SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Synthetic Image Data Generation Via Rendering Techniques for Training AI-Based Instance Segmentation

Dickson Kho Yik Cheng[1], Norazlianie Sazali[1,*], Maurice Kettner[2], Christian Friedrich[3], Constantin Schempp[4], Naqib Salim[4], Ismayuzri Ishak[1], Saiful Anwar Che Ghani[5]

[1] Faculty of Manufacturing and Mechatronic Engineering Technology, University Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[2] Institut für Kälte-, Klima- und Umwelttechnik, Karlsruhe University of Applied Sciences, Germany
[3] Institute for Control Engineering of Machine Tools and Manufacturing Units, Seidenstraße 36, Stuttgart 70174, Germany
[4] Karlsruhe University of Applied Sciences, 76131 Karlsruhe, Germany
[5] Faculty of Mechanical and Automotive Engineering, Universiti Malaysia Pahang, Malaysia

ARTICLE INFO

Article history:
Received 30 April 2024
Received in revised form 7 October 2024
Accepted 15 October 2024
Available online 20 November 2024

Keywords:
Virtual Reality; Simulator; Signal processing

ABSTRACT

Within the scope of the development of the project, the acceleration and gyro signal, VR implementation, and User Control for VR were realized. The starting point was a list of MPU9250 data, which are acceleration and gyro data. The goal is to do the signal processing of the signal that is got from MPU9250 and implement the data with a specific format so that the LINAK LA36 actuator can be controlled in such a way that it performs as desired position. For this purpose, a multi-axis acceleration and position sensor is mounted on a surfing board which then collects the acceleration and Gyro data. A MATLAB Program had been written to convert the desired position into a CAN message. Before that, the raw acceleration data and gyro must be converted and filtered into actual data. First, the acceleration data was collected from another group. A signal processing method has been implemented. This includes gravitational force filter, noise filter, integration, and complementary filter method was used in order to get the accurate actual data that is able to be implemented into the system. For those methods, it was necessary to deal extensively with the characteristics of signal processing. VR application was implemented in this project to give the user a sensational surfing experience in the virtual world. User control has been implemented as well to allow users to control the video. Unity with Arduino work together so that both Arduino and Unity can communicate with each other, and signal messages will be sent to the Arduino from Unity and vice versa to achieve the synchronization of the video signal and Unity VR application.

1. Introduction

The rapid progress of Virtual Reality (VR) technology has transformed multiple industries, including experiential education, high-risk simulation, and entertainment [1, 2]. VR adoption in industrial settings is still restricted, despite its potential [3, 4]. Yet, the potential for immersive experiences has inspired creative initiatives focused on closing this divide [5]. This paper explores the

* *Corresponding author.*
E-mail address: azlianie@umpsa.ed.my

https://doi.org/10.37934/aram.53.1.237253

creation of a surf simulator, highlighting the potential of VR technology and tackling issues related to real-time data synchronization and user interface design.

The main goal of the project is to build a surf simulator that can provide a genuine surfing experience regardless of external factors. The key aspect of this objective is the incorporation of VR technology, which offers to transport users to virtual beaches, delivering an immersive and authentic experience of the sport [6-8]. To achieve this, it is crucial to synchronize video signals with VR glasses to guarantee a smooth viewing experience. Translating acceleration signals accurately into motor controls is crucial for replicating the dynamic movements of surfing [9, 10].

The surf simulator combines hardware and software ingenuity [11]. The simulator utilizes real-time sensor data to mimic the complexities of surfing, with a rotatable board that can be adjusted for tilt and height [12]. Nevertheless, the project faces challenges. Constraints in tilt angles and height adjustments require careful planning to maintain a seamless user experience and prevent any risks of surpassing operational limits.

The simulator's hardware design highlights the project's technical complexity, being governed by Arduino Mega Microcontrollers and employing CAN messages for signal transmission [13, 14]. The collaboration of various fields, such as engineering and computer science, drives this project forward [15, 16]. Data processing, originating from unprocessed sensor data, is crucial for the functioning of the simulator, highlighting the project's interdisciplinary character [17, 18].

Moreover, incorporating VR into the simulator enhances the user experience significantly. By utilizing game engine software and 360° video footage, users are immersed in a virtual world that vividly simulates the experience of surfing. The project combines physical simulation with virtual immersion to showcase the integration of technology and recreation, providing a preview of the future of experiential entertainment.

2. Methodology

2.1 Hardware

VR technology generates immersive experiences using sophisticated visual and auditory software. Specialized headsets with screens offer clear views, allowing users to navigate virtual environments. Proximity, gyroscope, and accelerometer sensors improve interactions. The VR package includes a headset, 2 controllers, and Oculus software. Setting up the Oculus Quest involves establishing boundaries to prevent collisions. Each controller features a thumb stick, three buttons, and an index trigger.

2.2 Software

2.2.1 Oculus

The Oculus VR desktop is software designed for gaming and created by Facebook Technologies. It is a supporting application for the Oculus virtual reality headsets that not only manages the hardware but also enables users to download applications from the Oculus Store and explore live VR events. Developing a VR application in 3D Unity requires Oculus packages and an Oculus PC desktop app [19].

2.2.2 Unity

Unity is an effective platform used for creating 3D and 2D applications on various platforms. It is known for its flexibility and intuitive interface, which resembles CAD software [20]. The software

provides an intricate editor for manipulating virtual objects, essential for developing dynamic and interactive scenes, particularly for immersive VR experiences. Unity offers a strong scripting environment for developers to define object behaviors, manage interactions, and control game logic effectively, with a focus on the C# programming language [21]. Unity Hub serves as a centralized management tool, streamlining project organization and facilitating version control to ensure compatibility across different project iterations. Maintaining compatibility with specific Unity versions is crucial to preserve project integrity, necessitating meticulous management and version control within Unity Hub. Updating projects to newer Unity versions may require manual modifications to guarantee compatibility with current code and assets. This careful approach ensures smooth transitions and optimal performance across different Unity environments.

2.2.3 Visual studio

Visual Studio is an integrated development environment (IDE) utilized for building a variety of computer programs ranging from web applications, web services and mobile apps [22]. This project is utilized for writing scripts for Unity. Visual Studio IDE has a built-in variety of programming languages that includes C, C++, C++/CLI, Visual Basic.NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. C# Programming language is used when writing scripts with Unity for a Game object in the Scene.

2.2.4 Arduino IDE

The Arduino organization developed the Arduino Integrated Development Environment (IDE) for programming Arduino-based microcontrollers. A complete platform for writing, compiling, and uploading Arduino code. Access to a wide range of software libraries and simplification of the C/C++ programming language make the IDE useful for development. Users can synchronize and transfer data between their microcontroller boards and computers via a serial port, usually a USB port, using the IDE [23]. For instance, an Arduino Mega board with an SD card module can read and store SD card data. Users can also monitor and interact with their projects using real-time data visualization in the IDE. Arduino IDE programming revolves around Setup() and Loop(). Program initialization occurs at system startup with Setup(). In contrast, Loop() executes the main program logic and calls other functions in an infinite loop. The microcontroller board executes these C++ programs compiled by the IDE.

2.2.5 MATLAB

MATLAB is a programming platform exclusively for engineers and scientists to analyze and design world-changing systems and products. The matrix-based MATLAB language provides the most intuitive way to express computational mathematics [24]. When researching signal processing, MATLAB is invaluable. Data graphs, data changes, and signal controller implementation allow us to see data. It helps us see how outcomes change.

A module's user control component manages user integrations and actions in a software application. User controllers facilitate communication and coordination between the user interface (UI) and application logic. User authentication, session management, and access control are usually in the user controller. Displaying appropriate messages is part of user interface management. Overall, the user controller is crucial to a smooth and efficient user experience. The hub interacts with the

application to work properly. This project creates a user controller for 360° video and simulator signal.

2.3 Method

The project concept revolves around enhancing a simulator system by integrating a virtual reality (VR) environment, particularly focused on creating an immersive surfing experience. To achieve this, the system's components, including the robot simulator system and the CAN bus signal structure, are thoroughly analyzed. Figure 1 provides a visual representation of the initial steps involved in starting 3D printing, which is integral to the development process.

