Robotics Studio 4: Quadrotor three – Progress Update I

Tanach Chinbutarnont ID 62340500026
Nattasit Phaisalrittiwong ID 60340500063
Pakapak Silpapinun ID 62340500042

Advisor – Pi Thanacha Choopojcharoen

I. Introduction

The quadrotor is a four-rotor Unmanned Aerial Vehicle (UAV), which contains of four rigidly attached rotors on the vehicle. Due to its unique rotor's configuration, a control scheme for quadrotor can be synthesized to maneuver with four degrees of freedom. This maneuverability helps users accomplishing desired tasks in dangerous or inaccessible environment. To gain better understanding of control scheme and its application, our team proposes a project on simulating the physics and behavior of the quadrotor.

The scope of this project consists of the following:

1. The entire system will be developed only in computer-simulated environment.
2. The goal of the quadrotor is to vary its altitude and yaw angle, while maintaining minimum change in lateral position.
3. The input from the user interface is the desired altitude and forward direction of the quadrotor.
4. The control scheme provides 3-dimentional resultant torque and force along the rotational axis of the quadrotor.
5. All physical parameters of the quadrotor are constant and known.
6. At least two sensor types are equipped on the quadrotor: 6-axis IMU (3-axis accelerometer; and 3-axis gyroscope) and range sensor.
7. The simulation result is visualized as the movement in a 3D plane; graphs of position signal and rotation signal are plotted against time.

The model of the quadrotor consists of two main parts: dynamics and kinematics model. The kinematics model equation was constructed in the body frame, while state-space representation is in the global coordinate frame. The controlling system of the quadrotor divides into three subsystems: attitude control, altitude control, and lateral flight control [1]. Sensor models are added to the model to create the closed-loop system. In the physical system, many uncertainties occur, such as model error and disturbance of surroundings. Therefore, the state estimation is required[1].

II. Quadrotor Model & Control System

To understand the essence of the quadrotor, we firstly created the quadrotor model and identified its control scheme.

2.1 Kinematic Model of Quadrotor

We utilized the knowledge of FRA333, kinematics, to find the pose and rotation matrix of the quadrotor.

- Rotation (Euler's Angle)

To control the orientation of the quadrotor, we used Z-Y-X Euler angles to model its rotation in the global frame. In this report, the global frame, G, is defined by axes $\hat{x}_G, \hat{y}_G, \hat{z}_G$ with $\hat{z}_G$ pointing upward [1]. The Body frame, B, is attached to the center of the mass of the quadrotor with $\hat{x}_B$ pointing in the quadrotor forward direction and $\hat{z}_B$ perpendicular to the plane of rotors. Therefore, the rotation matrix transforming the global frame to the body frame is given by equation 3,

$$R_G^B = \begin{bmatrix} c_\psi c_\theta - s_\phi s_\psi s_\theta & -c_\phi s_\psi & c_\psi s_\theta + c_\theta s_\phi s_\psi \\ s_\psi c_\theta - s_\phi c_\psi s_\theta & c_\phi c_\psi & s_\psi s_\theta - c_\theta s_\phi c_\psi \\ -c_\phi s_\theta & s_\phi & c_\theta c_\phi \end{bmatrix} \quad (3)$$

where ψ is yaw angle, θ is pitch angle, ϕ is roll angle, and c_ϕ and s_ϕ denote $\cos(\phi)\ and\ \sin(\phi)$.

Nevertheless, describing orientation in Z-Y-X Euler angles creates singularity issues and complexity while changing the frame reference. Therefore, besides the rotation matrix, all angular variables are in quaternion in this report.

- Orientation Kinematics (Quaternion)

From the research [1], the derivative of quaternion is given by the equation 1.

$$\frac{d}{dt}q = \begin{bmatrix} -q \cdot \omega \\ q_0\ \omega + q_v \times \omega \end{bmatrix} \tag{1}$$

The quaternion of the quadrotor can be found by integrating the derivative of quaternion as in figure 1.

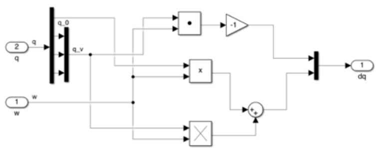

Figure 1: block orientation kinematic

Then, the integration was applied to find the quaternion.

- Position Kinematics

Likewise in angular position, the linear position can be found by integrating the input linear velocity.

2.2 *Dynamics Model of Quadrotor*

From the kinematics equation, the velocity must identify to find the pose of the quadrotor. The inputs of our quadrotor model are torque and force, respecting to global and body frame, respectively. Therefore, before applying the dynamics equation, the input force is transformed into the body frame by a rotation matrix in equation 3.

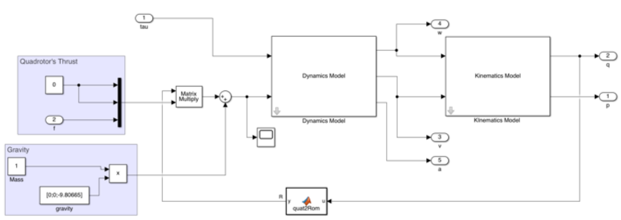

Figure 2: the overview of the quadrotor model

- Orientation Dynamics

The orientation dynamics of quadrotor can be described as equation 1.

$$I\dot{\omega} = -\omega \times I\omega + \sum_i \tau_i$$

$$\frac{d}{dt}\omega = I^{-1}(-\omega \times I\omega + \textstyle\sum_i \tau_i) \tag{2}$$

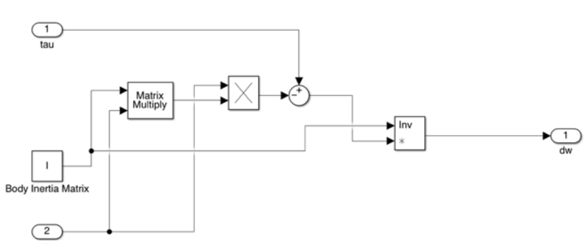

Figure 3: the Orientation Dynamics block

Then, the integration was applied to find the angular velocity.

