt.

Figure 9 documents real-time field classification testing to validate model performance under operational conditions at four locations. This test collected 30 consecutive predictions as the robot traversed Corridor Ceramics (Figure 9A) for standard surfaces and Fine Ceramics (Figure 9B) for smoother textures. Outdoor testing included Paving Blocks (Figure 9C) to evaluate resistance to rough vibration patterns and Asphalt (Figure 9D) to validate accuracy at different levels of road roughness. All predictions are calculated to determine the model's percentage of direct field detection accuracy.

This real-time test data is crucial to proving the effectiveness of signal preprocessing. With the Kalman filter data, the model is expected to be able to maintain predictive stability despite environmental noise disturbances in the field. The consistency of results between laboratory and field tests is a key indicator of the success of the classification system in supporting the safe navigation of robotic autonomous navigation.

TABEL VIII
CONFUSION MATRIX IMPLEMENTATION REALTIME RESULT

Type of Terrain		**Predict terrain**				
		Not Moving	Corridor Ceramic	Smooth Ceramic	Paving Block	Asphalt
Actual Terrain	Not Moving	**100%**	0%	0%	0%	0%
	Corridor Ceramic	6.67%	**70.0%**	16.67%	3.33%	3.33%
	Smooth Ceramic	23.33%	10.0%	**66.67%**	0%	0%
	Paving Block	0%	0%	0%	**90.0%**	10.0%
	Asphalt	3.33%	0%	6.67%	6.67%	**80.0%**

Based on the data from Table 8, the results of real-time testing show how the system performs under real operational conditions. The accuracy for each field varies. The system successfully detected the not moving condition with perfect accuracy, which is 100%. This signifies that the robot is very reliable when stationary. However, performance on other terrain is not always ideal. For corridor ceramic, the accuracy is 70.0%, with prediction errors that occur frequently in smooth ceramic and other terrains. This shows that the vibration characteristics of corridor ceramics have similarities with other terrains.

The lowest evaluation results were recorded in the test in the smooth ceramic field, where the accuracy level achieved was only 66.67%. The most dominant prediction failure in this area occurs when the system identifies the terrain as a stationary condition (not moving) or as corridor ceramic. This phenomenon indicates a significant technical challenge in differentiating the vibration pattern between a very fine surface and a robot's static state. On the other hand, the system showed very impressive performance on paving block terrain with an accuracy of 90.0%, where only a few misclassification errors were found on the asphalt terrain. Meanwhile, testing on asphalt surfaces yielded an accuracy rate of 80.0%, with some predictive error records categorizing them into fine ceramic groups or paving blocks. This misclassification indicates the need for algorithm refinement and optimization to improve precision, especially in fields with similar or ambiguous vibration characteristics.

This variation in accuracy presents a major challenge in separating vibration features on similar smooth surfaces. The inability to distinguish fine ceramics from the stationary state reflects a low vibration threshold that is difficult for sensors to read.

C. *Analyzing System Performance During Terrain Changes*

Figure 10. Online terrain transition test classification

To evaluate the performance of the system in real-time, we also perform field transition testing. This test simulates a scenario in which the robot moves from a smooth ceramic surface to a paving blocks (Figure 10). Here is a visualization of the test results :

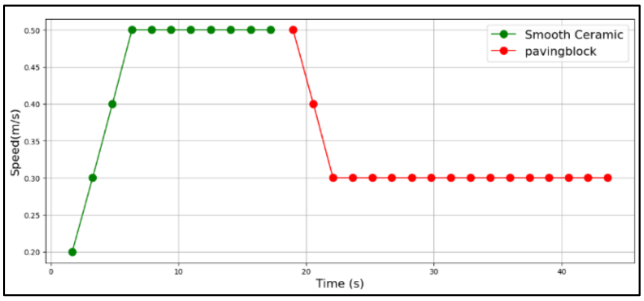


Figure 11. Speed transition based on terrain classification

Figure 11 above visualizes the robot's speed response to changes in terrain type during real-time testing. The X-axis represents the time (seconds) and the Y-axis represents the robot's speed (m/s). This test focuses on the transition from smooth ceramic fields to paving blocks.

In the initial stage, the robot's speed is set at 0.2 m/s. After the terrain classification successfully identified the smooth ceramics, the control system adaptively increased the robot's speed towards the target of 0.5 m/s. This increase is done in stages with an increase of 0.1 m/s in each step, where each classification takes approximately 1.5 seconds. This becomes a limitation in real-time applications because robots take time to classify and respond, which can lead to delays. Nonetheless, this approach was chosen to ensure a smooth speed transition, which is critical to maintaining the stability of the robot and the safety of the payload it carries.

After reaching a speed of 0.5 m/s, the robot maintains this speed as long as it is in the smooth ceramic field. Once the classification model detects a transition to the paving blocks, the control system immediately responds by gradually decreasing the speed. This decrease was carried out until it reached the speed target that had been set for the paving blocks terrain, which was 0.3 m/s. This calibrated speed response demonstrates the system's ability to adapt to different terrain conditions, making it an efficient solution for autonomous robot navigation.