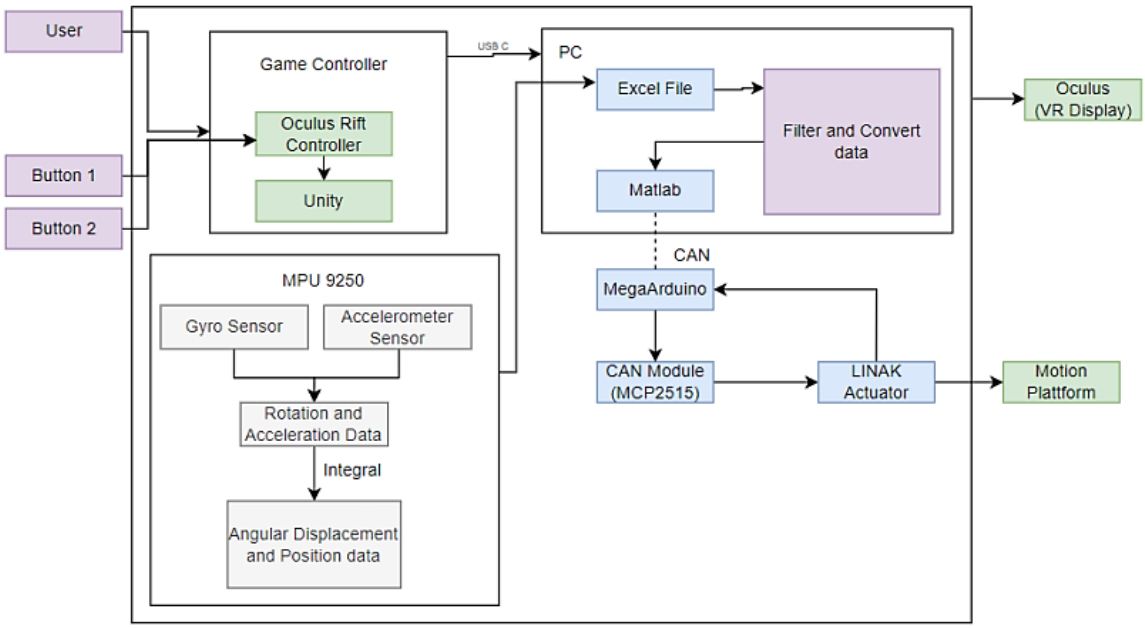


Fig. 1. Flowchart of starting 3D printing

2.3.1 Output situation

The output situation analysis assesses the existing surfing simulator, emphasizing its main features like Arduino-controlled actuators and MPU9250 sensors. The simulator's design is limited by its use of a tripod concept with three Linak LA36 actuators and one rotary motor to provide four degrees of freedom. The Arduino Mega 2560 controls actuators through an MCP2515 CAN module, requiring a CAN Message array in the Arduino program. Analysis is conducted on past software and programming, such as MATLAB functions used for data conversion. Emphasis is placed on integrating virtual reality (VR) to enhance the surfing experience, using Unity and Oculus Rift for development. An intuitive control interface is created for concurrent video and simulator operation.

2.3.2 Strategy selection

It is essential to comprehend the complexities of raw data when choosing methods to convert signals into control signals. Acceleration signals in their raw form consist of components related to movement, gravity, and noise, requiring methods to differentiate between them. Utilizing a rotational matrix can separate the gravity component [25]. Filtering techniques are crucial for processing raw data, and the Low-Pass filter is chosen in this project for its ability to eliminate noise.

This filter effectively ensures signal accuracy and reduces noise. Numerical integration methods are used to process data, with the trapezoidal rule selected for calculating displacement data from acceleration data due to its accuracy compared to Riemann Sums. This ensures robust integration and reliable results for further analysis.

Resolving drift over time in gyro data is essential to keeping angle computations accurate. To address this issue, a complementary filter that utilizes both accelerometer and gyro data is used. The filter uses trigonometric principles to precisely determine sensor position angles, ensuring more dependable data for analysis [26]. Figure 2 illustrates the data conversion topology, showing the sequential process of converting raw data into useful control signals. This visualization clarifies the intricacy of data transformation, highlighting the significance of each step in guaranteeing the accuracy and dependability of the final output.

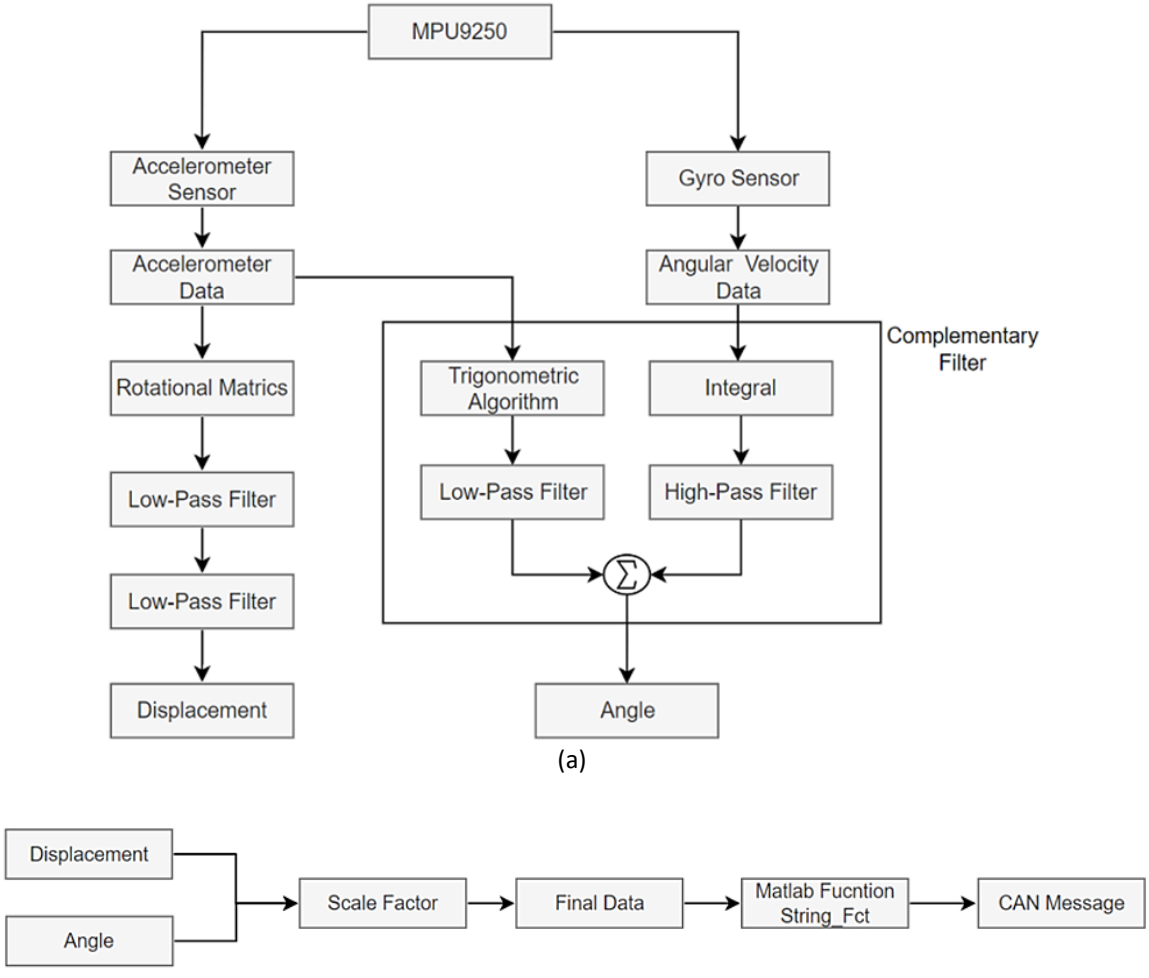


Fig. 2. (a) The translation of signal data (b) Conversion of signal data to final data

The last stage of signal conversion includes implementing a scaling method on the signal data to adjust it to a specific range and guarantee it aligns with the simulator's limitations. This modification is crucial to avoid problems like data overflow or underflow, ultimately improving the precision and authenticity of the simulation. The scaling method maintains smooth functionality by keeping the signal data within the operational limits of the simulator.