- Position Dynamics

For the position dynamics of quadrotor, the linear acceleration, is calculated by using Newton's second Law of motion.

$$m\dot{v} = \textstyle\sum_i F_i$$

$$\frac{d}{dt}\dot{v} = \frac{(\sum_i F_i)}{m} \tag{3}$$

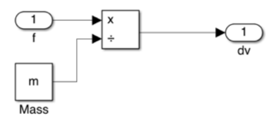

Figure 4: the Position Dynamics block

Then, the integration was applied to find the linear velocity.

2.3 *Control System*

To control the created model, applied torque and force are calculated.

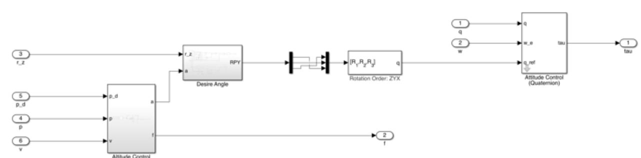

Figure 5: the overview of the hover control

- Lateral Flight

In this report, the quadrotor flight was controlled by a quintic polynomial trajectory leading to slight changes in position in the earliest stage. These insignificant changes will allow attitude control to perform synchronously with altitude control.

$$s(t) = C_0 + C_1 t + C_2 t^2 + C_3 t^3 + C_4 t^4 + C_5 t^5$$

$$\frac{d}{dt}s(t) = v(t) = C_1 + 2C_2 t + 3C_3 t^2 + 4C_4 t^3 + 5C_5 t^4$$

$$\frac{d}{dt}v(t) = a(t) = 2C_2 + 6C_3 t + 12C_4 t^2 + 20C_5 t^3$$

$$t_f = \sqrt{\frac{10(s_{final} - s_{start})}{a_{max}\ \sqrt{3}}}$$

$$C_3 = \frac{10(s_{final} - s_{start})}{t_f^3}$$

$$C_4 = -\frac{15(s_{final} - s_{start})}{t_f^4}$$

$$C_5 = \frac{6(s_{final} - s_{start})}{t_f^5}$$

where, C_0 is the initial position

C_1 is the initial velocity

C_2 is the initial acceleration

s_{final} is the desired position

s_{start} is starting position

a_{max} is the maximum acceleration

- Altitude Control

To control the applied force, we control it through acceleration. PID controller was applied to control the desired acceleration from the position error, $\Delta\vec{p} = \vec{p}_d - \vec{p}$.

$$a_d = K_p\Delta\vec{p} + K_i \int \Delta\vec{p}\, dt + K_d(-\vec{v}) \quad (4)$$

where, $\vec{p}_d$ is a desired position vector.

$\vec{p}$ is a current position vector.

$\vec{v}$ is a current velocity vector.

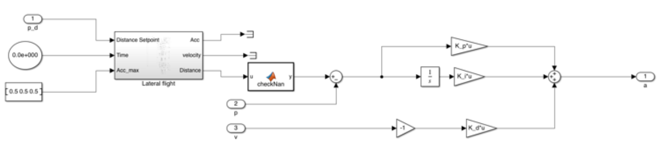

Figure 6: the overview of Altitude Control block

- Attitude Control

The result of attitude control is desired torque with the orientation as the input. According to the project's scope, yaw value has already inputted through the desired forward direction. Linearization applies to equation 5 to find roll and pitch.

$$m\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = R_G^B \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (5)$$

Linearization

$$a_x = (c_\psi\theta + s_\psi\phi)g$$

$$a_y = (s_\psi\theta - c_\psi\phi)g$$

$$\phi = \frac{a_x}{g}s_\psi - \frac{a_y}{g}c_\psi$$

$$\theta = \frac{a_x}{g}c_\psi + \frac{a_y}{g}s_\psi$$

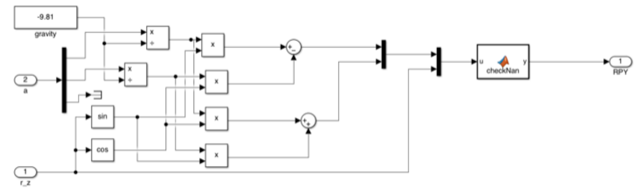

Figure 7: the overview of Desire Angle block

The quaternion-based attitude control is used, where Δq and $\Delta\omega$ are the quaternion error and angular velocity error, respectively.

$$\Delta q = q_d \otimes \bar{q}$$

$$\Delta q = \begin{bmatrix} q_{e,0} \\ q_{e,v} \end{bmatrix}$$

$$\tau^B = K_q \times q_{e,v} + K_\omega(\Delta\omega) \quad (6)$$

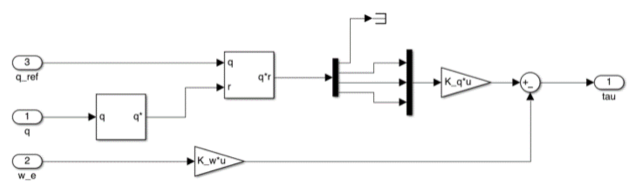

Figure 8: the overview of Attitude Control Block

To observe the result, we connect all blocks and visualize the movement in a 3D plane via MATLAB.

