Robot Operating System (ROS) in Quadcopter Flying Robot Using Telemetry System

Mohammad Iqbalul Faiq Hatta [a,1,*], Nuryono Satya Widodo [a,2]

[a] Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] hattafaiq@gmail.com; [2] nuryono.sw@uad.ac.id
*Corresponding Author

ARTICLE INFO

Article history
Received 05 February 2021
Revised 15 February 2021
Accepted 21 February 2021

Keywords
Flying Robot;
Robot Operating Sysem (ROS);
Quadcopter;
Telemetry

ABSTRACT

In this study implementing odometry using RVIZ on a quadcopter flying robot that uses the Pixhawk Cube firmware version 3.6.8 as the sub-controller. Then the Lenovo G400 laptop as the main-controller as well as the Ground Control Station using the ubuntu 16.04 linux operating system. The ROS platform uses the Kinetic and MAVROS versions as a quadcopter platform package using MAVlink communication with the telemetry module. The odometry system was tested using Rviz as a navigation for Quadcopter movements in carrying out movements that follow movement patterns in certain shapes and perform basic robot movements. Data were collected using a standard measuring instrument inclinometer as a measurement of the slope of the robot and visualization RVIZ as a visual display of the odometric robot. The results of the research obtained are that the flying robot can maneuver according to the shape on the RVIZ according to the movements carried out directly at the airport, as well as the effect of roll angle on the quadcopter (negative left roll, positive right) and the pitch angle on the quadcopter (negative forward pitch, the pitch returns positive).

1. Introduction

The Indonesian Flying Robot Contest has several divisions, one of which is Vertical Take Off Landing (VTOL). The robot used in the robot contest is in the form of a multirotor or a flying robot that uses several motors as the main driving force for flying movements. The type of multirobot used is the Quadcopter / Quadrotor which is a type of unmanned aircraft using four motors that are used to perform flight movements. Each year the VTOL division has a different mission or obstacle. In general, robots carry out missions in an automatic way without controlling them using the remote, but by pressing one command button to complete the mission. The automatic control system used by Quadrotor is using the Linux operating system with a program built from the dronekit open source platform, and the programming language used is python [1].

http://dx.doi.org/10.31763/ijrcs.v1i1.247

The problem that is often encountered in Quadrotor is evaluating the behavior of the flying robot when it will determine the best movement when conducting research or flight trials. Therefore, real time monitoring is needed and can be presented in three dimensions on the Linux operating system which can support the evaluation of flight behavior in robots. In this research, we will propose about odometry using RVIZ. Odometry is the use of data from robot motion sensors to estimate the position of the robot. Meanwhile, RVIZ is a service provided by Robot Operating Systems (ROS) that can display information visually on quadcopter flying robots [2].

2. Quadcopter Robot

One type of UAV (unnamed Aerial Vehicle) that is currently being studied is a quadcopter [3]. A helicopter type UAV that uses four motors to move the propeller. The strength of a quadcopter is its ability to take off and land vertically. In general, a quadcopter is controlled via a radio transmitter remotely by humans so that control errors often occur, especially in the landing mechanism.

2.1. Quadcopter Robot Frame Design

Robot modeling using four rotors can be seen in Fig. 1, there are three axes x, y, and z as the positioning of the accelerometer, gyro, and compass sensors to determine the position of the robot so that it can be adjusted to the robot's movement displayed through RVIZ.

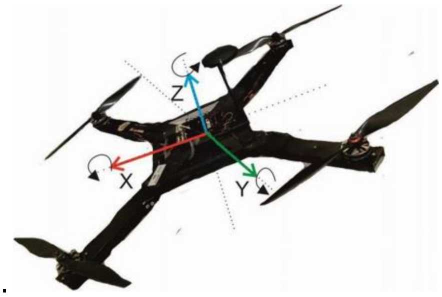


Fig. 1. Robot Axis Diagram

2.2. System Design

The diagram of a robot system using 4 rotors as a driver with 4 ESCs as motor drivers can be seen in Fig. 2. Pixhwak is flight controller, telemetry is a communication protocol between a robot and a laptop is a Ground Control Station (GCS).

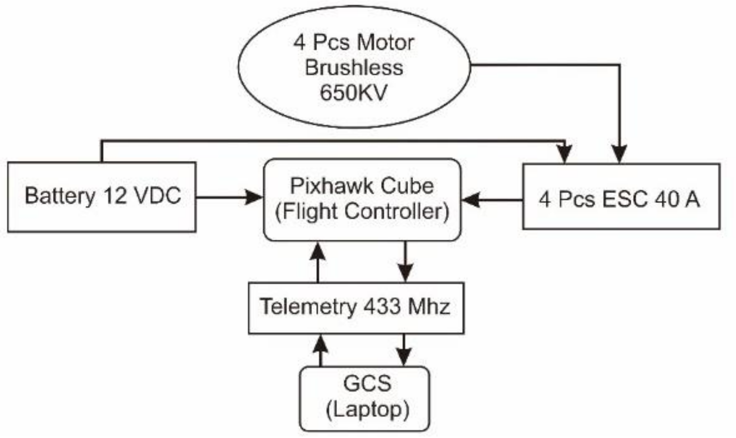


Fig. 2. System Block Diagram

3. Quadcopter Odometry System

3.1. Orientation

Orientation is a vector rotation in the form of a matrix that is used to determine the results of changes in pitch, roll, and yaw from processing data taken from the IMU sensor [4]. The orientation of the robot is influenced by two parts of the rotation equation according to its respective functions which can be seen in Fig. 3, the "base link" rotation equation is a rotation used as a parameter of child rotation or robot rotation and "aircraft" is used as the main rotation or movement medium. The quarternion is part of the orientation unit where it is divided into two parts, namely using NED (north-east-down), namely the axis consisting of north, east, and down and ENU (east-north-up), which is the part of the axis consisting of east, north, and up.