Furthermore, incorporating virtual reality (VR) is crucial for developing an immersive user experience. Ardity enables smooth communication between Unity and Arduino by facilitating their

connection [27]. Unity sends messages to Arduino through USB to activate movements in response to the messages received. Figure 3 illustrates the simultaneous integration of 360° video and signal processing. The visualization demonstrates the smooth incorporation of VR technology with processed signal data to create a cohesive and immersive simulation that faithfully reproduces real-world surfing scenarios.

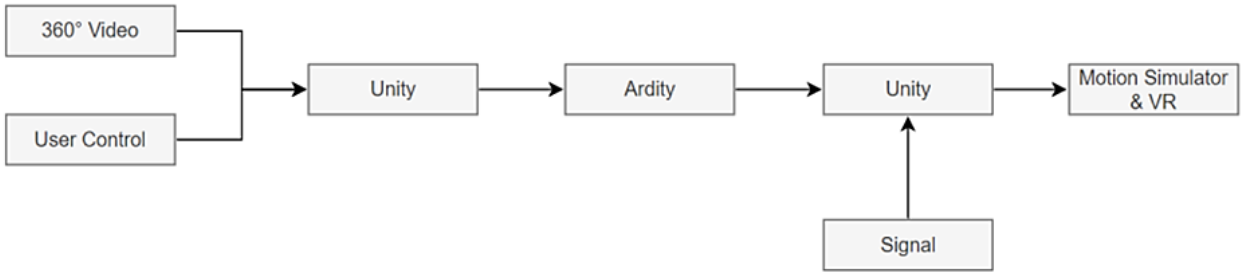


Fig. 3. Topology of 360 Video and Signal Implementation simultaneously

3. Results

3.1 360º Video Implementation

A 360° video recorded during surfing often contains unnecessary footage, such as board turning and swimming sequences, which need to be trimmed. A brief 12-second video clip is created after editing, displaying the precise data captured by the MPU9250 sensor. Three essential components needed to play a 360° video in Unity are a Video Player component, a Render Texture, and a Skybox Material. The Video Player showcases the video using a specialized texture called a Render Texture. The Render Texture is used as a texture for the Skybox Material, enabling the video to be displayed as a panorama in the scene.

Start by creating a new Unity project and importing necessary plugins like the Oculus Integration plugin, which can be downloaded from the Unity Asset Store or the Oculus developer website. The 360° video file is subsequently brought into the project. Subsequently, a Video Player component is included in the Unity Hierarchy window. A Skybox Material is generated and a panoramic shader is assigned within the material's inspector settings. A Render Texture is created in the Assets panel with a resolution of 4096x2048 pixels to match the video's resolution. Assign the Render Texture to the Video Player component's Target Texture property.

The 360° video is then placed into the video clip slot of the Video Player component by dragging it. Multiple settings are provided for the video, such as the ability to activate looping and play automatically when the scene starts. The 'Play on Awake' feature is turned off in this setup to initiate video playback manually, usually by clicking a specific 'PLAY' button. This process explains how to create a 360° video in Unity to develop immersive virtual reality experiences.

3.2 Programming in MATLAB

In order to generate a list of filtered data, you need to create a MATLAB script. The project has been executed using MATLAB. Figure 4 below illustrates the topology of MATLAB. The green color symbolizes the input, blue represents the MATLAB function, and orange signifies the output.

The MATLAB script requires time, raw acceleration data, and raw gyroscope data for X, Y, and Z axes to calculate acceleration and gyroscope data. The number of lines corresponds to the recorded data quantity, which does not affect the MATLAB function directly. Nevertheless, it is important to

acknowledge that this alters the duration of the program's simulation. The raw data will be processed using complementary filter sensor fusion and rotational matrices.

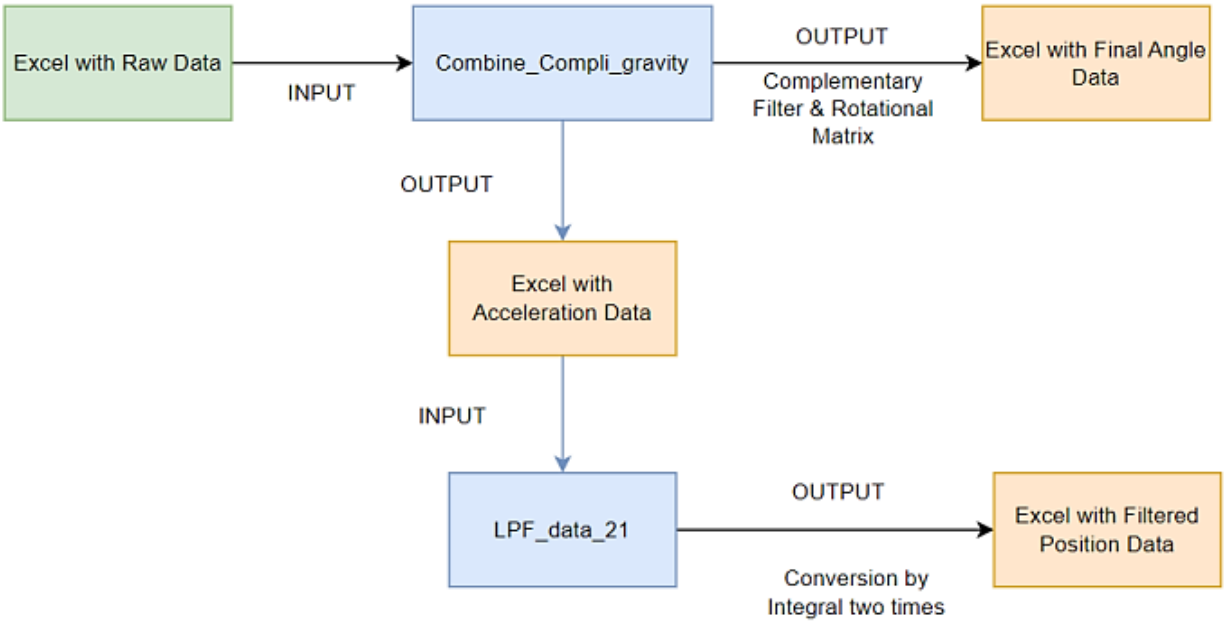


Fig. 4. Topology of MATLAB function

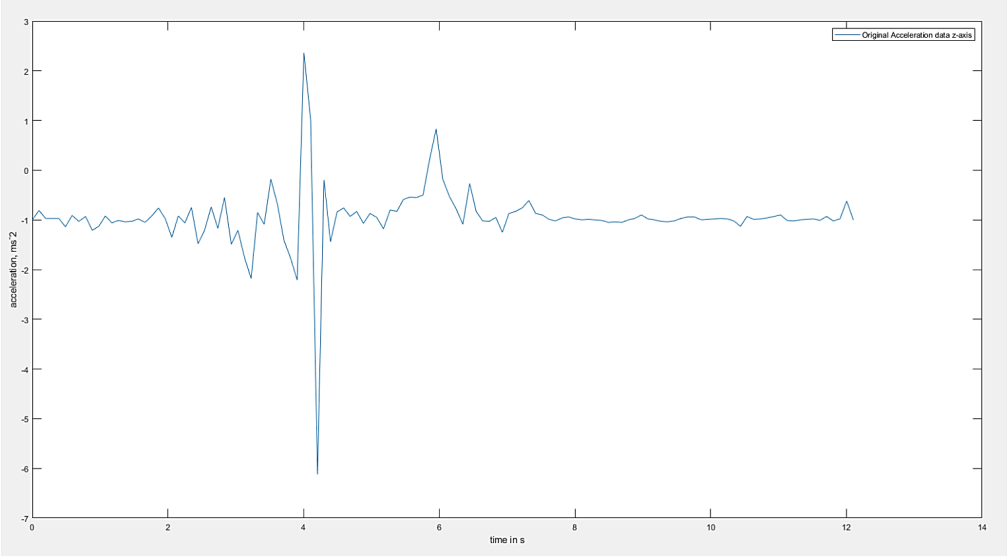
This MATLAB function implements two strategies: the complementary filter and rotational matrix to calculate the actual angle and linear acceleration, respectively. The calculated angle is subsequently utilized as an input for determining the linear acceleration data. Two outputs will be generated in an excel file: a list of Z-axis Linear Acceleration data and angles in Raw, Pitch, and Yaw. In this case, the X and Y Linear Acceleration will not be utilized as the Z-Axis accelerometer, which indicates the height or MaPa of the simulator, is crucial. The Z-Acceleration output is not yet suitable for determining travel positions. It must be filtered and transformed into Z-axis displacement data. The current angles of Roll, Pitch, and Yaw are prepared for the next stage simultaneously.