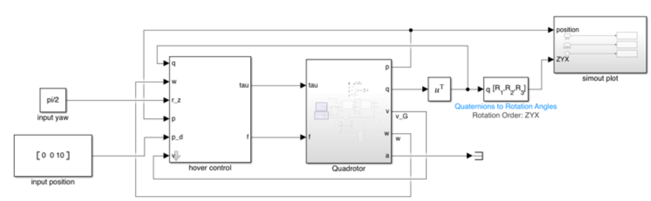

Figure 9: quadrotor model

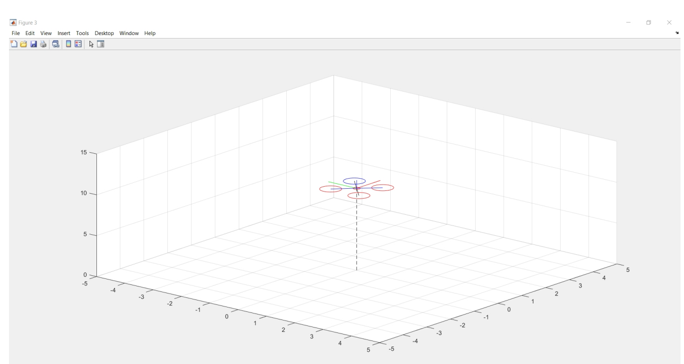

Figure 10: the visualization of quadrotor movement in a 3D plane

III. Sensor Model & State-Space Estimation

In the real physical system, all dependent variables are inputted from sensor, which leads to the occurrence of uncertainties.

3.1 Sensor Model

In this report, two sensor types are equipped on the quadrotor: 6-axis IMU (3-axis accelerometer; and 3-axis gyroscope) and range sensor (ultrasonic).

- Ultrasonic Sensor

Due to the possibilities of rotation in roll and pitch, the sensor equation can be described as follow,

$$\vec{p} = R_y R_x \vec{p}_u$$

$$R_y R_x = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\emptyset \\ 0 & s_\phi & c_\emptyset \end{bmatrix}$$

$$\vec{p} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} c_\theta & s_\theta s_\emptyset & c_\emptyset s_\theta \\ 0 & c_\emptyset & -s_\emptyset \\ -s_\theta & s_\emptyset c_\theta & c_\emptyset c_\theta \end{bmatrix}\vec{p}_u$$

Ultrasonic measures the length along its z axis. Therefore, the ultrasonic measured distance, Z_u, can be calculated as in equation 7.

$$\vec{p} = \begin{bmatrix} c_\theta & s_\theta s_\emptyset & c_\emptyset s_\theta \\ 0 & c_\emptyset & -s_\emptyset \\ -s_\theta & s_\emptyset c_\theta & c_\emptyset c_\theta \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ Z_u \end{bmatrix}$$

$$\vec{p} = \begin{bmatrix} c_\phi s_\theta Z_u \\ -s_\emptyset Z_u \\ c_\theta c_\emptyset Z_u \end{bmatrix}$$

$$Z_u = \frac{p_z}{c_\theta c_\emptyset} \tag{7}$$

where $\vec{p}$ is the actual position vector along z axis $\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$

$\vec{p}_u$ is a measured position vector $\begin{bmatrix} 0 \\ 0 \\ Z_u \end{bmatrix}$

Z_u is a measured position vector along z axis

- Accelerometer

The accelerometer is equipped on the quadrotor, such that the measured acceleration, $\vec{a}_a$, is calculated in the body frame as in the following equation.

$$\vec{a}_a = R_G^B\left(\vec{a} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}\right) \tag{8}$$

To find the quadrotor acceleration, $\vec{a}$, the Newton's second laws is used.

$$\Sigma F = m\vec{a}$$

$$R_G^{B^T} F^B - m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = m\vec{a}$$

$$\vec{a} = \frac{1}{m}\left(R_G^{B^T}\vec{F}^B - m\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}\right)$$

Therefore,

$$\vec{a}_a = R_G^B\left(\frac{1}{m}\left(R_G^{B^T}\vec{F}^B + m\begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}\right) - \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}\right)$$

$$\vec{a}_a = \begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix} = \frac{1}{m}F^B = \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{9}$$

where, $\vec{a}_a$ is measured acceleration $\begin{bmatrix} X_a \\ Y_a \\ Z_a \end{bmatrix}$

$\vec{a}$ is a quadrotor generated acceleration vector

$\vec{F}^B$ is a propellers' generated force $\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}$

T is a propellers' generated thrust

Z_a is a measured acceleration along z axis

- Gyroscope Sensor

Due to the quadrotor model, the torque will only apply around z axis, therefore the measured angular velocity around z axis, Z_g, is quadrotor angular velocity around z axis, ω.

Thus,

$$\begin{bmatrix} Z_u \\ Z_a \\ Z_g \end{bmatrix} = \begin{bmatrix} \frac{\vec{P}_z}{c_\theta c_\emptyset} \\ \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \\ \omega \end{bmatrix} \tag{10}$$

3.2 Estimator

The estimator constructs with 2 main parts: position estimator and orientation estimator. In position estimation, the Kalman Filter is used to estimate the quadrotor's position, while the Extended Kalman Filter is used to estimate the orientation.