IV. CONCLUSION

This study successfully proves that the integration of Kalman filters and speed-based training models is able to effectively address vibration variations in autonomous robots. The use of Kalman filters as a pre-processing stage proved crucial in improving the signal quality of the IMU, so that the model consistently exceeded the performance of raw data at all speed levels. Although classification challenges are still found on terrains with similar characteristics such as corridor ceramic and smooth ceramic, the implementation of three iterative validations successfully mitigated these constraints and significantly improved system accuracy. With a real-time processing time of approximately 1.5 seconds, the system demonstrates sufficient efficiency for practical applications in maintaining the navigation stability and safety of robot loads, although further development is still needed to improve accuracy on terrain with ambiguous vibration characteristics.

ACKNOWLEDGMENT

The author expresses his infinite gratitude to the Robot Delivery Team for their essential support, contribution, and teamwork in improving the quality and success of this research project. We also express our highest appreciation to the Batam State Polytechnic for facilitating and funding this research within the framework of Project Based Learning (PBL). Special thanks were also conveyed to the mentor as a place for in depth discussion and technical assistance in overcoming the challenges of the experimental phase.

REFERENCES

[1] Sanusi, M. A., Dewantara, B. S. B., Setiawardhana, & Sigit, R. (2023, Februari 28). Online terrain classification using neural network for disaster robot application. The Indonesian Journal of Computer Science, 12(1). https://doi.org/10.33022/ijcs.v12i1.3132

[2] Wang, M., Ye, L., & Sun, X. (2021). Adaptive online terrain classification method for mobile robot based on vibration signals. International Journal of Advanced Robotic Systems, 18(6), Article 17298814211062035. https://doi.org/10.1177/17298814211062035

[3] Hu, Y. (2025). Robotic terrain classification based on convolutional and long short-term memory neural networks. Cognitive Robotics, 5, 166–175. https://doi.org/10.1016/j.cogr.2025.04.002

[4] Setiawan, R., Triharminto, H. H., & Fahrurozi, M. (2021). Gesture Control Menggunakan IMU MPU 6050 Metode Kalman filter Sebagai Kendali Quadcopter. Dalam Prosiding Seminar Nasional Sains Teknologi dan Inovasi Indonesia (SENASTINDO), Vol.3,hlm. 411–422. https://doi.org/10.54706/senastindo.v3.2021.133

[5] Alfian, R. I., Ma'arif, A., & Sunardi, S. (2021). Noise reduction in the accelerometer and gyroscope sensor with the Kalman filter algorithm. Journal of Robotics and Control (JRC), 2(3), 180–190. https://doi.org/10.18196/jrc.2375

[6] Simanjuntak, W. P. S., & Wibisana, A. (2024). Depth camera-based human detection using YOLOv5. In Proceedings of the 7th International Conference on Applied Engineering (ICAE 2024) (pp. 150–162). Atlantis Press. https://doi.org/10.2991/978-94-6463-620-8_12

[7] Elmaz, F., Eyckerman, R., Casteels, W., Latré, S., & Hellinckx, P. (2021). CNN-LSTM architecture for predictive indoor temperature modeling. Building and Environment, 206, 108327. https://doi.org/10.1016/j.buildenv.2021.108327

[8] Lee, C. J., & Lee, J. K. (2024). IMU-Based Energy Expenditure Estimation for Various Walking Conditions Using a Hybrid CNN–LSTM Model. Sensors,24(2), 414.https://doi.org/10.3390/s24020414

[9] Coelho, R. M., Gouveia, J., Botto, M. A., Krebs, H. I., & Martins, J. (2022). Real-time walking gait terrain classification from foot-mounted Inertial Measurement Unit using Convolutional Long Short-Term Memory neural network. Expert Systems with Applications, 203, 117306. https://doi.org/10.1016/j.eswa.2022.117306

[10] Xue, F., Hu, L., Yao, C., Liu, Z., Zhu, Z., & Jia, Z. (2022, July). Sound-Based Terrain Classification for Multi-modal Wheel-Leg Robots. In 2022 IEEE Conference on Robotics and Automation (pp. 1-7). IEEE. DOI: 10.1109/ICARM54641.2022.9959511

[11] Jin, X., Wang, Z., Yu, Z., Choi, C., Carpenter, K., & Nanayakkara, T. (2025). Physical Reservoir Computing in Hook-Shaped Rover Wheel Spokes for Real-Time Terrain Identification. ArXiv. https://doi.org/10.48550/arXiv.2504.13348

[12] Kabir, M. M., Jim, J. R., & Istenes, Z. (2025). Terrain detection and segmentation for autonomous vehicle navigation: A state-of-the-art systematic review. Information Fusion, 113, 102644. https://doi.org/10.1016/j.inffus.2024.102644

[13] Moore, J., Stuart, S., McMeekin, P., Walker, R., Nouredanesh, M., Tung, J., Reilly, R., & Godfrey, A. (2023). Toward enhanced free-living fall risk assessment: Data mining and deep learning for environment and terrain classification. *Intelligence-Based Medicine, 8*, 100103. https://doi.org/10.1016/j.ibmed.2023.100103