MEKATRONIKA
e-ISSN: 2637-0883
VOL. 5, ISSUE 2, 41 – 49
DOI: https://doi.org/10.15282/mekatronika.v5i2.9824

ORIGINAL ARTICLE

Exploring Micro Aerial Vehicle Mechanism and Controller Design Using Webots Simulation

Dwi Pebrianti[1,2,*], Muhammad Syafiq Suhaimi[3], Md. Jobaher Hossain[1], and Luhur Bayuaji[2]

[1]Faculty of Engineering, International Islamic University Malaysia, 53100 Kuala Lumpur, Malaysia.
[2]Faculty of Information Technology, Universitas Budi Luhur, 12260 Jakarta, Indonesia.
[3]Faculty of Electrical and Electronics Engineering Technology, Universiti Malaysia Pahang, 26600 Pahang, Malaysia.

ABSTRACT – Quadrotor Micro Aerial Vehicle (MAV) is one of the Unmanned Aerial Vehicle (UAV) which very versatile with their capabilities for specific purposes. Understanding and controlling MAV is important for various applications, from surveillance and delivery services to disaster response and entertainment. In engineering education, providing a hardware platform for understanding MAV mechanism and the controller design is very challenging in terms of cost and safety issue. This paper presents a study on drone mechanisms and controller design using Webots, a versatile robot simulation environment. The primary objectives of this research were to simulate and analyze the behavior of drones in a line following task, employing PID controller as the control strategies, and to evaluate the suitability of Webots as a simulation tool for drone-related teaching and learning as well as research. In the methodology section, the setup of the Webots-based drone simulation, including the selection of a suitable drone model, sensor configurations, and the implementation of control algorithms will be explained. Line following task will be focused to analyze the drone mechanism and the controller design. Data on the drone performance will be collected through the experiment and will be analyzed rigorously. The findings reveal the effectiveness of Webots as a platform for simulating drone behavior. The advantages of using simulation tools like Webots, including cost-effectiveness, safety, and the ability to test and refine control algorithms before deployment on physical drones was proved in this study. It is shown that Webots can be leveraged to teach and train individuals in drone technology without the need for physical hardware.

ARTICLE HISTORY
Received: 4th Oct 2023
Revised: 20th Oct 2023
Accepted: 30th Oct 2023
Published: 12nd Nov 2023

KEYWORDS
Micro Aerial Vehicle
Engineering Education
Controller design
Webots
Simulator

INTRODUCTION

The rapid increasing of Unmanned Aerial Vehicles (UAVs), commonly known as drones, has transformed various industries, from surveillance and agriculture to logistics and entertainment. As drones continue to evolve and expand their role in our society, understanding their mechanisms and developing effective control strategies have become important.

One of the UAV's tasks that is often highlighted is the line following. There are many types of line following algorithms that are explained details in [1]–[3].

Commonly, a line following quadrotor is using multiple sensors to obtain input in order to gain information for the system to drive the quadrotor to desired position. These sensors included a camera, Inertial Measurement Units (IMU) and Global Positioning System GPS. These sensors will help the system to collect the data and provide necessary information for the system. This system required many sensors because GPS accuracy can be easily affected by the condition in the environment [4]. To overcome this problem, a vision-based system is researched and proposed. There are multiple applications that used quadrotor with a vision-based system such as power line inspection[5] and crop inspection [4]. A vision- based system requires image-processing capabilities to convert the image into usable data for the quadrotor and a controller that will help to stabilize the quadrotor in mid-air. With this combination, the quadrotor can move toward the destination by following the line.

Considering the importance of drone in the real-world applications, it is crucial for engineering students, especially in the field of electrical and electronics engineering, mechatronics engineering as well as mechanical and aerospace engineering to know and understand drones mechanism and the control strategies during their study. However, the real hardware has several limitations which are high cost, the risks during the flight operation, the limitated access, where only some students are able to work using the real hardware in a certain time, risk of damage, as well as the regulatory complince where only certified person can fly the drones.

The primary objectives of this research are twofold. First, we seek to simulate and analyze the behavior of drones and employing control strategy. Second, we aim to evaluate the suitability of Webots, a powerful robot simulation environment, as a pivotal tool for advancing the understanding of drone mechanisms and control.

*CORRESPONDING AUTHOR | Dwi Pebrianti | ✉ dwipebrianti@iium.edu.my

Webots offers a compelling rationale for its inclusion in this study. Its versatility and capabilities in simulating robot and drone behavior make it an ideal choice for the study. By utilizing Webots, the gap between theoretical knowledge and practical application can be reduced. Additionally, Webots helps to increase the students' understanding on drone behavior, fine-tune control algorithms, and assess the drone performance without the constraints and risks associated with real-world drone experimentation.

The remainder of this paper unfolds as follows. In the subsequent sections, an in-depth examination of drone mechanisms and control, offering foundational insights into the technology will be provided. The mthodology will explain the setup and configuration of the Webots-based drone simulations. Following this, the presentation of the experimental result, key findings and the implications will be discussed in detail. The paper will be concluded with a discussion of the advantages and limitations of Webots as a simulation platform and the broader implications of the research.

Through this comprehensive exploration, we aim to contribute to the expanding of drone technology, to give an effective way of teaching drone mechanisms and control while showcasing the invaluable role of simulation in advancing the understanding of this transformative technology.