- Position Estimator

The estimation of position, velocity, and acceleration equation is in the discrete-time domain. The Kalman Filter can be described in equation 11.

$$\vec{x}[k+1] = A\vec{x}[k] + B\vec{u}[k] + G\vec{w}[k]$$

$$\vec{y}[k] = C\vec{x}[k] + D\vec{u}[k] + \vec{V}[k] \quad (11)$$

where, $\vec{x}$ is a state space of the quadrotor

$\vec{y}$ is a input of the quadrotor

$\vec{w}$ is a process noise of the quadrotor

$\vec{v}$ is a measurement noise of the quadrotor

A, B, C and D are parameters

- Orientation Estimator

- Euler's rate $(\dot{r})$

find the angular velocity of each axis, $\omega = \dot{Z} + \dot{Y} + \dot{X}$

$$\dot{Z} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & s_\theta \\ 0 & -s_\theta & c_\theta \end{bmatrix} \begin{bmatrix} c_\phi & 0 & -s_\phi \\ 0 & 1 & 0 \\ s_\phi & 0 & c_\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{r}_z \end{bmatrix}$$

$$\dot{Y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\theta & s_\theta \\ 0 & -s_\theta & c_\theta \end{bmatrix} \begin{bmatrix} 0 \\ \dot{r}_y \\ 0 \end{bmatrix}$$

$$\dot{X} = \begin{bmatrix} \dot{r}_x \\ 0 \\ 0 \end{bmatrix}$$

$$\omega = \begin{bmatrix} 1 & 0 & -s_\theta \\ 0 & c_\theta & s_\theta c_\phi \\ 0 & -s_\theta & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} \dot{r}_x \\ \dot{r}_y \\ \dot{r}_z \end{bmatrix}$$

$$\dot{\vec{r}} = \begin{bmatrix} 1 & s_\theta t_\phi & c_\theta t_\phi \\ 0 & c_\theta & -s_\theta \\ 0 & \frac{s_\theta}{c_\phi} & \frac{c_\theta}{c_\phi} \end{bmatrix} \omega$$

From $\dot{\vec{r}} = J_r(r)\omega$

$$J_r(r) = \begin{bmatrix} 1 & s_\theta t_\phi & c_\theta t_\phi \\ 0 & c_\theta & -s_\theta \\ 0 & \frac{s_\theta}{c_\phi} & \frac{c_\theta}{c_\phi} \end{bmatrix}$$

The state space of the quadrotor orientation can be written as these following equations.

$$\vec{\omega}[k+1] = \vec{\omega}[k] + \vec{\alpha}[k]\Delta t + \frac{1}{2}\dot{\vec{\alpha}}[k](\Delta t^2)$$

$$\vec{\alpha}[k+1] = \vec{\alpha}[k] + \dot{\vec{\alpha}}[k]\Delta t$$

$$\vec{r}[k+1] = J_r(\vec{r}[k])\left(\vec{\omega}[k] + \vec{\alpha}[k]\Delta t + \frac{1}{2}\dot{\vec{\alpha}}[k]\Delta t^2\right)\Delta t + \vec{r}[k]$$

From those equations, they are a non-linear equation. Therefore, the Extended Kalman Filter is used to estimate the quadrotor orientation

3.3 *Implementation*

- Position Estimation

In the model, we create the Kalman filter by inseting the Kalman filter block with the calculated parameters and set the sample rate to 100 Hz. To test the performance of the Kalman filter, the noise is added into the system.

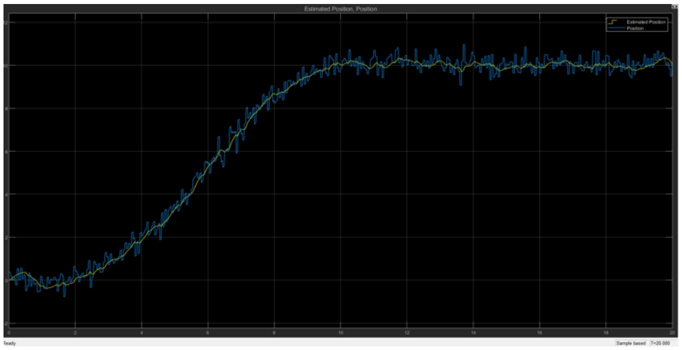

Figure 11: the graph of position along z axis after adding noise and position estimator into the system.

- Orientation Estimation

The orientation Estimation is in the validating process. Therefore, the Orientation Estimation block is not yet created.

References

[1] T. Choopojcharoen, "IMPLEMENTATION OF CONTROL & ESTIMATION OF QUADROTOR IN MATLAB", 2016.

Appendix A: Position State-Space Identification

From $\vec{a} = \frac{1}{m} R_B^G \begin{bmatrix} 0 \\ 0 \\ T+d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$

$$\vec{a}[k] = \frac{1}{m} R(\vec{r}[\mathrm{k}]) \begin{bmatrix} 0 \\ 0 \\ T+d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$$

where, $\vec{a}$ is the acceleration

$\vec{a}[k]$ is the discrete-time acceleration

m is mass

R_B^G is the rotation matrix from Body-frame to Global-frame

$R(\vec{r}[\mathrm{k}])$ is the discrete-time rotation matrix from Body-frame to Global-frame