The Matlab script requires the time and raw acceleration data for the X, Y, and Z axes as input parameters for the acceleration conversion. The linear acceleration calculated by the Combine_Compli_Gravity Matlab Function must be replicated and replaced in a new Excel file before being integrated into LPF_data_21. LPF_data_21 is a MATLAB function that performs filtering and integration. A low pass filter will be applied to eliminate noise, followed by the integration of the filtered acceleration data using the Cumtrapz MATLAB Function. The final position data is calculated using double integration. Prior to integration, a filter strategy must be established to prevent noise from amplifying and leading to the production of inaccurate data.

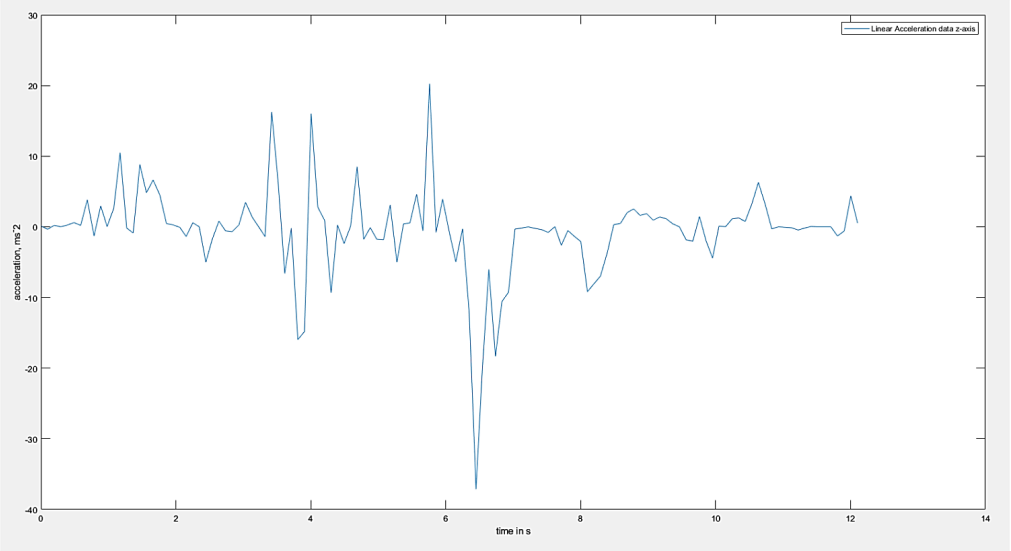
3.3 Rotational Matrix in MATLAB

Figure 5 below shows the raw acceleration signal and the linear acceleration. The data needs to be filtered to eliminate noise, a subject that will be addressed in the following sub-chapter. The graph generated from the raw data shows that the signal does not originate from zero. The discrepancy occurred because the MPU9250 was initially positioned on a nearly level surface on the surfing board, resulting in it tilting at a specific angle. As a result, gravity exerted forces on the accelerometer shortly after, causing the signal offset that was observed. A rotational matrix is used to calculate the acceleration caused by gravity, which is then removed from the original signal to solve the problem.

This correction process ensures that the signal begins at the initial position of 0, improving its accuracy for subsequent analysis and processing [28].



(a)



(b)

Fig. 5. (a) Acceleration data respect to time (b) Linear acceleration data respect to time

3.5 Filtering Accelerometer Data

3.5.1 Noise and error

Utilizing sensor data and formulas should ideally result in accurate angles, however, multiple factors can impact sensor output in practice. Accelerometer detects movement accelerations when in motion, and additional acceleration values can affect orientation accuracy. Noise frequently disrupts the electrical signal, leading to unwanted disturbances. An accelerometer can measure various angles, but its readings may still contain noise and have a margin of error, despite using a low-pass filter. Gyroscopes can develop bias instabilities leading to drift over time because of integrating inherent imperfections and noise from the device's initial zero reading. To address gyro

drift in applications requiring precise orientation using sensor fusion along with acceleration data [29].

3.5.2 Low past filter

A low-pass filter allows signals with frequencies below a set cut-off frequency to pass through but reduces signals with frequencies above the cut-off [30]. The filter's exact frequency response is determined by its design. In audio applications, it may also be known as a high-cut or treble-cut filter. A low-pass filter is the opposite of a high-pass filter. Accelerometer data was gathered for this project, and it exhibits high levels of noise. Implementing the raw accelerometer data in the simulator will lead to a hazardous outcome. Hence, it should be smoothed by applying a filter to eliminate unwanted noises. Figure 6 below represents one of the data examples. The solid line in blue and yellow represents the unfiltered and filtered data using a low-pass filter strategy. The code is written in MATLAB in this instance.

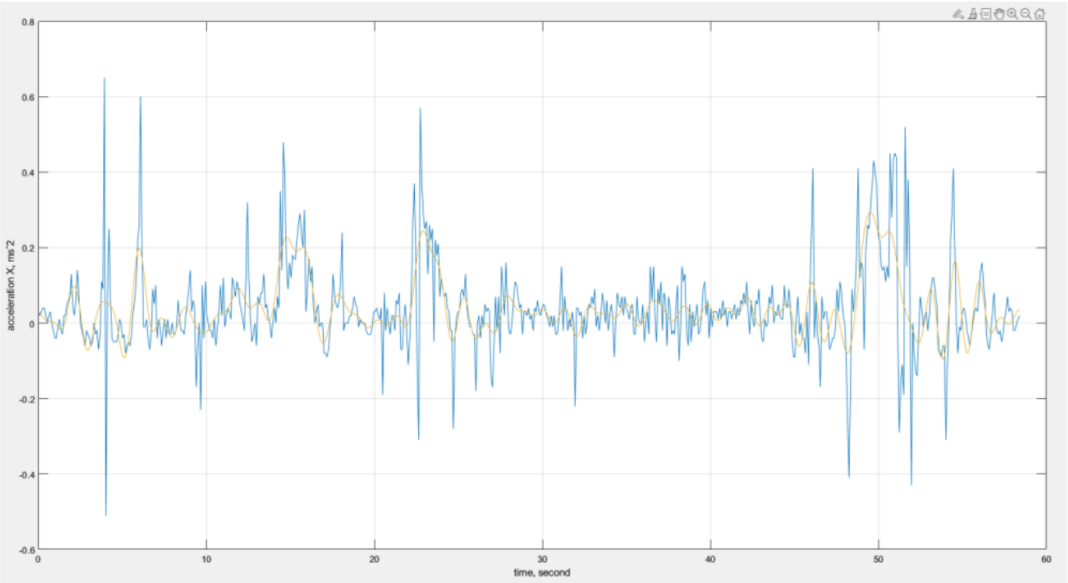
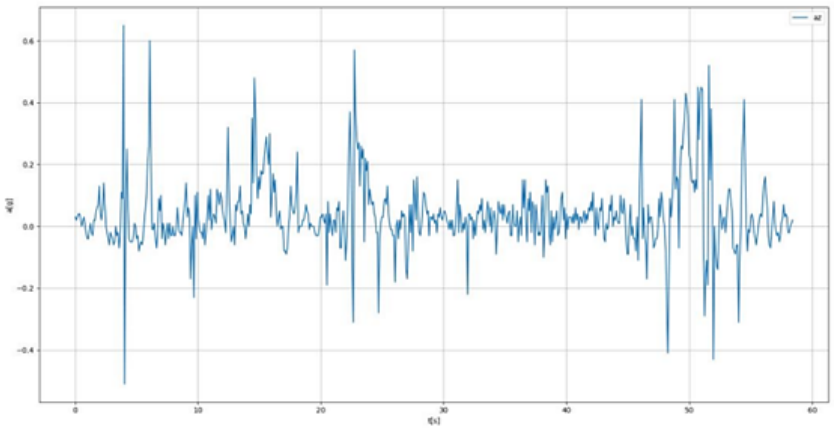