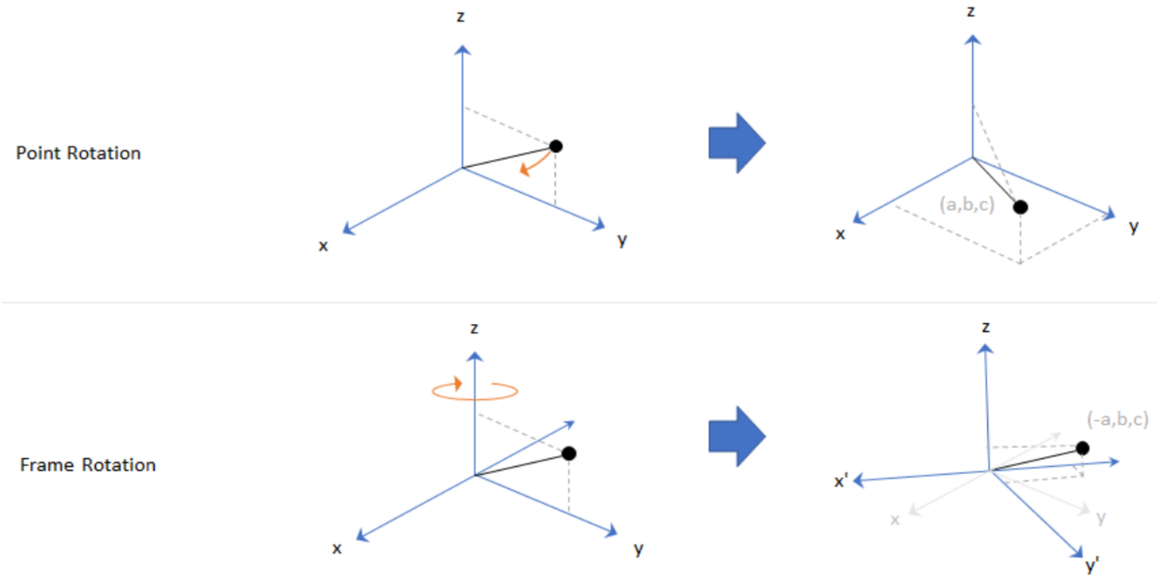


Fig. 3. Vector Rotation

The quarternion determination uses the gyro value instead of the eulerian angle equation. The results of the value from the raw gyro data are then processed by EFK (Extended Kalman Filter) which is a filter used to stabilize the flying attitude of the robot, so that coovariance is obtained in each determination of the robot's flying attitude as the basic value of the change in direction in the matrix [5]. Determination of the position of the robot can use DCM (Direction Cosine Matrix) from the value of the accelerometer and gyroscope to get real time measurements, so that the information that can be received is the euler angle in the form of roll, pitch, and yaw [6]. The orientation value is the variable value x, y, z. Where, the values of x, y, z are influenced by the size of the angle in degrees, which is the result of changes in drone position [7]. To obtain the orientation value, the following equation is used:

$$\mathrm{X} = \mathrm{R}(\emptyset) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\emptyset & -\sin\emptyset \\ 0 & -\sin\emptyset & \cos\emptyset \end{bmatrix} \quad (1)$$

$$\mathrm{Y} = \mathrm{R}(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad (2)$$

$$\mathrm{Z} = \mathrm{R}(\varphi) = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

The value of the x, y, z orientation is used to determine the angular velocity value for each variable x, y, z with the equation

$$V_x = \text{nilai } \textit{orientation} \text{ x/ t} \tag{4}$$

$$V_y = \text{nilai } \textit{orientation} \text{ y/ t} \tag{5}$$

$$V_z = \text{nilai } \textit{orientation} \text{ z/ t} \tag{6}$$

Where V_x is the angular velocity x (rad/s), V_y is the angular velocity y (rad/s), V_z is the angular velocity z (rad/s), t is the time in second (s).

The liner velocity equation is

$$V_{xl} = 2\pi r \text{ x/ T} \tag{7}$$

$$V_{yl} = 2\pi r \text{ y/ T} \tag{8}$$

$$V_{zl} = 2\pi r \text{ z/ T} \tag{9}$$

Where V_{xl} is the linear velocity x (m/s), V_{yl} is the linear velocity y (m/s), V_{zl} is the linear velocity z (m/s), and T is the time period in the second (s) unit.

The linear velocity x, y, z values are used to obtain the linear acceleration value with the following equation:

$$\dot{a}x = V_{xl}/t \tag{10}$$

$$\dot{a}y = V_{yl}/t \tag{11}$$

$$\dot{a}z = V_{zl}/t \tag{12}$$

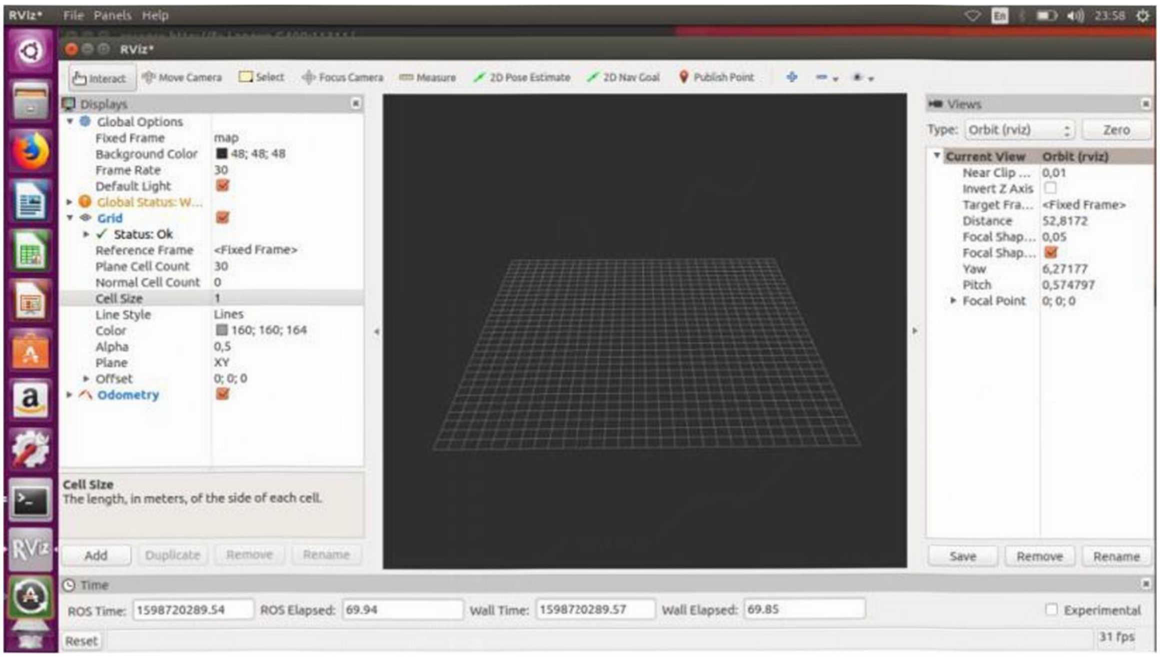
To obtain total linear acceleration, equations are used

$$\begin{bmatrix} ax \\ ay \\ az \end{bmatrix} = \begin{bmatrix} \dot{a}x \\ \dot{a}y \\ \dot{a}z \end{bmatrix} + \text{g} \begin{bmatrix} \sin\theta \\ -\cos\theta \sin\emptyset \\ -\cos\theta\cos\emptyset \end{bmatrix} \tag{13}$$