T is thrust

g is gravity

Because of $\frac{d\vec{v}}{dt} = \vec{a}$

$$\int_{\tau=0}^{t} \left(\frac{d\vec{v}}{dt}\right) d\tau = \int_{\tau=0}^{t} (\vec{a}) \, d\tau$$

$$\vec{v}(t) = \vec{v}(0) + \vec{a}(t)$$

at $t = t_k$

$$\vec{v}(t_k) = \vec{v}(0) + \vec{a}(t_k)$$

$$\vec{v}[k] = \vec{v}[0] + \vec{a}[k](t_k)$$

at $t_{k+1} = t_k + \Delta t$

$$\vec{v}(t_{k+1}) = \vec{v}(0) + \vec{a}(t_{k+1})$$

$$\vec{v}[k+1] = \vec{v}[0] + \vec{a}[k](t_{k+1})$$

$$\vec{v}[k+1] = \vec{v}[0] + \vec{a}[k](t_k + \Delta t)$$

$$\vec{v}[k+1] = \vec{v}[k] + \vec{a}[k]\Delta t \tag{1}$$

Replace $\vec{a}[k] = \frac{1}{m} R(\vec{r}[\mathrm{k}]) \begin{bmatrix} 0 \\ 0 \\ T+d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix}$ in equation (1)

$$\vec{v}[k+1] = \vec{v}[k] + \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \begin{bmatrix} 0 \\ 0 \\ T+d \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \Delta t$$

Because of $\frac{d\vec{p}}{dt} = \vec{v}$

$$\int_{\tau=0}^{t}\left(\frac{d\vec{p}}{dt}\right)d\tau = \int_{\tau=0}^{t}(\vec{v}\,)\,d\tau$$

$$= \int_{\tau=0}^{t}(\vec{v}(0) + \vec{a}(\mathrm{t})\,)\,d\,\tau$$

$$= \int_{\tau=0}^{t}(\vec{v}(0))d\tau + \int_{\tau=0}^{t}(\vec{a}(\mathrm{t})\,)\,d\tau$$

$$p(t) = \vec{p}(0) + \vec{v}(0)t + \frac{\vec{a}t^2}{2}$$

at $t = t_k$

$$\vec{\mathrm{p}}(\mathrm{t_k}) = \vec{\mathrm{p}}(0) + \vec{\mathrm{v}}(0)\mathrm{t_k} + \frac{\vec{\mathrm{a}}\mathrm{t_k}^2}{2}$$

$$\vec{\mathrm{p}}[\mathrm{k}] = \vec{\mathrm{p}}[0] + \vec{\mathrm{v}}[0]\mathrm{t_k} + \frac{\vec{\mathrm{a}}[\mathrm{k}]\mathrm{t_k}^2}{2}$$

at $t_{k+1} = t_k + \Delta t$

$$\vec{p}(t_{k+1}) = \vec{p}[k+1]$$

$$\vec{p}[k+1] = \vec{p}[0] + \vec{v}[0]t_k + \vec{v}[0]\Delta t + \frac{\vec{a}[k]t_k^2}{2} + \frac{\vec{a}[k]\,\Delta t^2}{2} + \vec{a}[k]t_k\Delta t$$

$$= \vec{p}[k] + \Delta t(\vec{v}[0] + \vec{a}[k]t_k) + \frac{\vec{a}[k]\,\Delta t^2}{2}$$

$$\vec{p}[k+1] = \vec{p}[k] + \vec{v}[k]\Delta t + \frac{\vec{a}[k]\,\Delta t^2}{2} \tag{2}$$

Replace $\vec{a}[k] = \frac{1}{m}R(\vec{r}[\mathrm{k}]\,)\begin{bmatrix}0\\0\\T+d\end{bmatrix} + \begin{bmatrix}0\\0\\-g\end{bmatrix}$ in equation (2)

$$\vec{p}[k+1] = \vec{p}[k] + \vec{v}[k]\Delta t + \frac{\Delta t^2}{2}\left(\frac{1}{m}R(\vec{r}[\mathrm{k}])\begin{bmatrix}0\\0\\T+d\end{bmatrix} + \begin{bmatrix}0\\0\\-g\end{bmatrix}\right)$$

$$= \vec{p}[k] + \vec{v}[k]\Delta t + \begin{bmatrix}0\\0\\-g\frac{\Delta t^2}{2}\end{bmatrix} + \frac{\Delta t^2}{2m}R(\vec{r}[\mathrm{k}])\begin{bmatrix}0\\0\\T+d\end{bmatrix}$$

Therefore

$$\vec{p}[k+1] = \vec{p}[k] + \vec{v}[k]\Delta t + \begin{bmatrix}0\\0\\-g\frac{\Delta t^2}{2}\end{bmatrix} + \frac{\Delta t^2}{2m}R(\vec{r}[\mathrm{k}])\begin{bmatrix}0\\0\\T\end{bmatrix} + \frac{\Delta t^2}{2m}R(\vec{r}[\mathrm{k}]\,)\begin{bmatrix}0\\0\\d\end{bmatrix}$$

$$\vec{v}[k+1] = \vec{v}[k] + \frac{\Delta t}{m}R(\vec{r}[\mathrm{k}]\,)\begin{bmatrix}0\\0\\T\end{bmatrix} + \frac{\Delta t}{m}R(\vec{r}[\mathrm{k}]\,)\begin{bmatrix}0\\0\\d\end{bmatrix} + \begin{bmatrix}0\\0\\-g\Delta t\end{bmatrix}$$

$$
\begin{bmatrix} \overrightarrow{p_x}[k+1] \\ \overrightarrow{p_y}[k+1] \\ \overrightarrow{p_z}[k+1] \\ \overrightarrow{v_x}[k+1] \\ \overrightarrow{v_y}[k+1] \\ \overrightarrow{v_z}[k+1] \\ \overrightarrow{a_x}[k+1] \\ \overrightarrow{a_y}[k+1] \\ \overrightarrow{a_z}[k+1] \end{bmatrix} = \begin{bmatrix} I_3 & \Delta t \cdot I_3 & 0_{3x3} \\ 0_{3x3} & I_3 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix} T + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix} d + \begin{bmatrix} 0 \\ 0 \\ \frac{-\Delta t^2}{2} \\ 0 \\ 0 \\ -\Delta t \\ 0 \\ 0 \\ -1 \end{bmatrix} \vec{g}
$$