RELATED WORK

The integration of robotic simulators in engineering education has gained increasing attention in recent years. This section reviews the key literature and related research in this field, highlighting the develoment and significance of robotics simulators in enhancing engineering education.

Simulations-Based Learning

Simulation-based learning has been widely adopted in various educational domains to provide students with hands-on experience and practical skills development. In the context of engineering education, simulators have become valuable tools for teaching complex concepts and enabling students to experiment with real-world scenarios in a safe and controlled environment. Researchers such as Joventino et.al [6] and Lammle et. al [7] have explored the effectiveness of simulation-based learning approaches and have demonstrated their potential to improve student engagement and learning outcomes.

Robotics Simulators in Engineering Education

Robotic simulators are computer programs or platforms that replicate the behavior and functionality of physical robots. They provide students with a virtual environment where they can design, program, and test robotic systems without the need for expensive hardware. Recent research by Chiou et. al [8] and Yue et. al [9] have highlighted the growing role of robotic simulators in engineering education. These simulators offer numerous advantages, including cost-effectiveness, scalability, and the ability to facilitate remote learning, which became especially important during the COVID-19 pandemic.

METHODOLOGY

In this section, the methodology employed in the study to simulate and analyze drone mechanisms and control using Webots will be discussed. The discussion will cover all the aspects of the simulation from the system configuration up to the analysis of the system performance. The explanation about the controller design of the MAV based on the simulation data will also be presented.

Setup and Configuration of Drone Simulation

The core of the research lies in the setup and configuration of the drone simulation within the Webots environment. To replicate real-world scenarios as faithfully as possible, a virtual environment that emulates diverse landscapes and conditions will be desinged. This virtual environment includes urban landscapes, natural terrains, and indoor settings, each posing unique challenges for drone navigation and control. In this study, line following task will be focused to obtain a more comprehensive understanding about the drone mechanism and the control strategy.

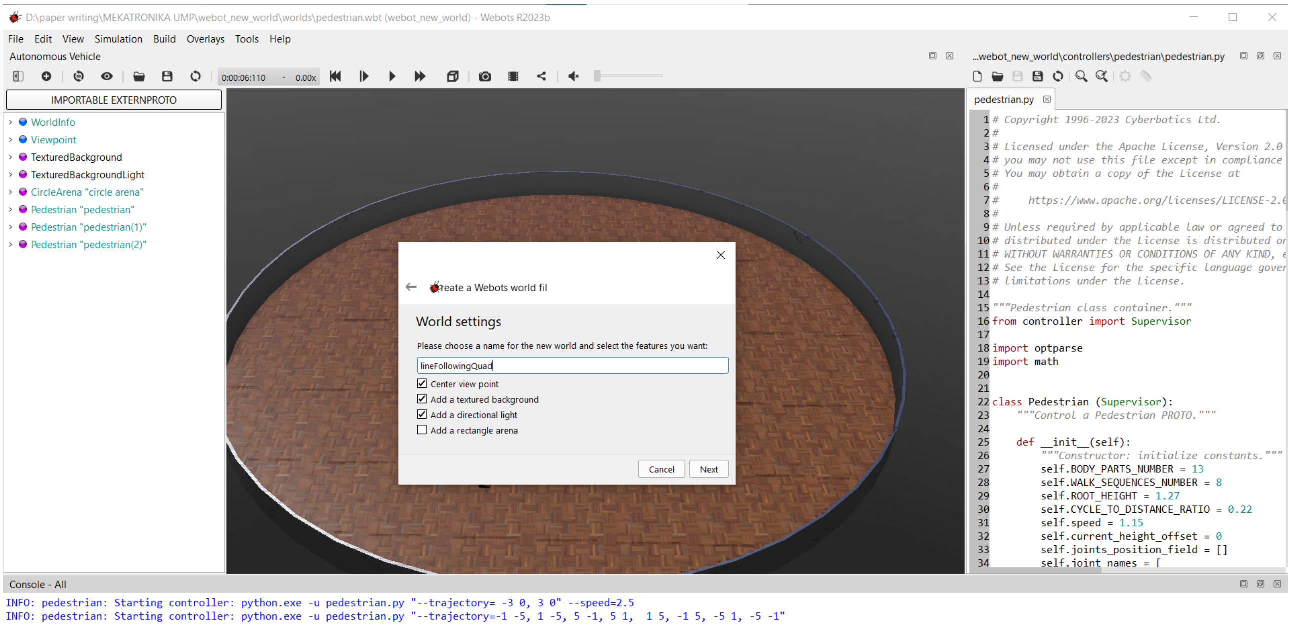
Within Webots, the necessary parameters, including terrain topography, weather conditions, and lighting effects are establshed. This comprehensive setup allows us to conduct experiments that closely resemble real-world situations, providing a robust testing ground for drone mechanisms and control.

To create a new world in Webots, select "File" > "New" > "New World File". A popup window will appear as shown in Figure 1, where user needs to give the world name.

Choice of Drone Model and Sensors

Selecting an appropriate drone model and sensor suite is pivotal to the accuracy and relevance of the simulations. After careful consideration, this study opted for the Mavic 2 Pro from DJI, known for its versatility and widespread use in both hobbyist and professional applications. This choice ensures that the simulations align with real-world scenarios and facilitate the development of practical insights.