With the total acceleration ax detected by the accelerometer sensor on the x-axis (m/s^2), ay the total acceleration detected by the accelerometer sensor on the y-axis (m/s^2), az the total acceleration detected by the accelerometer sensor on the z-axis (m/s^2), (ax)$^{\cdot}$ x-axis linear acceleration (m/s^2), (ay)$^{\cdot}$ y-axis linear acceleration (m/s^2), (az)$^{\cdot}$ z-axis linear acceleration (m/s^2), and T is the time in second (s)

3.2. Rviz

Robot Operating System or ROS is a platform for developing Robot applications [8]. Some examples such as sending and receiving information between one node and another node in a topic is one of the basic functions of ROS [9]. Utilization of ROS in flying robots is often used for real time mapping and data retrieval [10]. One of the features of ROS is Rviz which functions to display two-dimensional and three-dimensional simulations, some of the conditions provided by MAVROS [11] in utilizing Rviz as a GUI (Graphical User Interface) display for linux, can be seen in Fig. 4. MAVROS Services [12] which can be seen in Fig. 4 sends information from several IMU 3 DOF data nodes, mission way points, battery status, attitude, acceleration, and actuator controls. This study uses Rviz as a navigation for quadcopter movements in carrying out movements that follow movement patterns in certain forms. The settings applied to Rviz as a monitoring medium use the grid function as a mapping form of thirty squares as shown in Fig. 4 in the middle.



Global Options
Fixed Frame: map
Background Color: 48; 48; 48
Frame Rate: 30
Default Light: ✓
Global Status: W...
Grid: ✓
Status: Ok
Reference Frame: <Fixed Frame>
Plane Cell Count: 30
Normal Cell Count: 0
Cell Size: 1
Line Style: Lines
Color: 160; 160; 164
Alpha: 0,5
Plane: XY
Offset: 0; 0; 0
Odometry: ✓

Cell Size
The length, in meters, of the side of each cell.

Type: Orbit (rviz)
Current View: Orbit (rviz)
Near Clip ...: 0,01
Invert Z Axis: □
Target Fra...: <Fixed Frame>
Distance: 52,8172
Focal Shap...: 0,05
Focal Shap...: ✓
Yaw: 6,27177
Pitch: 0,574797
Focal Point: 0; 0; 0

ROS Time: 1598720289.54 ROS Elapsed: 69.94 Wall Time: 1598720289.57 Wall Elapsed: 69.85 □ Experimental
31 fps

Fig. 4. RVIZ Window

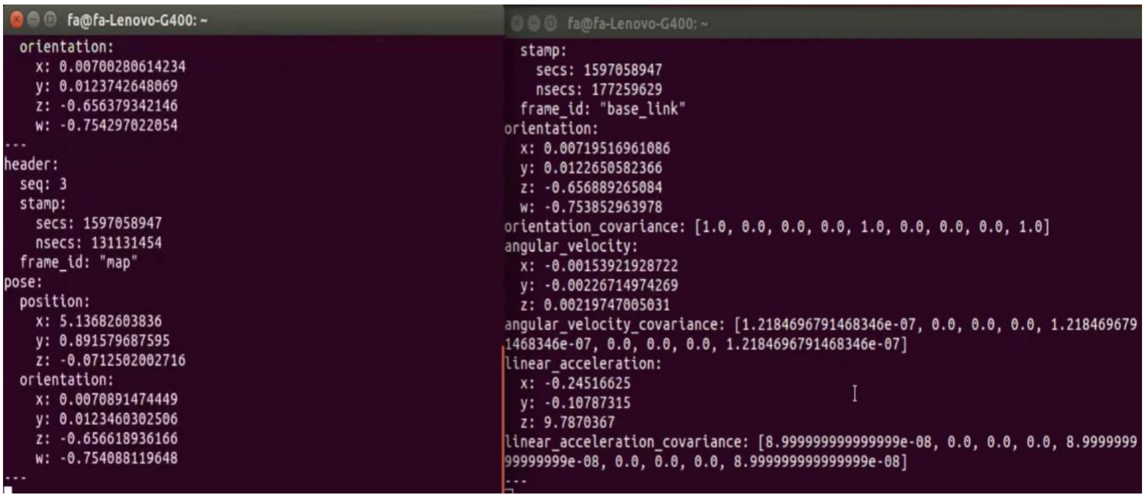
4. Result and Discussion

To determine the success rate of the robot in implementing a system that has been built using ROS, it is necessary to have a testing stage for each sub-system on the robot. The first test is hardware testing and system integration or system communication between the use of telemetry and GCS (laptop) which is connected so that the system can run according to using Rviz.

The second test is testing the movement accuracy between the Rviz navigation system and the movements carried out by the robot in simulating maneuvers using basic forward, backward, right-side and left-side movements. Testing the accuracy of the Movement was carried out in the Kotagede area, Yogyakarta. The third test is to carry out the test by flying the robot according to several types of movements carried out, including a square motion with each side having a side length of five meters, then performing a triangular motion with each side having a side length of five meters, and a movement that resembled a side length of five meters. Movement testing that resembles a square, triangle and the letter "V" was carried out in the Godean area, Yogyakarta.

4.1. Testing the Local Position Topic Axis

In the hardware and software configuration process, telemetry is required as a link between the two so that to find out whether the configuration process is carried out correctly or successfully, you can test the call of one of the commands from the ROS or call the topic provided by Mavros as in Fig. 5. After testing the function basic has been fulfilled, then next check on RVIZ by calling the topic "mavros / local position / local" on RVIZ. These stages can be seen in Fig. 6 by adjusting the shape of the robot modeling in the direction of the arrow which is blue. In the direction of the blue arrow shows the x-axis of the robot, so it can be seen that the robot is in the x-axis position on the left or the robot's face is facing to the left.