Because of position estimator interest just 2 sensors which are ultrasonic sensor and accelerometer (Z_u, Z_a), respectively. The equation can be rewritten it in form of state

From
$$
\begin{bmatrix} Z_a \\ Z_u \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ R_B^G \vec{r}[\mathrm{k}]^T (\overrightarrow{a_z}[k] - \vec{g}) \\ \frac{\vec{P}_z}{c_\theta c_\emptyset} \end{bmatrix}
$$

Rearrange them in form of state variables

$$
\begin{bmatrix} Z_a \\ Z_u \end{bmatrix} = \begin{bmatrix} 0_{3x3} & 0_{3x3} & R_B^G(\vec{r}[\mathrm{k}])^T \\ 0\ 0\ \frac{1}{c_\theta c_\emptyset} & 0\ 0\ 0 & 0\ 0\ 0 \end{bmatrix} \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T \vec{g} \\ 0 \end{bmatrix}
$$

where, $\overrightarrow{p_\iota}[k+1]$ is the next step of i-axis position where i = x, y, z

$\overrightarrow{v_\iota}[k+1]$ is the next step of i-axis velocity where i = x, y, z

$\overrightarrow{a_\iota}[k+1]$ is the next step of i-axis acceleration where i = x, y, z

$\overrightarrow{p_\iota}[k]$ is the discrete-time of i-axis position where i = x, y, z

$\overrightarrow{v_\iota}[k]$ is the discrete-time of i-axis velocity where i = x, y, z

$\overrightarrow{a_\iota}[k]$ is the discrete-time of i-axis acceleration where i = x, y, z

Appendix B: Estimation of Kalman Filter's Parameter

Rearrange the equation of state and output equation in form of Kalman filter

From $\vec{x}[k+1] = A\vec{x}[k] + B\vec{u}[k] + G\vec{w}[k]$ (11)

$$\vec{y}[k] = C\vec{x}[k] + D\vec{u}[k] + \vec{V}[k] \quad (12)$$

Define $\vec{x}[k] = \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix}$ and $\vec{u}[k] = \begin{bmatrix} u_1 \\ \\ u_2 \end{bmatrix}$

From Output equation, which is sensor equation from Appendix A, it cannot rearrange into the equation which its variables are thrust and gravity. Because sensor equation doesn't have thrust variables, we must combine them in matrix form, $\vec{u}[k] = \begin{bmatrix} u_1 \\ \\ u_2 \end{bmatrix}$

Where u_1 is

$$\begin{bmatrix} \overrightarrow{p_x}[k+1] \\ \overrightarrow{p_y}[k+1] \\ \overrightarrow{p_z}[k+1] \\ \overrightarrow{v_x}[k+1] \\ \overrightarrow{v_y}[k+1] \\ \overrightarrow{v_z}[k+1] \\ \overrightarrow{a_x}[k+1] \\ \overrightarrow{a_y}[k+1] \\ \overrightarrow{a_z}[k+1] \end{bmatrix} = \begin{bmatrix} I_3 & \Delta t \cdot I_3 & 0_{3x3} \\ 0_{3x3} & I_3 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix} T + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix} d + \begin{bmatrix} 0 \\ 0 \\ \frac{-\Delta t^2}{2} \\ 0 \\ 0 \\ -\Delta t \\ 0 \\ 0 \\ -1 \end{bmatrix} \vec{g}$$

Rearrange the above equation

$$\begin{bmatrix} \overrightarrow{p_x}[k+1] \\ \overrightarrow{p_y}[k+1] \\ \overrightarrow{p_z}[k+1] \\ \overrightarrow{v_x}[k+1] \\ \overrightarrow{v_y}[k+1] \\ \overrightarrow{v_z}[k+1] \\ \overrightarrow{a_x}[k+1] \\ \overrightarrow{a_y}[k+1] \\ \overrightarrow{a_z}[k+1] \end{bmatrix} = \begin{bmatrix} I_3 & \Delta t \cdot I_3 & 0_{3x3} \\ 0_{3x3} & I_3 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix} \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix} d + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2} \\ 0 \\ 0 \\ \Delta t \\ 0 \\ 0 \\ 1 \end{bmatrix} (\frac{1}{m} R(\vec{r}[\mathrm{k}]T - \vec{g})$$

Then, substitute $\begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2} \\ 0 \\ 0 \\ \Delta t \\ 0 \\ 0 \\ 1 \end{bmatrix} (\frac{1}{m} R(\vec{r}[\mathrm{k}]T - \vec{g}) = u_1$