For the line following task, a suite of sensors commonly found in commercial drones, including GPS, IMU (Inertial Measurement Unit), and camera are included in the model. These sensors enable the virtual drones to perceive their surroundings and collect data essential for navigation, obstacle avoidance, and other control tasks.



D:\paper writing\MEKATRONIKA UMP\webot_new_world\worlds\pedestrian.wbt (webot_new_world) - Webots R2023b
File
Edit
View
Simulation
Build
Overlays
Tools
Help
Autonomous Vehicle
0:00:06:110
0.00x
IMPORTABLE EXTERNPROTO
WorldInfo
Viewpoint
TexturedBackground
TexturedBackgroundLight
CircleArena "circle arena"
Pedestrian "pedestrian"
Pedestrian "pedestrian(1)"
Pedestrian "pedestrian(2)"
...webot_new_world\controllers\pedestrian\pedestrian.py
pedestrian.py
Console - All

reate a Webots world fil

World settings

Please choose a name for the new world and select the features you want:

lineFollowingQuad

☑ Center view point
☑ Add a textured background
☑ Add a directional light
☐ Add a rectangle arena

Cancel Next

```
# Copyright 1996-2023 Cyberbotics Ltd.
#
# Licensed under the Apache License, Version 2.0
# you may not use this file except in compliance
# You may obtain a copy of the License at
#
#     https://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to
# distributed under the License is distributed or
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, e
# See the License for the specific language gover
# limitations under the License.

"""Pedestrian class container."""
from controller import Supervisor

import optparse
import math


class Pedestrian (Supervisor):
    """Control a Pedestrian PROTO."""

    def __init__(self):
        """Constructor: initialize constants."""
        self.BODY_PARTS_NUMBER = 13
        self.WALK_SEQUENCES_NUMBER = 8
        self.ROOT_HEIGHT = 1.27
        self.CYCLE_TO_DISTANCE_RATIO = 0.22
        self.speed = 1.15
        self.current_height_offset = 0
        self.joints_position_field = []
        self.joint_names = [
```

```
INFO: pedestrian: Starting controller: python.exe -u pedestrian.py "--trajectory= -3 0, 3 0" --speed=2.5
INFO: pedestrian: Starting controller: python.exe -u pedestrian.py "--trajectory=-1 -5, 1 -5, 5 -1, 5 1,  1 5, -1 5, -5 1, -5 -1"
```

Figure 1. Creating New World in Webots

Control Algorithm and Approaches

The heart of the simulation lies in the implementation of control algorithms and approaches. As the preliminary step, PID controller is applied in the study. The PID controller will be used for stabilization, path planning algorithms, and line following task. The effect of each gain in the PID cntroller will be meticulously configured and fine-tuned to suit the specific scenarios presented in the experiments.

1) Image Processing

A camera is used as the sensor for completing the line following task. The image processing used in this study is shown in Figure 2. The line detection is conducted by utilizing several image processing algorithm. The first step is to change the Red Green Blue (RGB) image into grayscale. The next step is to change the grayscale image into binary image. The purpose of this step is to simplify the image data and enhances the visibility of specific features or objects of interest. It plays a crucial role in various image processing tasks, including segmentation, feature extraction, object detection, and pattern recognition. The choice of thresholding method and threshold value depends on the specific application and characteristics of the grayscale image.

Once the binary image is obtained, the image will be processed into a skeletonized binary image. The purpose of this process is to again simplify the image and extract essential structural information from the object pattern, which in this study is the line on the ground. Compared to binary image, skeletonized binary image will have even more small pixel size. This will make the line detection process will be faster.

The next step is to implemeny Hough transformation to the skeletonized binary image. The Hough Transform is a powerful technique used for detecting lines and other parametric shapes in binary images, including skeletonized binary images. The Hough Transform is particularly well-suited for line detection because it can robustly identify lines even when they are broken, curved, or partially obscured.

Hough Transform is particularly useful because the skeletonization process often simplifies complex shapes into single-pixel-wide curves or lines. By applying the Hough Transform to a skeletonized image, effective identification and characterization of the structural components, in this study is line on the ground can be conducted. Additionally, Hough Transform is a well known algorithm and commonly used due to its robustness, parameterization capabilities, accuracy, and adaptability to various line types.

2) PID Control

Control is the most important part in MAV. For the task of Line Following Quadrotor, the controller design aims to stabilize the system and instruct the quadrotor to achieve the objective. In the quadrotor system, the angles that need to be controlled are the roll, pitch and yaw angles. This objective can be done through the Proportional (P) controller which is able to control the altitude and attitude of the quadrotor. By using the P controller, the quadrotor is able to detect the line while stabilizing in mid-air.