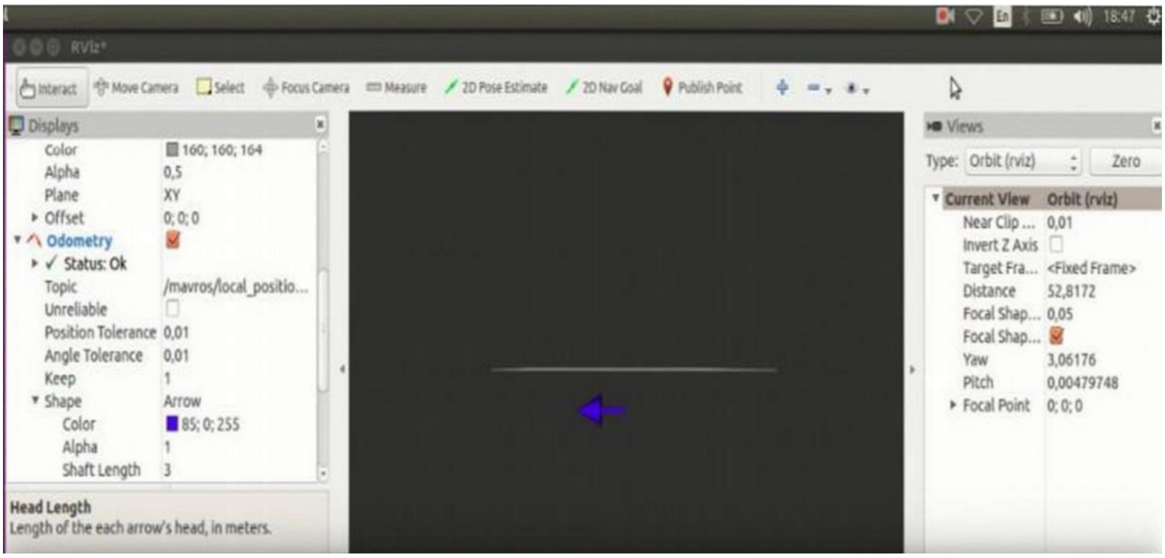


```
fa@fa-Lenovo-G400: ~
  orientation:
    x: 0.00700280614234
    y: 0.0123742648069
    z: -0.656379342146
    w: -0.754297022054
---
header:
  seq: 3
  stamp:
    secs: 1597058947
    nsecs: 131131454
  frame_id: "map"
pose:
  position:
    x: 5.13682603836
    y: 0.891579687595
    z: -0.0712502002716
  orientation:
    x: 0.0070891474449
    y: 0.0123460302506
    z: -0.656618936166
    w: -0.754088119648
---
```

```
fa@fa-Lenovo-G400: ~
  stamp:
    secs: 1597058947
    nsecs: 177259629
  frame_id: "base_link"
orientation:
  x: 0.00719516961086
  y: 0.0122650582366
  z: -0.656889265084
  w: -0.753852963978
orientation_covariance: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
angular_velocity:
  x: -0.00153921928722
  y: -0.00226714974269
  z: 0.00219747005031
angular_velocity_covariance: [1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.218469679
1468346e-07, 0.0, 0.0, 0.0, 1.2184696791468346e-07]
linear_acceleration:
  x: -0.24516625
  y: -0.10787315
  z: 9.7870367
linear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.9999999
99999999e-08, 0.0, 0.0, 0.0, 8.999999999999999e-08]
---
```

Fig. 5. Calling Local Position & IMU Topics

In basic motion testing, 4 basic movements are carried out, namely when the robot makes a forward motion by testing the forward tilt data of the robot, the robot performs a reverse motion by testing the backward tilt of the robot, the robot performs right movements by testing the robot tilt data to the right, and the robot performs a left movement by testing the tilt data of the robot to the left. The basic movement consists of four different tables of motion, so that you can see the effect of the axis value on the robot's movement.



Color	▨ 160; 160; 164
Alpha	0,5
Plane	XY
Offset	0; 0; 0
Odometry	☑
Status: Ok	
Topic	/mavros/local_positio...
Unreliable	□
Position Tolerance	0,01
Angle Tolerance	0,01
Keep	1
Shape	Arrow
Color	▨ 85; 0; 255
Alpha	1
Shaft Length	3

Head Length
Length of the each arrow's head, in meters.

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0,01
Invert Z Axis	□
Target Fra...	<Fixed Frame>
Distance	52,8172
Focal Shap...	0,05
Focal Shap...	☑
Yaw	3,06176
Pitch	0,00479748
Focal Point	0; 0; 0