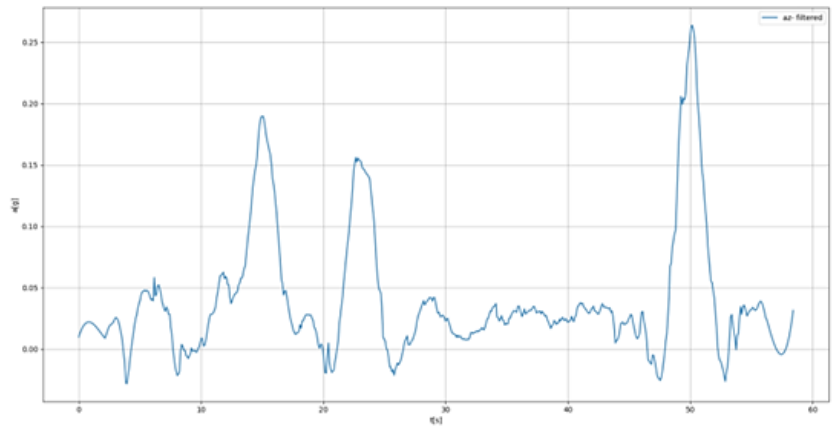
Fig. 6. The graph plotted from low pass filter

3.5.3 Savitzky-Golay filter

A Savitzky-Golay filter is a digital filter that can smooth a set of digital data points. The purpose is to improve data accuracy by minimizing noise while preserving the signal's overall trend [31]. Savitzky-Golay smoothing filters are used to remove noise from a signal with a broad frequency range while preserving the signal. They are also known as digital smoothing polynomial filters or least-squares smoothing filters. Savitzky-Golay filters can outperform traditional averaging FIR filters in specific scenarios by effectively eliminating noise while preserving high-frequency content. Figure 7(a) displays raw data, while Figure 7(b) shows the application of the Savitzky-Golay Filter in Python using Visual Studio Code. The data is measured in grams, as the MPU9250 provides raw output in grams, representing acceleration due to gravity at 9.81m/s^2. The x-axis represents time in this example, with the time recorded at approximately 60 seconds.



(a)



(b)

Fig. 7. The raw acceleration data (a) Without filter (b) With Savitzky -Golay filter

In the previous chapter, a low-pass filter was selected as the method to remove noise from the accelerometer data. The accelerometer data being referred to is different from the data used for the previously mentioned Savitzky-Golay Filter. Figures 6 and 7 compare the Low-Pass Filter with the Savitzky-Golay Filter. Both raw and processed data are displayed in MATLAB. The yellow line in the graphs indicates the filtered data, whereas the blue line represents the unfiltered data. Figure 8 shows the filtered data, which is then used to convert into position data through integration.

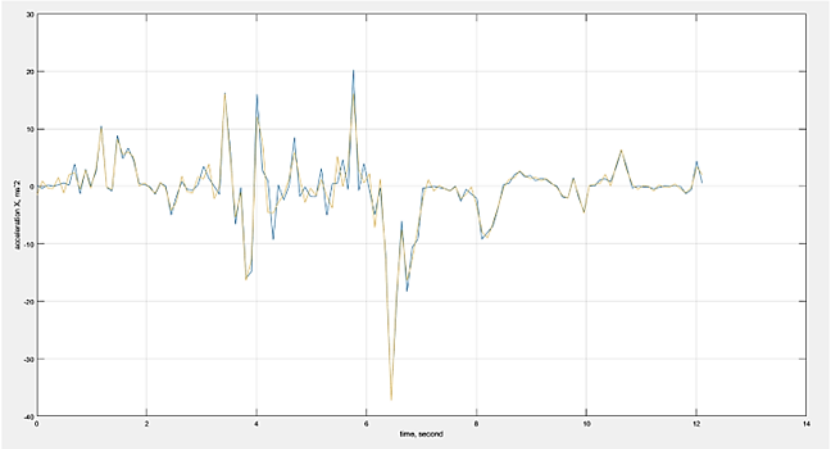
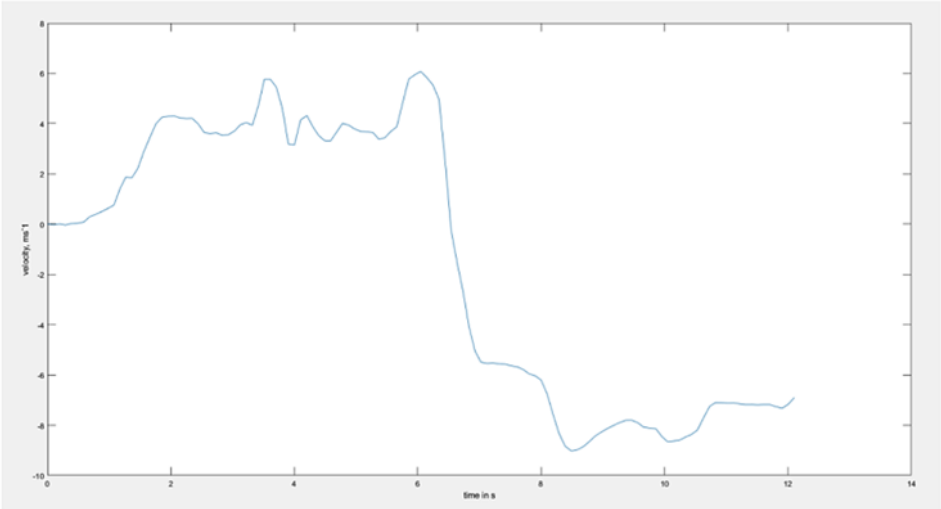


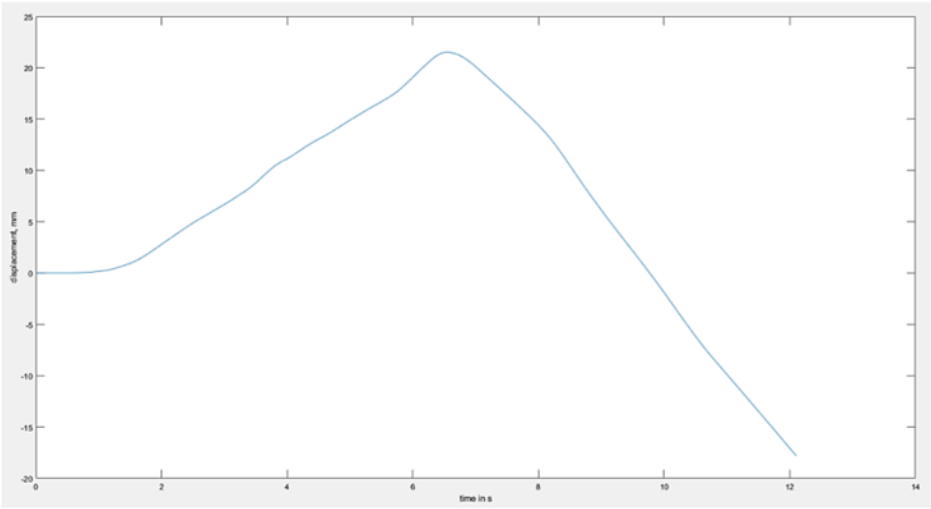
Fig. 8. Plot of the filtered and unfiltered data

3.6 Integration

The trapezoidal rule calculates the area under a curve by dividing the total area into small trapezoids instead of rectangles, resulting in a more accurate estimation compared to the Riemann sum [32]. This project utilizes Trapezoidal numerical integration to process a string of numbers and acceleration data. Figure 9 displays the velocity and displacement graphs, which are the result of integrating acceleration twice. A 12-second segment of a 360° video is viewed through VR glasses. This video depicts a moment when a wave almost reaches the surfing board. After six seconds, as the tide is almost finished, a sudden decrease in displacement coincides with the pre-recorded video. A web server that converts accelerometer data to position data is compared with Cumtrapz in MATLAB. The graph generated by the web server in Figure 9(b) lacks a filter but exhibits the same pattern as Figure 10, confirming the validity of the approach.