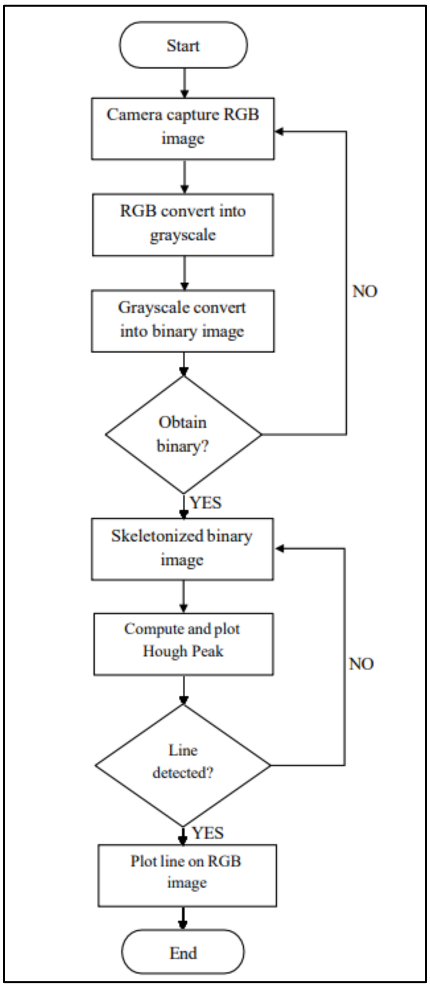

Start
Camera capture RGB image
RGB convert into grayscale
Grayscale convert into binary image
Obtain binary?
NO
YES
Skeletonized binary image
Compute and plot Hough Peak
Line detected?
NO
YES
Plot line on RGB image
End

Figure 2. Flowchart of Image Processing

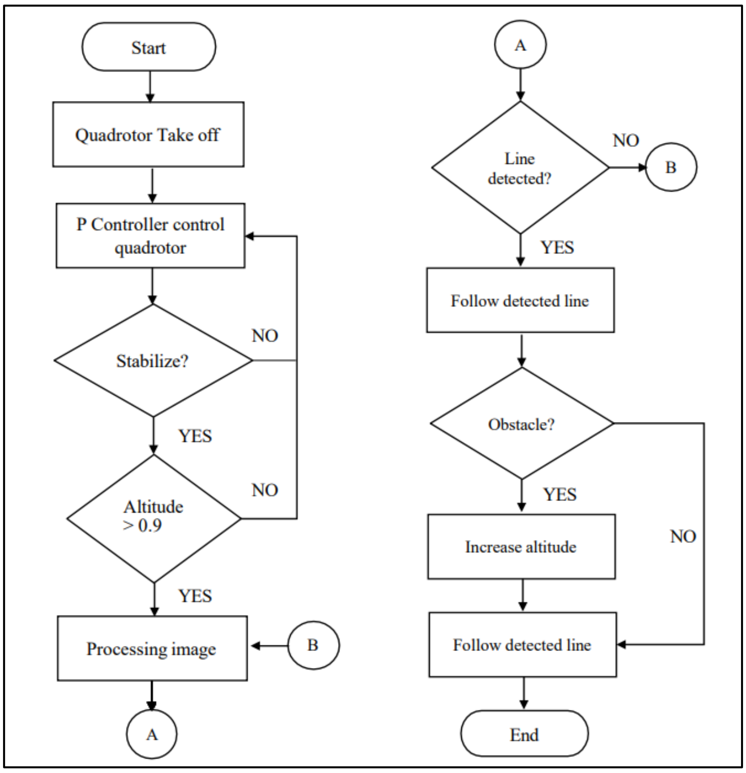


Figure 3. Flowchart of Line Following Task

Figure 3 shows the flowchart of the proposed system implemented for the study. The selection of the P gain will be done through Simulink simulation in the first step. Once the time response of the system is appropriate where the value of overshoot is not more than 20% then the controller will be implemented in Webots environments.

The block diagram of the proposed system is shown in Figure 4. In this step, the controller for image processing part is set to be 1, which is unity gain in Simulink simulation. Therefore, the main concern in the controller design is the altitude control and also the attitude control which includes the Roll, Pitch and Yaw movement.

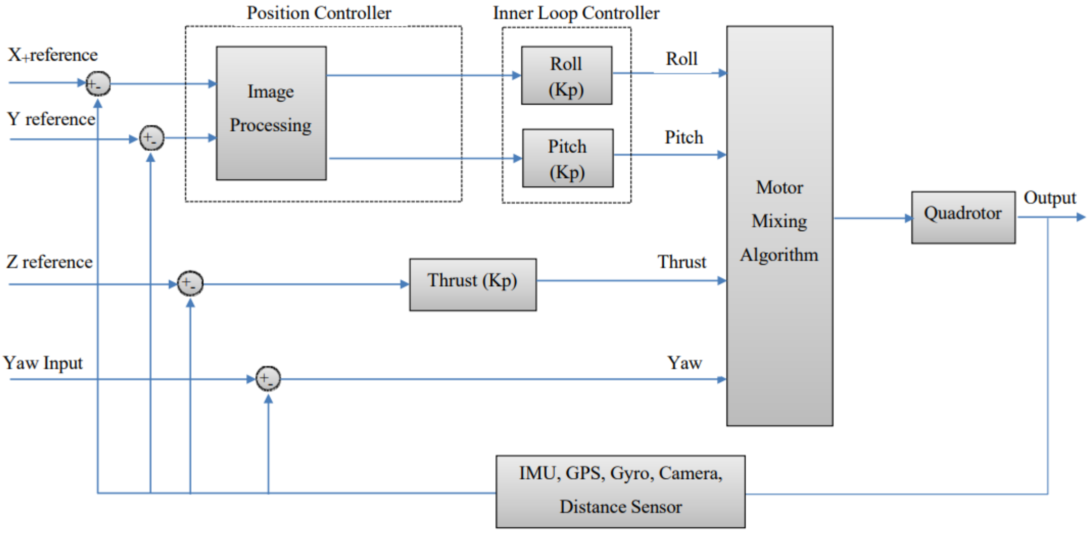


Figure 4. Block Diagram of Line Following MAV

RESULT AND DISCUSSION

Webots Environment and Platform Selection

After a few study literature, Figure 5 is created as the experimental environment for the line following task using autonomous quadrotor. The starting point is indocated with a quadrotor image. The quadrotor will need to follow a straight path qoing forward, then turn left, near a wall the quadrotor will need to turn right. The quadrotor will follow another straight path until it almost reach to the wall and needs to make a right turn. In this part, an obstacle is created to see the capability of the quadrotor to avoid it. After the obstacle is avoided, quadrotor will need to fly back to a certain level of altitude and go straight line until it reaches the goal position.