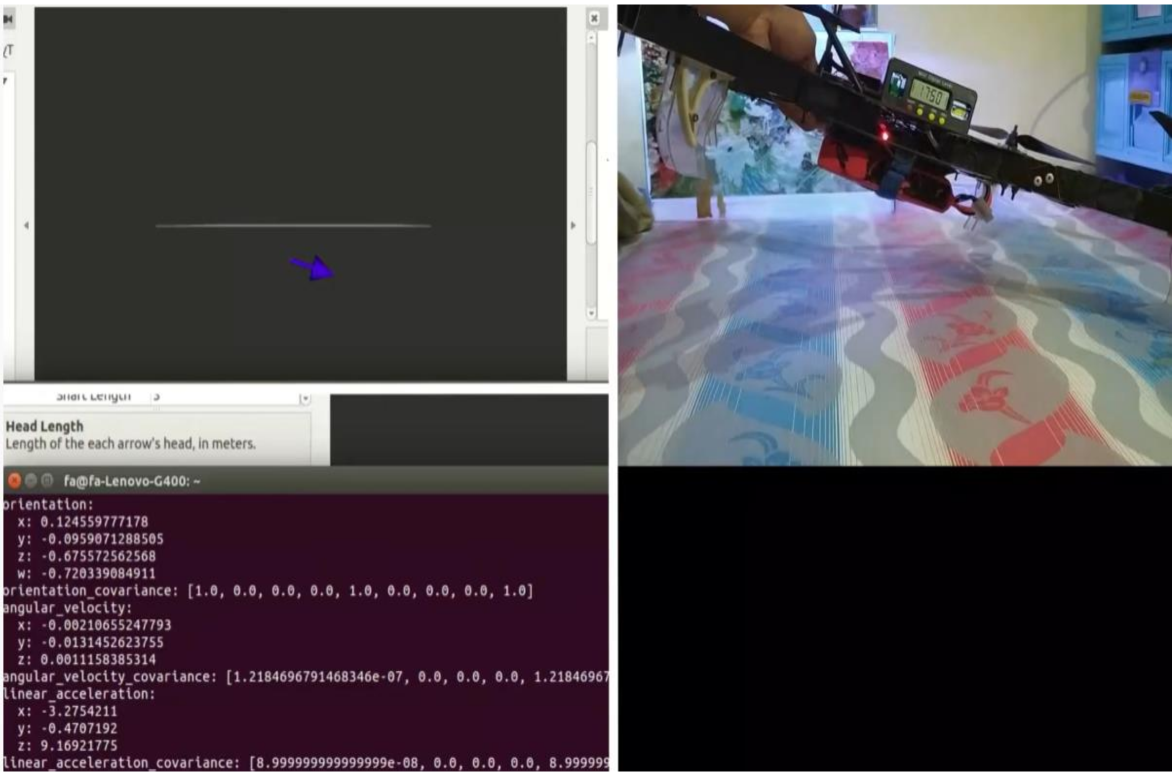
Fig. 6. Robot Modeling Display Results

4.2. Forward Robot Tilt Testing

Basic motion testing by doing forward tilt testing can be seen on the following youtube link https://www.youtube.com/watch?v=xJJ8W16Eidc. In Table 1 and Fig. 7 shows the results of the forward slope test showing that the X value in the IMU data is experiencing change in value reduction by moving away from the number 0. The Y value in the IMU data changes in value addition by approaching the number 0. The Z value in the IMU data changes in value reduction by approaching the number 0. In the forward movement the pitch value is a movement that can be done by increasing the speed at the rear and reduces the speed at the front of the propeller.

Table 1. Forward Robot Tilt Data

Data	Slope (Degree)	IMU Data		
		X	**Y**	**Z**
1	1.30	0.2353596	-0.06864655	9.81645665
2	3.20	0.9610517	-0.0588399	9.75761675
3	6.00	1.3925443	-0.0392266	9.77723005
4	11.50	2.1966896	-0.00980665	9.58109705
5	16.70	2.9616083	0.04903325	9.4928372
6	18.10	3.1969679	0.04903325	9.34573745
7	22.10	4.0403398	0.0392266	9.18883105
8	28.10	5.12887795	0.0392266	8.9828914
9	36.20	6.2174161	0.14709975	7.56092715
10	41.60	6.8058151	0.16671305	7.0215614



```
orientation:
  x: 0.124559777178
  y: -0.0959071288505
  z: -0.675572562568
  w: -0.720339084911
orientation_covariance: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
angular_velocity:
  x: -0.00210655247793
  y: -0.0131452623755
  z: 0.0011158385314
angular_velocity_covariance: [1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.21846967
linear_acceleration:
  x: -3.2754211
  y: -0.4707192
  z: 9.16921775
linear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999
```

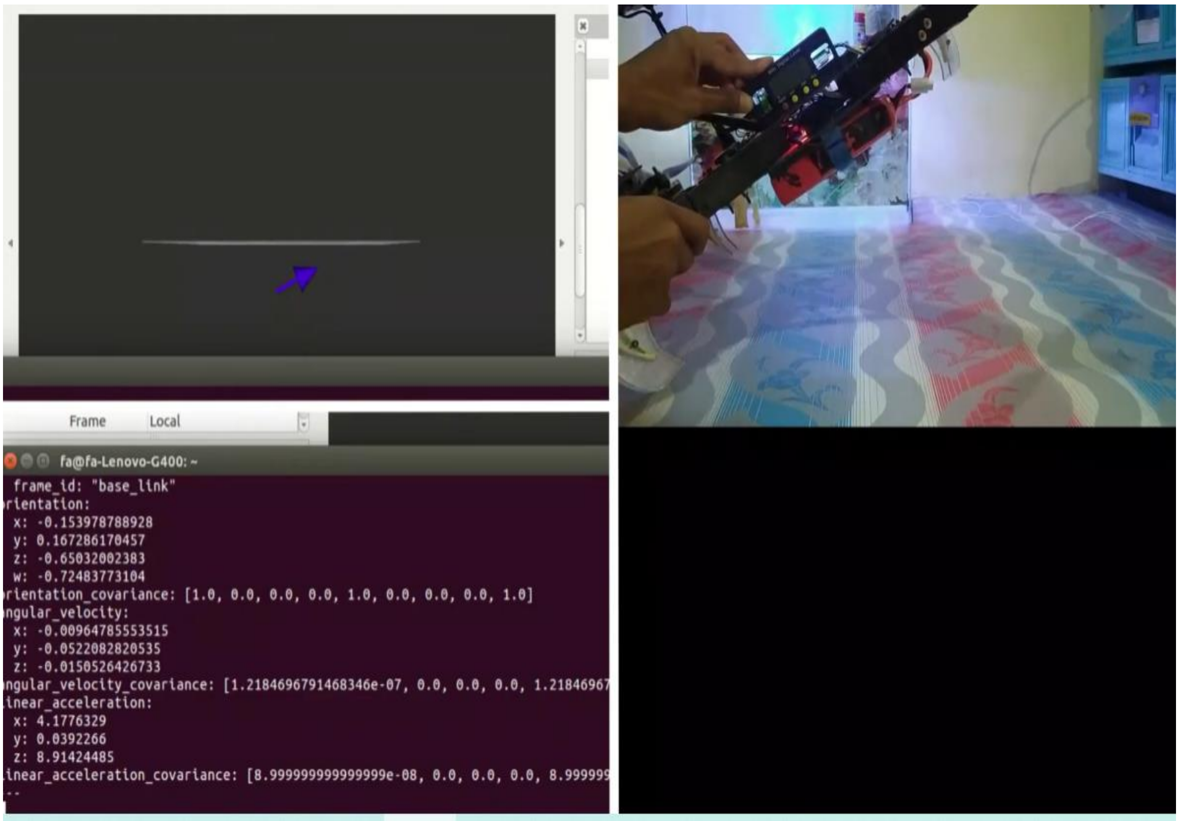
Fig. 7. CCW Pitch Data Retrieval Results

4.1. Backward Tilt Testing of Robot

Basic motion testing by doing a backward tilt test can be seen on the youtube link as https://www.youtube.com/watch?v=xJJ8W16Eidc. In Table 2 and Fig. 8, the results of the forward slope test show that the X value in the IMU data changes in value addition by moving away from the number 0. The Y value in the IMU data changes in value addition by approaching the number 0 to more. The Z value on the data imu changes the value reduction by approaching the number 0. In the backward or backward movement the pitch value is a movement that can be done by reducing the velocity on the back and increasing the speed at the front of the propeller.