And u_2 can be calculate from

$$\begin{bmatrix} Z_a \\ Z_u \end{bmatrix} = \begin{bmatrix} 0_{3x3} & 0_{3x3} & R_B^G(\vec{r}[\mathrm{k}])^T \\ 0\ 0\ \frac{1}{\cos\theta\cos\emptyset} & 0\ 0\ 0 & 0\ 0\ 0 \end{bmatrix} \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T \vec{g} \\ 0 \end{bmatrix}$$

Because of gravity is one of inputs of this system

Therefore, $\begin{bmatrix} 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T \vec{g} \\ 0 \end{bmatrix} = u_2$

$$\vec{u}[k] = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{\Delta t}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{1}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T \vec{g} \\ 0 \end{bmatrix}$$

All Kalman filter variables are ready to write in their form

$$\vec{x}[k+1] = \begin{bmatrix} I_3 & \Delta t \cdot I_3 & 0_{3x3} \\ 0_{3x3} & I_3 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix} \vec{x}[k] + \begin{bmatrix} I_9 & 0_{9x4} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{\Delta t}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{1}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T \vec{g} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix} d$$

$$\vec{y}[k] = \begin{bmatrix} 0_{3x3} & 0_{3x3} & R_B^G(\vec{r}[\mathrm{k}])^T \\ 0\ 0\ \frac{1}{c_\theta c_\emptyset} & 0\ 0\ 0 & 0\ 0\ 0 \end{bmatrix} \vec{x}[k] + \begin{bmatrix} 0_{4x9} & I_4 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{\Delta t}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{1}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T \vec{g} \\ 0 \end{bmatrix}$$

Therefore

A in the equation 11 is $\begin{bmatrix} I_3 & \Delta t \cdot I_3 & 0_{3x3} \\ 0_{3x3} & I_3 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix}$

B in the equation 11 is $\begin{bmatrix} I_9 & 0_{9x4} \end{bmatrix}$

C in the equation 11 is $\begin{bmatrix} 0_{3x3} & 0_{3x3} & R_B^G(\vec{r}[\mathrm{k}])^T \\ 0\ 0\ \frac{1}{\cos\theta\cos\emptyset} & 0\ 0\ 0 & 0\ 0\ 0 \end{bmatrix}$

D in the equation 11 is $\begin{bmatrix} 0_{4x9} & I_4 \end{bmatrix}$

G in the equation 12 is $\begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{\Delta t}{m} R(\vec{r}[\mathrm{k}]) \\ 0 \\ 0 \\ \frac{1}{m} R(\vec{r}[\mathrm{k}]) \end{bmatrix}$

where, $\vec{x}[k] = \begin{bmatrix} \overrightarrow{p_x}[k] \\ \overrightarrow{p_y}[k] \\ \overrightarrow{p_z}[k] \\ \overrightarrow{v_x}[k] \\ \overrightarrow{v_y}[k] \\ \overrightarrow{v_z}[k] \\ \overrightarrow{a_x}[k] \\ \overrightarrow{a_y}[k] \\ \overrightarrow{a_z}[k] \end{bmatrix}$ and $\vec{u}[k] = \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta t^2}{2m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{\Delta t}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ \frac{1}{m}(R_B^G(\vec{r}[\mathrm{k}])T - \vec{g}) \\ 0 \\ 0 \\ -R_B^G(\vec{r}[\mathrm{k}])^T\vec{g} \\ 0 \end{bmatrix}$

Appendix C: Orientation State-Space Identification

Prove the equation for Orientation

From $\dot{\vec{r}} = \begin{bmatrix} 1 & s_\theta t_\phi & c_\theta t_\phi \\ 0 & c_\theta & -s_\theta \\ 0 & \frac{s_\theta}{c_\phi} & \frac{c_\theta}{c_\phi} \end{bmatrix} \omega$

Define that $J_r(r) = \begin{bmatrix} 1 & s_\theta t_\phi & c_\theta t_\phi \\ 0 & c_\theta & -s_\theta \\ 0 & \frac{s_\theta}{c_\phi} & \frac{c_\theta}{c_\phi} \end{bmatrix}$ and $\dot{\vec{r}} = J_r(r)\omega$

Re-write in term of discrete time

$$\frac{\vec{r}[k+1] - \vec{r}[k]}{\Delta t} = J_r(\vec{r}[k])\overline{\omega}[k]$$

$$\vec{r}[k+1] = J_r(\vec{r}[k])\vec{\omega}[k]\Delta t + \vec{r}[k]$$

Assume that is $\dot{\vec{\alpha}}$ (rate of change of Angular acceleration) is a gaussian distribution with 0 mean

$$\int_{\tau=0}^{t} (\frac{d\vec{\alpha}}{dt})\, d\tau \quad = \int_{\tau=0}^{t} (\dot{\vec{\alpha}}\,)\, d\tau$$

$$\vec{\alpha}(t) - \vec{\alpha}(0) \quad = \dot{\vec{\alpha}}\, t$$

$$\vec{\alpha}(t) \quad = \vec{\alpha}(0) + \dot{\vec{\alpha}}\, t$$

at $t = t_k$

$$\vec{\alpha}(t_k) = \vec{\alpha}[k] = \vec{\alpha}\,[0] + \dot{\vec{\alpha}}[k](t_k)$$

at $t_{k+1} = t_k + \Delta t$;

$$\vec{\alpha}(t_{k+1}) = \vec{\alpha}(0) + \dot{\vec{\alpha}}(t_k) + \dot{\vec{\alpha}}\,\Delta\, t$$