As seen in the scenario, the environment will need the quadrotor to be able to 1) following straight line, 2) make a right or left turn based on the line images 3) avoid obstacle and lastly 4) stop when it reaches the goal position.

Figure 6 shows the model used as the quadrotor platform in the study. The model chosen is DJI Mavic 2 Pro which includes several sensors onboard such as gyro, compass, camera for capturing the line, distance sensor to detect obstacle in front, IMU for attitude measurement and also GPS for the positioning.

Image Processing

As mentioned in the methodology part, this study will implement the RGB to grey scale trasformation, followed by the grey image to binary image transformation, binary to skeletonized binary image transformation and lastly the Hough transform. The example of the image processing result is shown in Figure 7 and Figure 8. In Figure 7, the transformation from the RGB to grey and then to binary and skeletonized binary images are presented. It is seen that the shape of the line becomes thinner and small in pixel size. This information is really crucial to decrease the processing time for controller to give command to the quadrotor. By redusing the processing time, it is expected that the quadrotor will be able to detect and follow line more accurate. The result for this statement will be presented in the next section.

Figure 8 shows the Hough transform result from image in Figure 7 (d), and the detected line. The skeletonized line in Figure 7 (d) will be computed using the Hough Transform function in the Matlab application. After that, the peaks for the Hough Transform will be calculated and plot in the graph. Figure 8 (a) shows the peak plotted in the Hough Transform. Lastly, the number of Hough peaks in the Hough Transform will determine the number of lines detected in the image. From Figure 8 (a), a single Hough peak is plotted in the graph. Thus, only one line will be available in the RGB image captured by the quadrotor's camera. Figure 8 (b) shows the detected line where the center line is marked with the blue line on the picture. As for the yellow marker, it is the starting and end point of the line detected which has been set up in the program based on the desired length of the line.

Performance of PID Controller for Hovering Task

As the preliminary step of the study, Proportional (P) controller is implemented in the study. In the future the combination of Proportional, Integral and Derrivative (PID) controller will be considered.

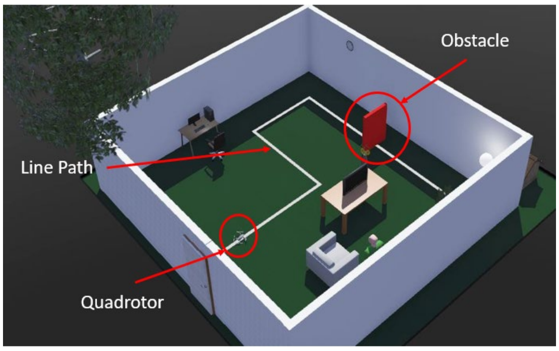


Figure 5. Block Diagram of Line Following MAV

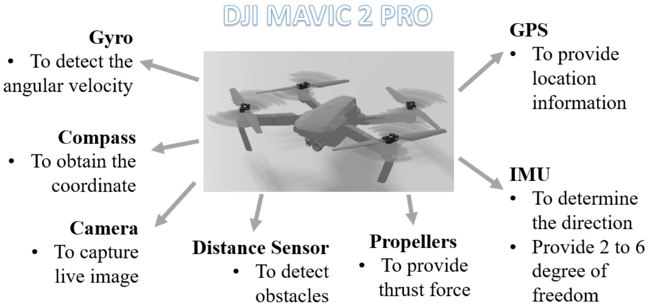
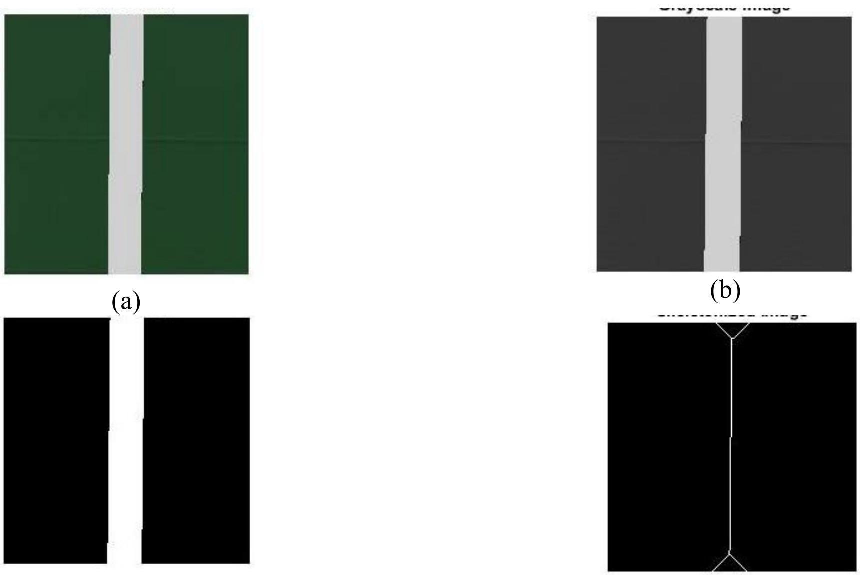