Table 2. Backward Slope Robot Data

Data	Slope (Degree)	IMU Data		
		X	Y	Z
1	1.30	0.2353596	-0.06864655	9.81645665
2	3.20	0.9610517	-0.0588399	9.75761675
3	6.00	1.3925443	-0.0392266	9.77723005
4	11.50	2.1966896	-0.00980665	9.58109705
5	16.70	2.9616083	0.04903325	9.4928372
6	18.10	3.1969679	0.04903325	9.34573745
7	22.10	4.0403398	0.0392266	9.18883105
8	28.10	5.12887795	0.0392266	8.9828914
9	36.20	6.2174161	0.14709975	7.56092715
10	41.60	6.8058151	0.16671305	7.0215614



```
 frame_id: "base_link"
rientation:
 x: -0.153978788928
 y: 0.167286170457
 z: -0.65032002383
 w: -0.72483773104
rientation_covariance: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
ngular_velocity:
 x: -0.00964785553515
 y: -0.0522082820535
 z: -0.0150526426733
ngular_velocity_covariance: [1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.21846967
inear_acceleration:
 x: 4.1776329
 y: 0.0392266
 z: 8.91424485
inear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999
--
```

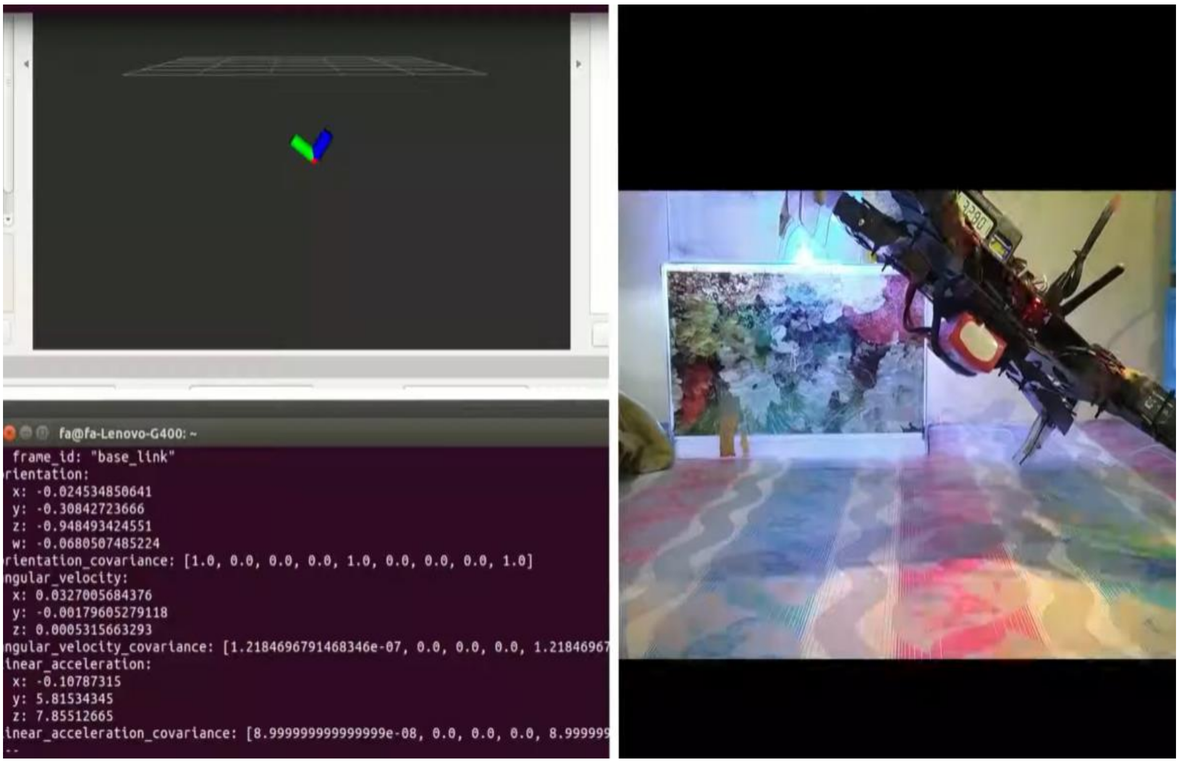
Fig. 8. Results of CW Pitch Data Retrieval

4.2. Pengujian Kemiringan Robot Ke Kanan

Basic motion testing by doing tilt testing to the right and left can be seen at the link https://www.youtube.com/watch?v=63uQA4iySsQ. In Table 3 and Fig. 9 shows the results of the tilt test to the right show that the X value in the IMU data changes in value addition by moving away from 0 the value is close to the number 0. In the movement to the right or to the left, the roll value is a movement that can be done by reducing the speed on the right side and increasing the speed on the left side of the propeller.

Table 3. Right Tilt Robot Data

Data	Slope (Degree)	IMU Data		
		X	Y	Z
1	0.10	-0.44129925	0.0392266	9.8262633
2	3.20	-0.392266	0.51975245	9.8262633
3	5.70	-0.3726527	1.18660465	9.7478101
4	8.70	-0.36284605	1.65732385	9.6889702
5	11.20	-0.3726527	1.96133	9.62032365
6	12.70	-0.36284605	2.16726965	9.6097436
7	14.10	-0.32361945	2.4908891	9.4928372
8	16.60	-0.3138128	2.8243152	9.3947707
9	18.60	-0.26477955	3.2558078	9.2770909
10	20.30	-0.28439285	3.5696206	9.1594111



```
 frame_id: "base_link"
rientation:
 x: -0.024534850641
 y: -0.30842723666
 z: -0.948493424551
 w: -0.0680507485224
rientation_covariance: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
ngular_velocity:
 x: 0.0327005684376
 y: -0.00179605279118
 z: 0.0005315663293
ngular_velocity_covariance: [1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.21846967
inear_acceleration:
 x: -0.10787315
 y: 5.81534345
 z: 7.85512665
inear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999
--
```