$$\vec{\alpha}[k+1] = \vec{\alpha}\,[k] + \dot{\vec{\alpha}}[k](\Delta\, t)$$

$$\int_{\tau=0}^{t} (\frac{d\vec{\omega}}{dt})\, d\tau = \int_{\tau=0}^{t} (\alpha\,)\, d\tau$$

$$\vec{\omega}(t) - \vec{\omega}(0) = \int_{\tau=0}^{t} \big(\vec{\alpha}(0)\big) d\tau + \int_{\tau=0}^{t} \big(\dot{\vec{\alpha}} t\big) d\tau$$

$$\vec{\omega}(t) - \vec{\omega}(0) = \vec{\alpha}(0)t + \frac{1}{2}\dot{\vec{\alpha}} t^2$$

$$\vec{\omega}(t) = \vec{\omega}(0) + \vec{\alpha}(0)t + \tfrac{1}{2}\dot{\vec{\alpha}} t^2$$

at $t = t_k$

$$\vec{\omega}(t_k) = \vec{\omega}[k] = \vec{\omega}[0] + \vec{\alpha}[0]t + \tfrac{1}{2}\dot{\vec{\alpha}}[k]t^2$$

at $t_{k+1} = t_k + \Delta t$

$$\vec{\omega}(t_k+\Delta t)=\vec{\omega}[k+1]=\vec{\omega}[0]+\vec{\alpha}[0]t+\vec{\alpha}[0]\Delta\, t+\frac{1}{2}\dot{\vec{\alpha}}[k](t^2+2t\Delta t+\Delta t^2)$$

$$\vec{\omega}[k+1]=\ \vec{\omega}[k]+\vec{\alpha}[0]\Delta t+\ \dot{\vec{\alpha}}[k]t\Delta t+\tfrac{1}{2}\dot{\vec{\alpha}}[k](\Delta t^2)$$

$$=\ \vec{\omega}[k]+\big(\vec{\alpha}[0]\dot{\vec{\alpha}}[k]t\big)\Delta t+\tfrac{1}{2}\dot{\vec{\alpha}}[k](\Delta t^2)=\vec{\omega}[k]+\vec{\alpha}[k]\Delta t+\tfrac{1}{2}\dot{\vec{\alpha}}[k]\Delta t^2$$

Therefore,

$$\vec{\omega}[k+1]=\ \vec{\omega}[k]+\vec{\alpha}[k]\Delta t+\frac{1}{2}\dot{\vec{\alpha}}[k](\Delta t^2)$$

$$\vec{\alpha}[k+1]=\vec{\alpha}[k]+\dot{\vec{\alpha}}[k]\cdot\Delta t$$

$$\vec{r}[k+1]=J_r(\vec{r}[k])\left(\vec{\omega}[k]+\ \vec{\alpha}[k]\cdot\Delta t+\frac{1}{2}\dot{\vec{\alpha}}[k]\Delta t^2\right)\cdot\Delta t+\vec{r}[k]$$

$$=\vec{r}[k]+J_r(\vec{r}[k])\vec{\alpha}[k]\cdot\Delta t+J_r(\vec{r}[k])\vec{\alpha}[k]\Delta t^2+\frac{1}{2}J_r(\vec{r}[k])\dot{\vec{\alpha}}[k]\Delta t^3$$

where, $\vec{\omega}[k+1]$ is the next step of Angular velocity

$\vec{\alpha}[k+1]$ is the next step of Angular acceleration

$\vec{r}[k+1]$ is the next step of Euler's orientation

$\vec{\omega}[k]$ is the discrete time of Angular velocity

$\vec{\alpha}[k]$ is the discrete time of Angular acceleration

$\dot{\vec{\alpha}}[k]$ is the discrete time of rate of change of Angular acceleration

$\vec{r}[k]$ is the discrete time of Euler's orientation

J_r is the Jacobian of orientation and its rate of change

The result of equation will lead to do Extended Kalman Filter because $\vec{r}[k+1]$ equation is non-linear type

Planning Table

	TASK	Description	RESPONSIBLE	START	FINISH	STATUS
1	Planning and Listing all tasks		everyone	24/Aug/21	30/Aug/21	100%
2	Modelling					
	Controller's Model	Altitude Control	Pakapak	31/Aug/21	7/Sep/21	100%
		Attitude Control				100%
		Lateral Control				100%
	Dynamic Model	Motion		8/Sep/21	14/Sep/21	100%
	Integrate Modelling			15/Sep/21	21/Sep/21	100%
3	Estimation					
	Range Sensor	State Estimation	Tanach&Nattasit	31/Aug/21	7/Sep/21	100%
	IMU Sensor	State Estimation		8/Sep/21	14/Sep/21	100%
	Integrate estimation of IMU and Range sensor			15/Sep/21	21/Sep/21	100%
	Kalman filter	Do kalman filter of state estimation of IMU and Range sensor			25/Sep/21	75%
4	Simulation and Visualization					
	Do Plant		Tanach&Nattasit	22/Sep/21	28/Sep/21	100%
	Controller	PID Tuning				100%
	3D-plot graph			29/Sep/21	5/Oct/21	100%
5	Further Develop	Visualization Develop	everyone	6/Oct/21	19/Oct/21	25%
		Adding sensor				25%

	Presentation					
6	Rechecking		everyone	20/Oct/21	26/Oct/21	0%
	report and presentation			27/Oct/21	24/Nov/21	0%
	Proposal Presentation			9/Sep/21	9/Sep/21	100%
	Progress Presentation			19/Oct/21	19/Oct/21	100%
	Final Presentation			25/Nov/21	25/Nov/21	0%