(a)



(b)

Fig. 9. (a) The conversion of acceleration to velocity (b) The conversion of velocity to position

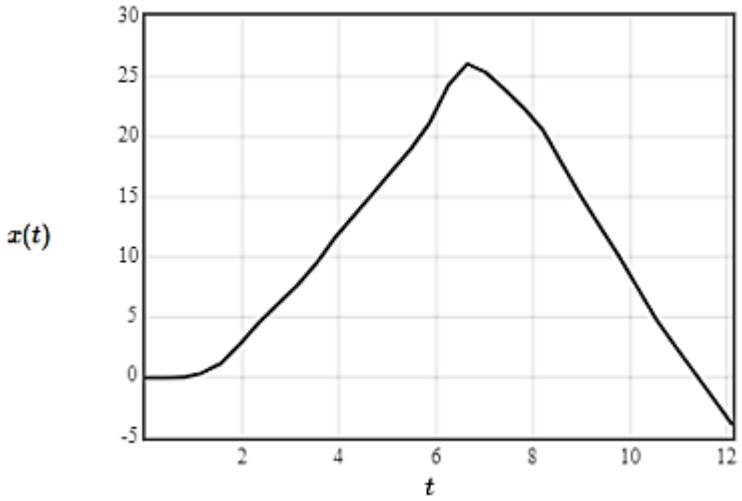
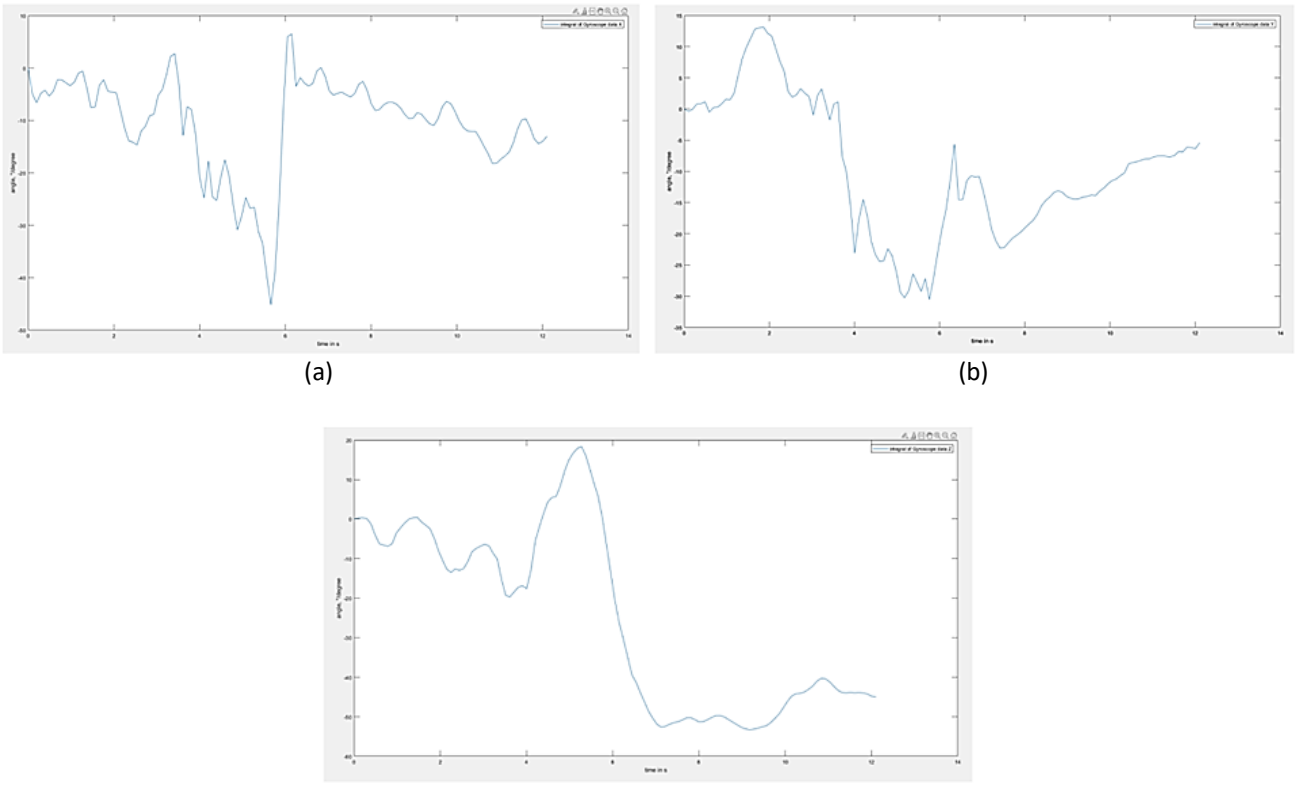


Fig. 10. Position graph generated from the web server

3.7 Integration on Gyroscope Data

A gyroscope can measure angular velocity, which can be used to calculate the angle by integrating the angular velocity over time [33]. However, this method leads to an output that could gradually deviate over time because of the vibrating components in the gyroscope. The final calculated data is plotted in three graphs with X, Y, and Z axes displayed below. The data is influenced by vibration and lacks consistency over time. Figure 11 displays the final output of the angle using a complementary filter. The graphs are almost identical to those shown in Figure 11. Yet, the data varies slightly, and using a complementary filter result in a more precise output.



(a) (b)

(c)

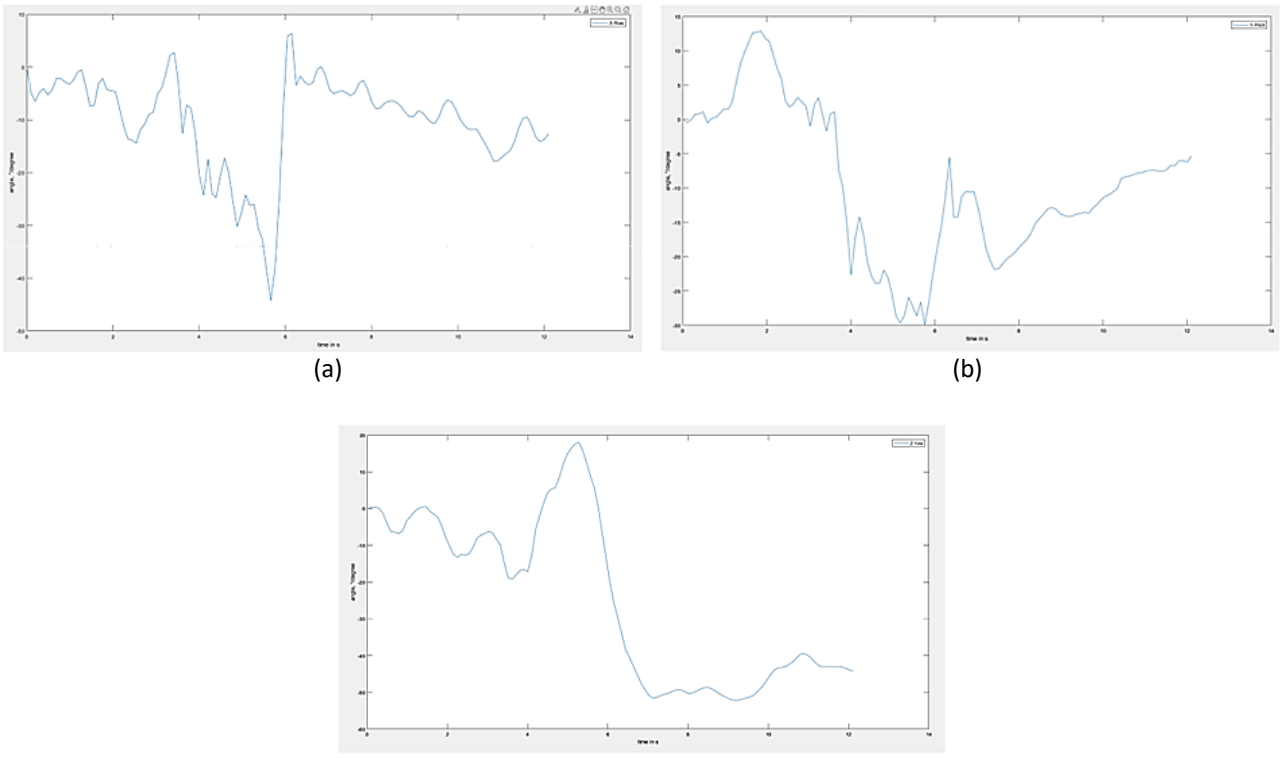
Fig. 11. The final computed angle (a) X-Row, (b) Y-Pitch (c) Z-Yaw after gyroscope data filter

3.8 Complementary Filter

A complementary filter is a signal-processing algorithm that improves precision by combining data from various sensors [34]. A complementary filter calculates the absolute angle of an object using accelerometers and gyroscopes. Accelerometers measure acceleration to determine an object's angle in relation to gravity, but their readings may be imprecise and susceptible to gradual deviation. Gyroscopes measure angular velocity, which, when integrated, gives the object's angle. Gyro measurements are prone to noise and drift, particularly during integration.