DJI MAVIC 2 PRO
Gyro
• To detect the angular velocity
Compass
• To obtain the coordinate
Camera
• To capture live image
Distance Sensor
• To detect obstacles
Propellers
• To provide thrust force
GPS
• To provide location information
IMU
• To determine the direction
• Provide 2 to 6 degree of freedom

Figure 6. DJI Movic as Platform



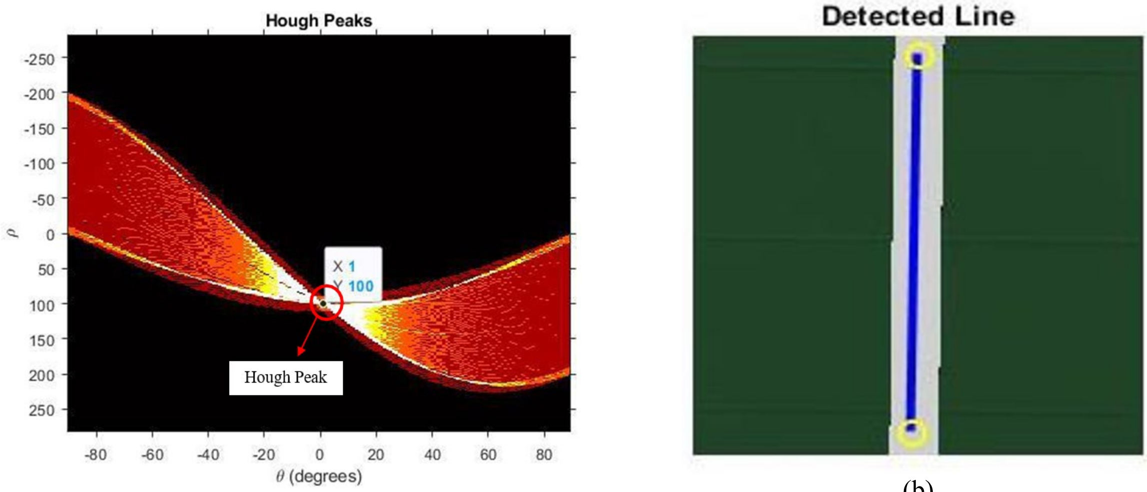
(a)

(b)

(c)

(d)

Figure 7. (a) RGB (b) Grey Scale (c) Binary (d) Skeletonized Binary



(a)

(b)

Figure 8. (a) Hough Peak (b) Detected Line

The experiment is conducted by considering a certain P gain for different altitude and attitude control as tabulated in Table 1. In this study, for attitude control, only Roll and Pitch movements are considered. For this part, only the hovering characteristic will be evaluated which means the P-value of roll and pitch will be constant and the value of P for thrust increasing from one to five. The characteristic of the quadrotor during hovering will be shown using Matlab graphs.

Figure 9 shows the result of different values of P gain for hovering task. The time response analysis is tabulated in Table 2. From the result, it is seen that there is no value of P controller gain that can achieve best performance for all the parameters covered in time response analysis. For example, $K_p = 1$ gave the fastest settling time and the lowest overshoot percentage, which are good. However, it has the highest steady state error. Highest steady state error is commonly unwanted. This is due to not achieving the target or the goal set to the system. Meanwhile, for $K_p = 5$, it has 15.46 seconds of settling time, 0.01 meter of steady state error and 64.59% of overshoot percentage. In terms of steady state error, this controller gain has the best performance. However, the system is slow and has very high overshoot which is unpreferable for real system. This is because, a very high overshoot value will cause damage to the system. Hence, another controller gain needs to be added, which will be the future works of the study.

Table 1. P Gain Values for Hovering.

Thrust (K_{pz})	Roll (K_{py})	Pitch (K_{px})
1	50	30
2	50	30
3	50	30
4	50	30
5	50	30