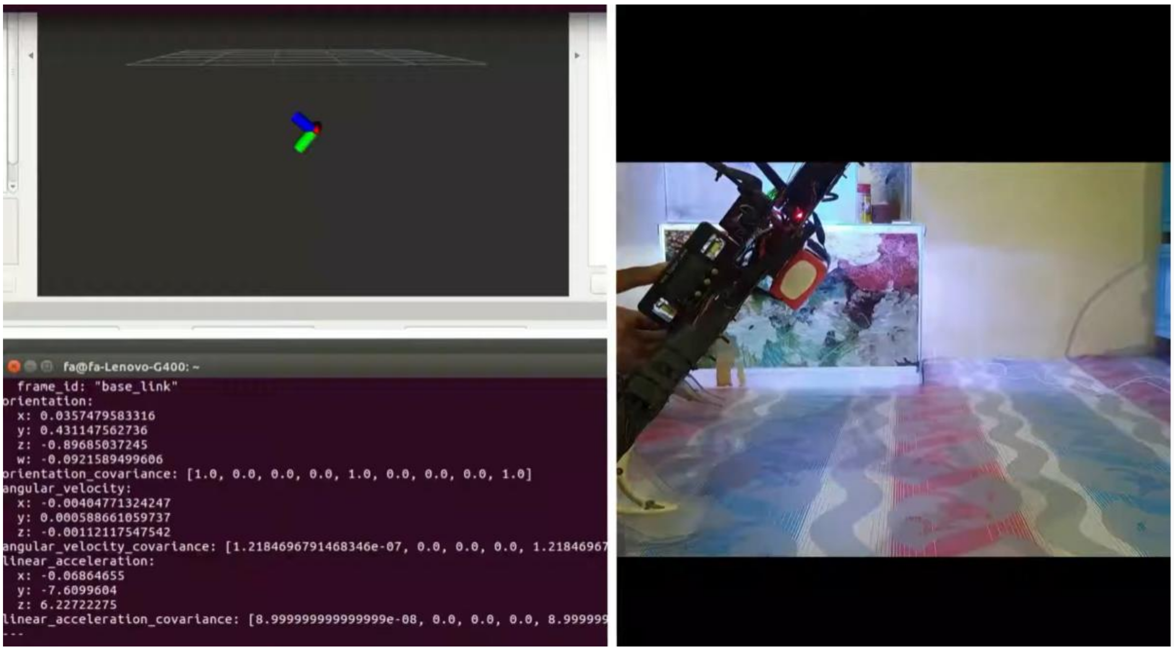
Fig. 9. CCW Roll Data Retrieval Results

4.3. Tilt Test of Robot to the Left

Basic motion testing by doing tilt testing to the right and left can be seen at the link https://www.youtube.com/watch?v=63uQA4iySsQ. In Table 4 and Fig. 10 shows the results of the tilt test to the left show that the X value in the IMU data changes in value addition by moving away from the number 0. The Y value in the IMU data changes in value addition by moving away from the number 0. The Z value on the IMU data changes in reduction the value is close to the number 0. In the movement to the right or to the left the roll value is a movement that can be done by reducing the speed on the left and increasing the speed on the right side of the propeller.

Table 4. Right Tilt Robot Data

Data	Slope (Degree)	IMU Data		
		X	Y	Z
1	1.60	-0.4314926	-0.20593965	9.87529655
2	3.10	-0.42168595	-0.50013915	9.8851032
3	4.90	-0.392266	-0.99047165	9.80665
4	7.30	-0.38245935	-1.67693715	9.6889702
5	10.70	-0.38245935	-1.96133	9.6301303
6	12.90	-0.34323275	-2.2359167	9.62032365
7	14.40	-0.34323275	-2.4320492	9.54187045
8	16.10	-0.2941995	-3.4519408	9.28689755
9	18.60	-0.20593965	-4.20705285	8.9240515
10	20.60	-0.20593965	-4.9621649	8.48275225



```
  frame_id: "base_link"
orientation:
  x: 0.0357479583316
  y: 0.431147562736
  z: -0.89685037245
  w: -0.0921589499606
orientation_covariance: [1.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0]
angular_velocity:
  x: -0.00404771324247
  y: 0.000588661059737
  z: -0.00112117547542
angular_velocity_covariance: [1.2184696791468346e-07, 0.0, 0.0, 0.0, 1.21846967
linear_acceleration:
  x: -0.06864655
  y: -7.6099604
  z: 6.22722275
linear_acceleration_covariance: [8.999999999999999e-08, 0.0, 0.0, 0.0, 8.999999
---
```

Fig. 10. Hasil Pengambilan Data *Roll* CCW

4.4. Testing Forms a Flight Pattern

Retrieval of data using Rviz in monitoring the movement of the robot using rectangular modeling, the length of each side of the square arena is five meters and a grid of twenty cells. In Fig. 11 is a view from above when the robot maneuvers that resemble a square shape, robot modeling can be seen in the red arrow, while the robot maneuver process can be seen on the link as follows: https://www.youtube.com/watch?v=jp1zN7loz_Q. Maneuvers that resemble a triangle shape, robot modeling can be seen in the green arrow in the middle, while the robot maneuver process can be seen in the following link: https://www.youtube.com/watch?v=F1ShhQDoNXE. maneuvers that resemble the "V" shape in the right image, robot modeling can be seen in the green array, while the robot maneuvering process can be seen on the following link: https://www.youtube.com/watch?v=F1ShhQDoNXE.