A complementary filter combines accelerometer and gyroscope measurements to improve the accuracy of angle estimation and overcome obstacles [35]. The device functions by implementing a high-pass filter on gyro measurements and a low-pass filter on accelerometer data, which are then merged using weighted averages. The high-pass filter removes low-frequency noise and drift from gyro data while maintaining high-frequency angular variations. The low-pass filter eliminates high-frequency noise and vibrations from accelerometer data, preserving gradual changes in acceleration that indicate changes in object angle.

The complementary filter output combines filtered gyro and accelerometer data, with their importance determined by a weighting parameter. This method provides a more precise absolute angle estimation than using each sensor separately. Figure 12 shows the ultimate angle output using the complementary filter, displaying similar results to Figure 11 but with slight variations that highlight the superior accuracy of the complementary filter.



(a) (b)

(c)

Fig. 12. The final computed angle (a) X-Row (b) Y-Pitch (c) Z-Yaw after complementary filter

3.9 Scaling Method

A scaling strategy would involve adjusting its parameters or components to meet the desired performance or functionality. Table 1 presents the filtered and converted data. Due to the limitations

of the surfing simulator, the minimum and maximum angles that can be performed are -13° and 13°, while the minimum and maximum height will be 550mm to 590mm (Table 2). The range of the data received from MPU9250 mostly falls into the range of limitation. However, certain data will fall outside the limits. Therefore, scaling may involve implementing a strategy to ensure that the simulator stays within limits (Table 3).

$$m = \frac{m - rmin}{rmax - rmin}(tmax - tmin) + tmin \quad (1)$$

Table 1
The filtered and converted data

Final Theta X	Final Theta Y	Final Theta Z	Filtered Position
0	0	0	0
-4.95629389	-0.40699389	0.25745611	-0.001384605
-6.617804833	-0.030319166	0.314786175	-0.002741366
-4.869653666	-0.775112844	0.160659419	-0.004389779
-4.226160992	0.83483737	-1.218690261	-0.005129515
-5.350103698	1.14391409	-4.027366869	-0.002395532
-4.41352903	-0.490527638	-6.280727701	-0.003276803
-2.214296162	0.217605165	-6.596559176	-0.021836843
-2.27392691	0.309818453	-6.897982153	-0.055787061
-2.8378593	0.846446294	-6.208154459	-0.100003062
-3.371755547	1.573626278	-3.548109119	-0.153548303

Table 2
The minimum and maximum value the data

	X	Y	Z	MaPa
Maximum	6.486275	13.15961	18.3914	21.53752
Minimum	-45.18	-30.5244	-53.2845	-17.802

Table 3
The scaled value

Alpha	Beta	Gamma	MpMa
9.735913	5.167618	6.328634	568.1009
7.241757	4.925382	6.422024	568.0995
6.405635	5.149573	6.44282	568.0981
7.285357	5.628953	6.386912	568.0964
7.609182	5.6645	5.886561	568.0957
7.04358	5.848457	4.867731	568.0985
7.514893	4.875664	4.050338	568.1042
8.621613	5.297133	3.935772	568.1231
8.591605	5.352017	3.826433	568.1576
8.307817	5.671409	4.076663	568.2026
8.039144	6.104215	5.041579	568.257

4. Conclusions

A 4 DOF surfing simulator with a rotatable table able to tilt at a certain angle and height with the help of the LINAK LA36 Actuator. A MATLAB program was written by the previous group, which functions as converting a list of data with a desired format into a CAN message array which is then programmed into Arduino IDE to control all the LINAK Actuators. A measuring box with MPU9250 was mounted onto the surfing board, and a list of acceleration and gyro data was recorded. Signal

processing strategies were implemented in the raw data signal so that it is in the right format to be converted into MATLAB function before producing a CAN Message array.

A low-pass filter was implemented into acceleration data to remove the noise. Then displacement data is achieved from accelerometer data by integral twice. In this case, the z-position is the only data needed for the simulator as it symbolizes the MpMa, which is the height of the simulator. A sensor fusion is implemented to obtain accurate angular data, with the complementary filter, accelerometer data, and gyroscope data both used to calculate the angle. They were then filtered by low-pass and high-pass filters before combining by weigh factor.

Virtual Reality was implemented in this project. A recorded 360° Video was programmed and put into OCULUS Rift so that users could have a real surfing experience in the virtual world. A strategy that synchronizes the video and signal of the simulator was designed by building a communication tunnel between Arduino IDE and Unity so that both of them can communicate by sending and receiving signals or messages between them. In Unity, a user control was designed to control the signal and video simultaneously. For safety reasons and future development, a user control was developed in Unity as well so that a second user could control both videos and signals on the laptop while the first user was standing on top of the surfing simulator.

Acknowledgement

This research was funded by a grant from Universiti Malaysia Pahang for funding under grant RDU232707 and UIC231511.

References

[1] Hamad, Ayah, and Bochen Jia. "How virtual reality technology has changed our lives: an overview of the current and potential applications and limitations." *International journal of environmental research and public health* 19, no. 18 (2022): 11278. https://doi.org/10.3390/ijerph191811278

[2] Hashim, Mohd Ekram Alhafis, Nur Safinas Albakry, Wan Azani Mustafa, Banung Grahita, Miharaini Md Ghani, Hafizul Fahri Hanafi, Suraya Md Nasir, and Catherina ana Ugap. "Understanding the impact of animation technology in virtual reality: A systematic literature review." *International Journal of Computational Thinking and Data Science* 1, no. 1 (2024): 53-65. https://doi.org/10.37934/CTDS.1.1.5365

[3] Coban, Murat, Yusuf Islam Bolat, and Idris Goksu. "The potential of immersive virtual reality to enhance learning: A meta-analysis." *Educational Research Review* 36 (2022): 100452. https://doi.org/10.1016/j.edurev.2022.100452

[4] Tom Dieck, M. Claudia, and Dai-in Danny Han. "The role of immersive technology in customer experience management." *Journal of marketing theory and practice* 30, no. 1 (2022): 108-119. https://doi.org/10.1080/10696679.2021.1891939

[5] Al-Ansi, Abdullah M., Mohammed Jaboob, Askar Garad, and Ahmed Al-Ansi. "Analyzing augmented reality (AR) and virtual reality (VR) recent development in education." *Social Sciences & Humanities Open* 8, no. 1 (2023): 100532. https://doi.org/10.1016/j.ssaho.2023.100532

[6] Dwivedi, Yogesh K., Laurie Hughes, Abdullah M. Baabdullah, Samuel Ribeiro-Navarrete, Mihalis Giannakis, Mutaz M. Al-Debei, Denis Dennehy, Bhimaraya Metri, Dimitrios Buhalis, Christy M.K. Cheung, Kieran Conboy, Ronan Doyle, Rameshwar Dubey, Vincent Dutot, Reto Felix, D.P. Goyal , Anders Gustafsson, Chris Hinsch, Ikram Jebabli, Marijn Janssen, and Samuel Fosso Wamba. "Metaverse beyond the hype: Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy." *International journal of information management* 66 (2022): 102542. https://doi.org/10.1016/j.ijinfomgt.2022.102542

[7] Checa, David, and Andres Bustillo. "A review of immersive virtual reality serious games to enhance learning and training." *Multimedia Tools and Applications* 79, no. 9 (2020): 5501-5527. https://doi.org/10.1007/s11042-019-08348-9