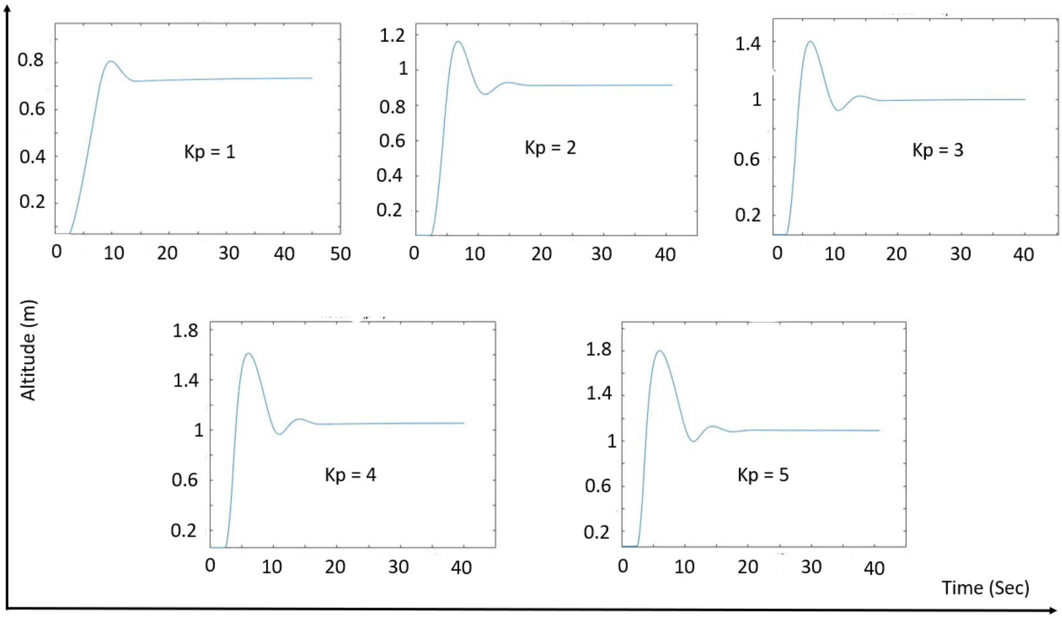


Figure 9. Different values of P controller gain for hovering task

Table 2. Time Response Analysis for Different P Gain Values, with Altitude Target is 1 m.

Thrust (Kp)	**Settling Time (Sec)**	**Rise Time (Sec)**	**Steady state Error (m)**	**Peak Time (Sec)**	**Overshoot %**
1	**12.24**	4.16	0.33	9.80	**9.01**
2	12.67	1.86	0.17	6.78	27.12
3	14.96	1.38	0.09	6.23	40.27
4	15.19	1.15	0.04	6.07	72.74
5	15.46	**1.00**	**0.01**	**6.01**	64.59

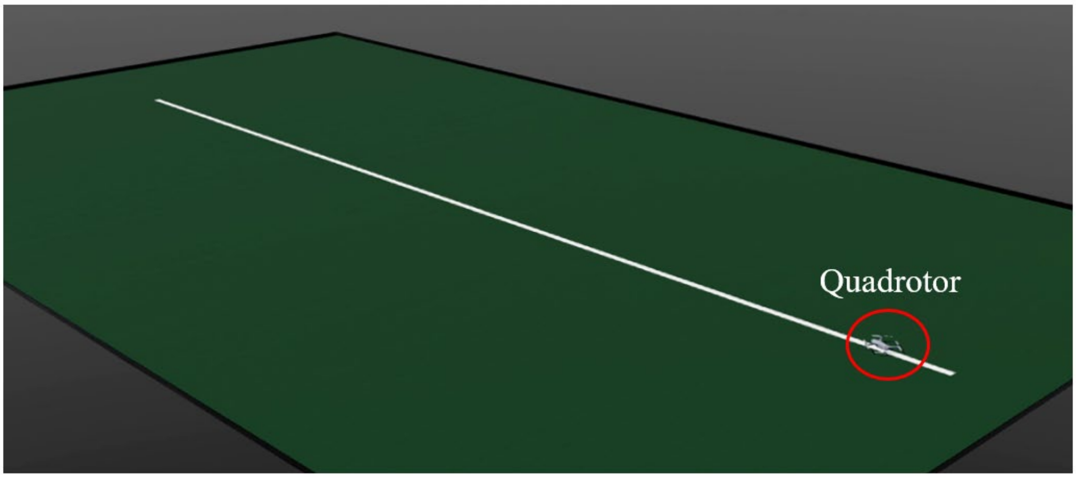


Figure 10. Webots Environment for Line Following Task.

Line Following Task

Figure 10 shows the Webots environment for conducting line following task. The task will be started from take off condition until 1 meter of altitude then followed by the detection of the line and the quadrotor must follow the line. The video of the simulation can be accessed through link untuk video following, https://bit.ly/followLine. Additionally, for the full task, which is to complete the environment in Figure 5, can be accessed through this link, https://bit.ly/FullTask.

To conduct the complete task, three different combination of P controller gain for Roll and Pitch are implemented. While, the P gain for the altitude controller is set to be $K_p = 3$ base on result in Table 2. The time response analysis for the line following task completed with obstacle avoidance is summarized in Table 3. From the result, it is seen that the system reached the best performance when $\boldsymbol{K_{py}} = 50$ and $\boldsymbol{K_{px}} = 30$. Figure 11 shows the performance of the system in terms of its position. Based on the graph plotted for the X and Y position of the quadrotor, during the line following, it is clearly shown that the quadrotor is successfully following the line. The evidence from the video is also showing that Webots can be used as a tools to understand the mechanism of quadrotor and to prove the controller algorithm that has been designed.

Table 3. Time Response Analysis for Line Following with Obstacle Avoidance with $\boldsymbol{K_{pz} = 3}$.