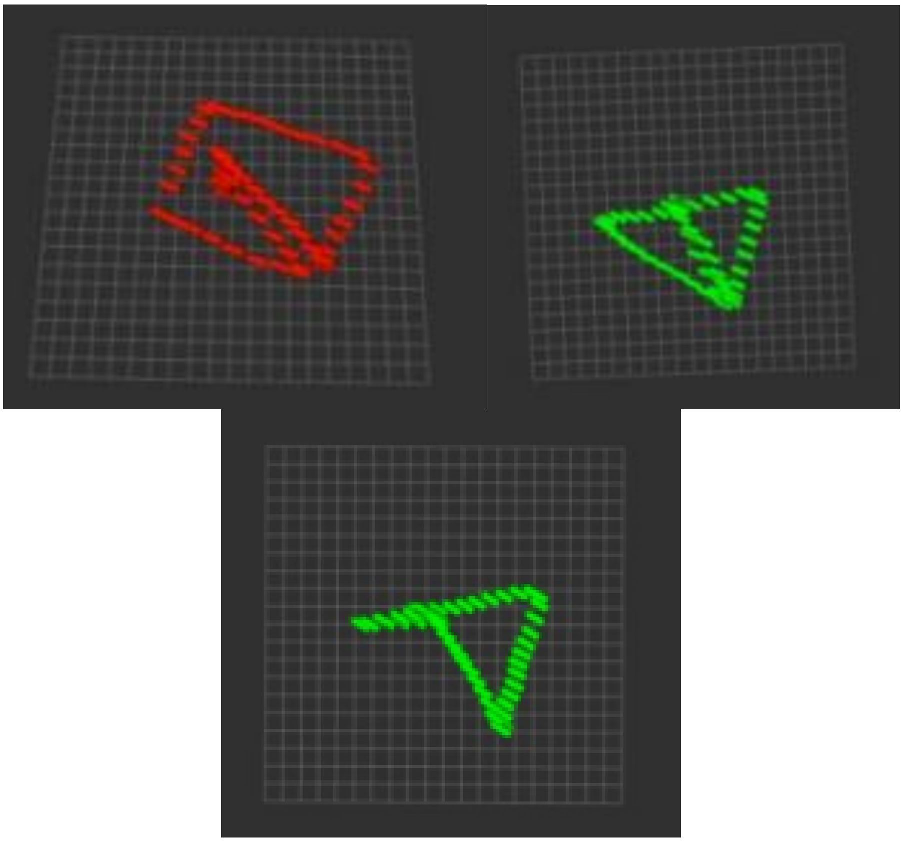


Fig. 11. Results of Square, Triangle, and Letter V Maneuver Modeling

5. Conclusion

Based on the results of research and testing of the ROS application system as navigation on quadcopter flying robots using telemetry that has been done, it can be concluded as follows. Quadcopter flying robots can implement ROS as a support for the robot in making basic movements using telemetry. Based on the tilt data to the front of the robot, it can be concluded that the X value on the IMU cube sensor changes very significantly starting from a slope of 1.10 ° to 40.10 °, experiencing a change in value from -0.3 to -6.4 compared to the Y value. and Z. Based on the slope data to the back of the robot, it can be concluded that the X value on the IMU cube sensor changes very significantly starting from a slope of 1.30 ° to 41.60 °, experiencing a change in value from 0.2 to 6.8 compared to the value Y and Z. Based on the slope data to the right of the robot, it can be concluded that the Y value on the IMU cube sensor changes very significantly starting from a slope of 0.10 ° to 20.30 °, experiencing a change in value from 0 to 3.5 compared to the X value and Z. Based on the slope data to the left of the robot, it can be concluded that the Y value on the IMU cube sensor changes very significantly starting from a slope of 1.60 ° to 20.60 °, experiencing a change in value from -0.2 to k e value -4.9 is compared to the value of X and Z. The flying robot can maneuver according to the shape on the Rviz according to the movements performed directly on the airfield.

References

[1] A. C. Subrata, "Automatic Landing And Waypoint System Berbasis Kombinasii GPS Dan Mesin Visi Untuk Multirotor Pada Kontes Robot Terbang Indonesia Divisi Vertical Take Off Landing," *Jurnal Ilmiah Teknik Elektro Komputer Dan Informatika*, vol. 2, no. 2, pp. 110-122, 2017. http://dx.doi.org/10.26555/jiteki.v2i2.4896

[2] A. Rahman, “Penerapan SLAM Gmapping dengan Robot Operating System Menggunakan Laser Scanner pada Turtlebot,” *Jurnal Rekayasa Elektrika*, vol. 16, no. 2, pp. 103–109, Aug. 2020. https://doi.org/10.17529/jre.v16i2.16491

[3] I. Iswanto, A. Ma’arif, O. Wahyunggoro, and A. Imam, “Artificial Potential Field Algorithm Implementation for Quadrotor Path Planning,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, 2019. https://doi.org/10.14569/IJACSA.2019.0100876

[4] M. Kok, J. D. Hol and T. B. Schon, “Using Inertial Sensors for Position and Orientation Estimation,” *Foundations and Trends in Signal Processing*, vol. 11, no. 1-2, pp 1-153, 2017. http://dx.doi.org/10.1561/2000000094

[5] M. E. Pittelkau, “Rotation Vector in Attitude Estimation,” *Journal of Guidance, Control and Dynamics*, vol. 26, no. 6, November-Desember 2003. https://doi.org/10.2514/2.6929

[6] J. Garcia, J. M. Molina, J. Trincado, “A Methodology for Design and Analysis of Sensor Fusion with Real Data in UAV platforms,” *Preprints*, 2018010077, 2018. https://dx.doi.org/10.20944/preprints201801.0077.v1

[7] G. Lachow, A. Khoury, J. Quach, E. Ho, I. Schroeder, J. Ren, “Autonomous Ouadcopter for Multiple Robot Tracking and Interaction in GPS-Denied Environments,” 2017 *Symposium of Aerial Robotics Competition, pp. 1-12, 2017.* http://www.aerialroboticscompetition.org/assets/downloads/2017SymposiumPapers/UniversityofCaliforniaSanDiego.pdf

[8] A. Saphala and P. I. Tanaya, “Implementation and Reconfiguration of Robot Operating System on Human Follower Transporter Robot,” *CommIT (Communication and Information Technology) Journal*, vol. 9, no. 2, p. 59, Oct. 2015. https://doi.org/10.21512/commit.v9i2.1646

[9] J. M. O’Kane, *A Gentle Introduction to ROS*, *Independently Published*, Oktober 2013. http://www.cse.sc.edu/~jokane/agitr/

[10] L. K. Amifia, M. I. Riansyah, and P. D. Putra, “Design of Logistic Transporter Robot System,” *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 6, no. 1, p. 19, Jul. 2020. https://doi.org/10.26555/jiteki.v16i1.16750

[11] S. Moon, J. J. Bird, S. Borenstein and E. W. Frew, "A Gazebo/ROS-based Communication-Realistic Simulator for Networked sUAS," *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, 2020, pp. 1819-1827. https://doi.org/10.1109/ICUAS48674.2020.9213892

[12] F. Balampanis, I. Maza, and A. Ollero, “Coastal Areas Division and Coverage with Multiple UAVs for Remote Sensing,” *Sensors*, vol. 17, no. 4, p. 808, Apr. 2017. https://doi.org/10.3390/s17040808