[8] Flavián, Carlos, Sergio Ibáñez-Sánchez, and Carlos Orús. "The influence of scent on virtual reality experiences: The role of aroma-content congruence." *Journal of Business Research* 123 (2021): 289-301. https://doi.org/10.1016/j.jbusres.2020.09.036

[9] Sadaf, Memoona, Zafar Iqbal, Abdul Rehman Javed, Irum Saba, Moez Krichen, Sajid Majeed, and Arooj Raza. "Connected and automated vehicles: Infrastructure, applications, security, critical challenges, and future aspects." *Technologies* 11, no. 5 (2023): 117. https://doi.org/10.3390/technologies11050117

[10] Fan, Yanan, Zhongcai Pei, Chen Wang, Meng Li, Zhiyong Tang, and Qinghua Liu. "A Review of Quadruped Robots: Structure, Control, and Autonomous Motion." *Advanced Intelligent Systems* (2024): 2300783. https://doi.org/10.1002/aisy.202300783

[11] Wang, Chin-Chih, Chao-En Yen, Chuen-Liang Chen, Jia-Xiang Wu, and Pangfeng Liu. "Ocean wave simulation in real-time using GPU." In *2010 International Computer Symposium (ICS2010)*, pp. 419-423. IEEE, 2010. https://doi.org/10.1109/COMPSYM.2010.5685474
[12] Roberts, Michael, and Jess Ponting. "Waves of simulation: Arguing authenticity in an era of surfing the hyperreal." *International Review for the Sociology of Sport* 55, no. 2 (2020): 229-245. https://doi.org/10.1177/1012690218791997
[13] Zaro, Fouad, Ali Tamimi, and Anas Barakat. "Smart home automation system." *International Journal of Engineering and Innovative Research* 3, no. 1 (2021): 66-88. https://doi.org/10.47933/ijeir.781091
[14] Hanafi, Hafizul Fahri, Muhamad Hariz Adnan, Miftachul Huda, Wan Azani Mustafa, Miharaini Md Ghani, Mohd Ekram Alhafis Hashim, and Ahmed Alkhayyat. "IoT-enabled GPS tracking system for monitoring the safety concerns of school students." *Journal of Advanced Research in Computing and Applications* 35, no. 1 (2024): 1-9. https://doi.org/10.37934/arca.35.1.19
[15] Ong, Siew Har, Sai Xin Ni, and Ho Li Vern. "Dimensions Affecting Consumer Acceptance towards Artificial Intelligence (AI) Service in the Food and Beverage Industry in Klang Valley." *Semarak International Journal of Machine Learning* 1, no. 1 (2024): 20-30. https://doi.org/10.37934/sijml.1.1.2030
[16] Wu, Meng, Fengchun Sun, Jinrui Nan, and Dafang Wang. "Research on reflection of CAN signal in transmission line." In *2007 IEEE International Conference on Vehicular Electronics and Safety*, pp. 1-4. IEEE, 2007. https://doi.org/10.1109/ICVES.2007.4456406
[17] Ahmed, Zeeshan, Khalid Mohamed, Saman Zeeshan, and XinQi Dong. "Artificial intelligence with multi-functional machine learning platform development for better healthcare and precision medicine." *Database* 2020 (2020): baaa010. https://doi.org/10.1093/database/baaa010
[18] Bibri, Simon Elias. "On the sustainability of smart and smarter cities in the era of big data: An interdisciplinary and transdisciplinary literature review." *Journal of Big Data* 6, no. 1 (2019): 25. https://doi.org/10.1186/s40537-019-0182-7
[19] Egliston, Ben, and Marcus Carter. "Oculus imaginaries: The promises and perils of Facebook's virtual reality." *New Media & Society* 24, no. 1 (2022): 70-89. https://doi.org/10.1177/1461444820960411
[20] Huang, Lehui, and Bin Gui. "Research on the Application of Products based on Unity3D." In *2015 International Symposium on Computers & Informatics*, pp. 1213-1217. Atlantis Press, 2015. https://doi.org/10.2991/isci-15.2015.160
[21] Wheeler, Gavin, Shujie Deng, Nicolas Toussaint, Kuberan Pushparajah, Julia A. Schnabel, John M. Simpson, and Alberto Gomez. "Virtual interaction and visualisation of 3D medical imaging data with VTK and Unity." *Healthcare technology letters* 5, no. 5 (2018): 148-153. https://doi.org/10.1049/htl.2018.5064
[22] Amann, Sven, Sebastian Proksch, Sarah Nadi, and Mira Mezini. "A study of visual studio usage in practice." In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, vol. 1, pp. 124-134. IEEE, 2016. https://doi.org/10.1109/SANER.2016.39
[23] Louis, Leo. "working principle of Arduino and u sing it." *International Journal of Control, Automation, Communication and Systems (IJCACS)* 1, no. 2 (2016): 21-29. https://doi.org/10.5121/ijcacs.2016.1203
[24] Dunning, Iain, Joey Huchette, and Miles Lubin. "JuMP: A modeling language for mathematical optimization." *SIAM review* 59, no. 2 (2017): 295-320. https://doi.org/10.1137/15M1020575
[25] Steinacker, Harold. "Emergent geometry and gravity from matrix models: an introduction." *Classical and Quantum Gravity* 27, no. 13 (2010): 133001. https://doi.org/10.1088/0264-9381/27/13/133001
[26] Walcher, Hans. *Position sensing: Angle and distance measurement for engineers*. Elsevier, 2014.
[27] Moshayedi, Ata Jahangir, Amin Kolahdooz, and Liefa Liao. *Unity in Embedded System Design and Robotics: A Step-by-step Guide*. Chapman and Hall/CRC, 2022. https://doi.org/10.1201/9781003268581
[28] Geist, Jon, Muhammad Yaqub Afridi, Craig D. McGray, and Michael Gaitan. "Gravity-based characterization of three-axis accelerometers in terms of intrinsic accelerometer parameters." *Journal of Research of the National Institute of Standards and Technology* 122 (2017): 1. https://doi.org/10.6028%2Fjres.122.032
[29] Majumder, Sumit, and M. Jamal Deen. "A robust orientation filter for wearable sensing applications." *IEEE Sensors Journal* 20, no. 23 (2020): 14228-14236. https://doi.org/10.1109/JSEN.2020.3009388
[30] Pérez-Bailón, Jorge, Alejandro Márquez, Belén Calvo, and Nicolás Medrano. "A 0.18 µm CMOS widely tunable low pass filter with sub-hz cutoff frequencies." In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-4. IEEE, 2018. https://doi.org/10.1109/ISCAS.2018.8351166
[31] Dombi, Jozsef, and Adrienn Dineva. "Adaptive Savitzky-Golay filtering and its applications." *International Journal of Advanced Intelligence Paradigms* 16, no. 2 (2020): 145-156. https://doi.org/10.1504/IJAIP.2020.107011
[32] Kareem, Hasanain J., Mohammed A. Abdulwahid, and Hasril Hasini. "Experimental investigation of holdup fraction using the trapezoidal rule, Simpson's rule and the average offset formula in perforated horizontal wellbore." *Results in Engineering* 18 (2023): 101131. https://doi.org/10.1016/j.rineng.2023.101131

[33] Passaro, Vittorio MN, Antonello Cuccovillo, Lorenzo Vaiani, Martino De Carlo, and Carlo Edoardo Campanella. "Gyroscope technology and applications: A review in the industrial perspective." *Sensors* 17, no. 10 (2017): 2284. https://doi.org/10.3390/s17102284

[34] Liu, Mei, Yuanli Cai, Lihao Zhang, and Yiqun Wang. "Attitude estimation algorithm of portable mobile robot based on complementary filter." *Micromachines* 12, no. 11 (2021): 1373. https://doi.org/10.3390/mi12111373

[35] Madgwick, Sebastian OH, Samuel Wilson, Ruth Turk, Jane Burridge, Christos Kapatos, and Ravi Vaidyanathan. "An extended complementary filter for full-body MARG orientation estimation." *IEEE/ASME Transactions on mechatronics* 25, no. 4 (2020): 2054-2064. https://doi.org/10.1109/TMECH.2020.2992296