Roll (K_{py})	**Pitch (K_{px})**	**Settling Time (Sec)**	**Max Altitude (m)**	**Time taken to Goal Point (Seconds)**
50	30	**17.47**	1.69	**58.71**
55	35	23.33	1.68	63.41
60	40	24.09	1.68	69.58

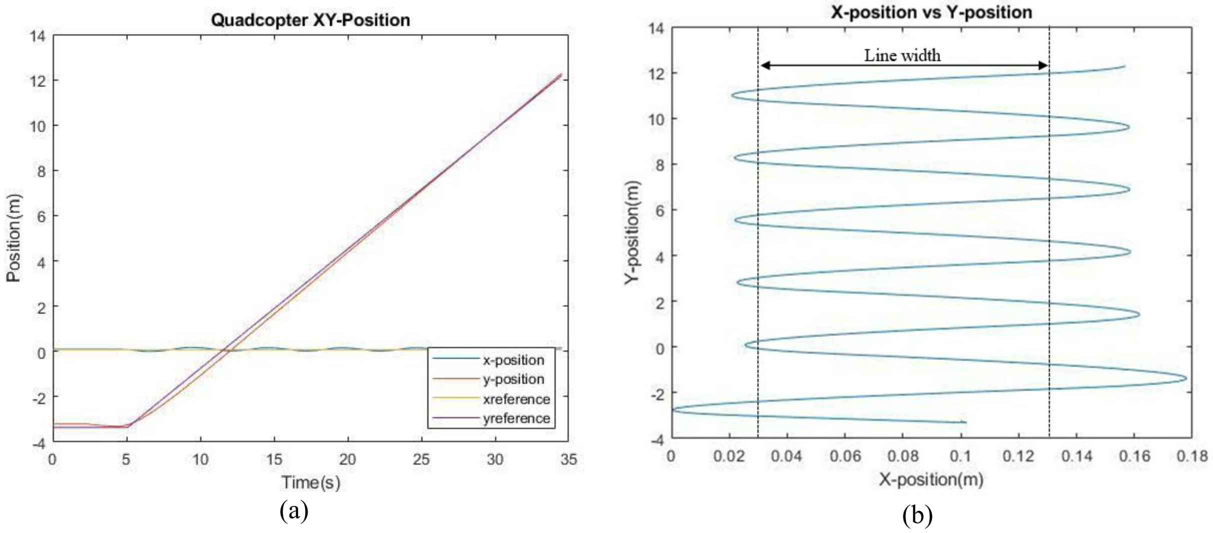


Figure 11. (a) XY Position Vs Time (b) X Position Vs Y Position

CONCLUSION

In this project, a quadrotor MAV model is used in the Webots application for simulation purposes. The quadrotor was able to detect and follow the line on the ground which is processed through image processing in Matlab. The controller used is based on a P controller that is simpler compared to a PID controller that can produce better results.

In the nutshell, vision-based system is suitable for line detection through a simple algorithm and produced desired results. During the hovering process, the results showed that the lowest percentage of overshoot is 9.01% and the settling time is 12.24 seconds. However, the quadrotor was not able to reach the target altitude. Then, for following a straight line fastest time taken for quadrotor to reach at the end of line is 26.68 seconds. Next, from the full simulation the shortest time taken to reach destination is 58.71 seconds. From here, we can conclude that the lower value of P controller produced the best results in term of settling time, overshoot percentage and time taken for the quadrotor to reach its destination. For the controller, the higher value of P will result in improving the position of the quadrotor. However, the time consumed for the quadrotor to reach its destination increases.

This study shows that Webots can be used as a platform for the teaching and learning process of the drone, quadrotor mechanism, image processing and the controller design to conduct a certain task, which in this case is line following and obstacle avoidance.

For the future works, the study can be extended as the platform to do the study regarding the optimization of controller design for autonomous quadrotor. Through the simulator environment, students will be able to experience, see, examine and analyze the performance of the quadrotor similar to the real hardware.

REFERENCES

[1] P. B. Sujit, S. Saripalli, and J. B. Sousa, “Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless,” *IEEE Control Syst*, vol. 34, no. 1, pp. 42–59, Feb. 2014, doi: 10.1109/MCS.2013.2287568.

[2] Willson Amalraj Arokiasami, Prahlad Vadakkepat', Kay Chen Tan, and Dipti Srinivasan, “Vector Directed Path Generation and Tracking for Autonomous Unmanned Aerial/ Ground Vehicles,” in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 1375–1381.

[3] Jyi-Shane Liu and Gong-Yi Lee, “A Carrot in Probabilistic Grid Approach for Quadrotor Line Following on Vertical Surfaces,” in *2019 International Conference on Unmanned Aircraft System (ICUAS)*, 2019, pp. 1234–1241.

[4] A. S. Brandão, F. N. Martins, and H. B. Soneguetti, “A Vision-based Line Following Strategy for an Autonomous UAV,” 2015.

[5] Sepehr Valipour, Hosein Khandani, and Hadi Moradi, “The Design and Implementation of a Hotline Tracking UAV,” 2015, pp. 252–258.

[6] C. F. Joventino, R. de A. A. e. Silva, J. H. M. Pereira, J. M. S. C. Yabarrena, and A. S. de Oliveira, “A Sim-to-real Practical Approach to Teach Robotics into K-12: A Case Study of Simulators, Educational and DIY Robotics in Competition-based Learning,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 107, no. 1, Jan. 2023, doi: 10.1007/s10846-022-01790-2.

[7] Arik Lammle *et al.*, “Simulation Based learning of the Peg-in-Hole Process Using Robot-Skills,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 9340–9346.

[8] R. Y. Chiou, “Implementing Virtual Reality Project Activities for Enhancing Student Learning Experience in Robotics and Automation,” 2023.

[9] H. Yue, J. Miao, J. Zhang, C. Fan, and D. Xu, “Simulation for senior undergraduate education of robot engineering based on Webots,” *Computer Applications in Engineering Education*, vol. 29, no. 5, pp. 1176–1190, Sep. 2021, doi: 10.1002/cae.22377.