Quadrotor Three

Member

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

Advisor

- Aj.Thanacha Choopojcharoen

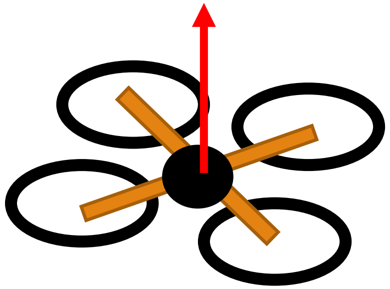
Topic

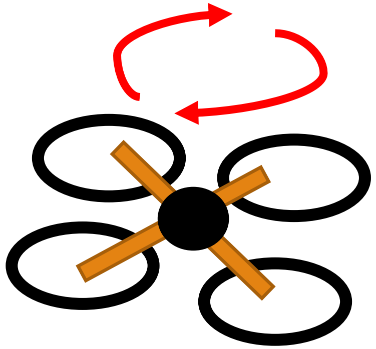
1. Introduction
2. Quadrotor Model & Control System
3. Sensor Model & State-Space Estimation
4. Vitualization

I. Introduction

Scope

- ระบบทั้งหมดจะทำการพัฒนาและจำลองขึ้นบนคอมพิวเตอร์ โดยใช้โปรแกรม MATLAB Sinulink
- ศึกษาการเคลื่อนที่แบบ altitude และการหมุนแบบ yaw เท่านั้น โดยที่ให้ Lateral position มีการเปลี่ยนแปลงน้อยมากๆ
- Input ของระบบคือมาจาก user interface ที่จะทำการป้อน Altitude หรือความสูงที่ต้องการให้ Quadrotor เคลื่อนที่ไป
- ใช้แรงและแรงบิดที่เกิดจากมอเตอร์ตามแนวแกนหมุนในการควบคุมการเคลื่อนที่ของ Quadrotor
- พารามิเตอร์ทางกายภาพต่างๆของระบบให้เป็นค่าคงที่
- ใช้ sensor อย่างน้อย 2 ชนิดในการติดตั้งกับตัว Quadrotor คือ 6-axis IMU sensor และ Range sensor
- แสดงการจำลองจากผลที่ได้เป็นกราฟ 3 มิติ โดยเป็นกราฟระยะทางที่เคลื่อนที่และการหมุนเทียบกับเวลา





Ⅱ. Quadrotor Model & Control System

II Quadrotor Model & Control System

2.1 Kinematic Model of Quadrotor

2.2 Dynamics Model of Quadrotor

2.3 Control System

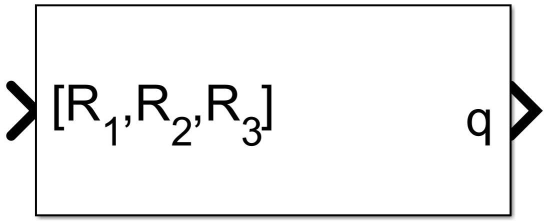
II Quadrotor Model & Control System (2.1)

2.1 Kinematic Model of Quadrotor

- Rotation (Euler's Angle)

Z-Y-X euler angles Global frame to Body frame

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



Rotation Order: ZYX

Quadrotor Model & Control System (2.1)

- Orientation Kinematics (Quaternion)

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

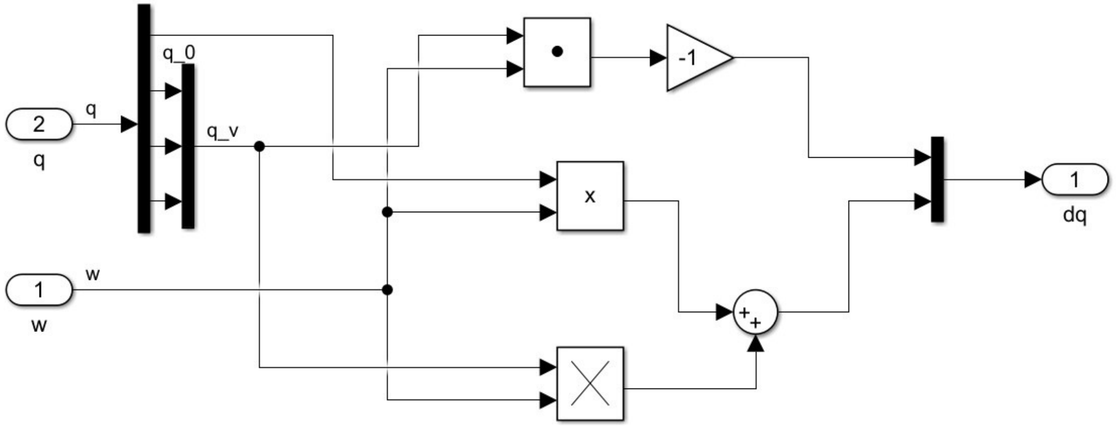
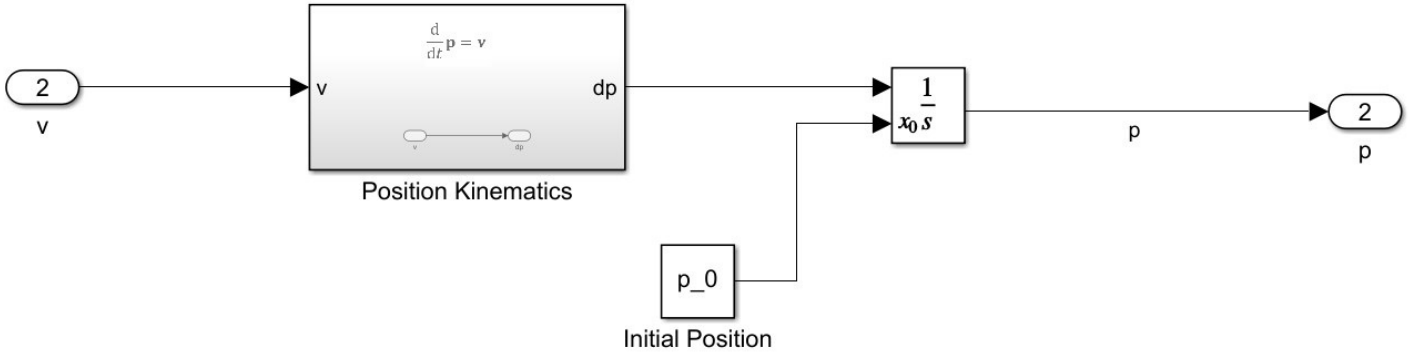


Figure 1: block orientation kinematic

- Position Kinematics



Quadrotor Model & Control System (2.2)

2.2 Dynamics Model of Quadrotor

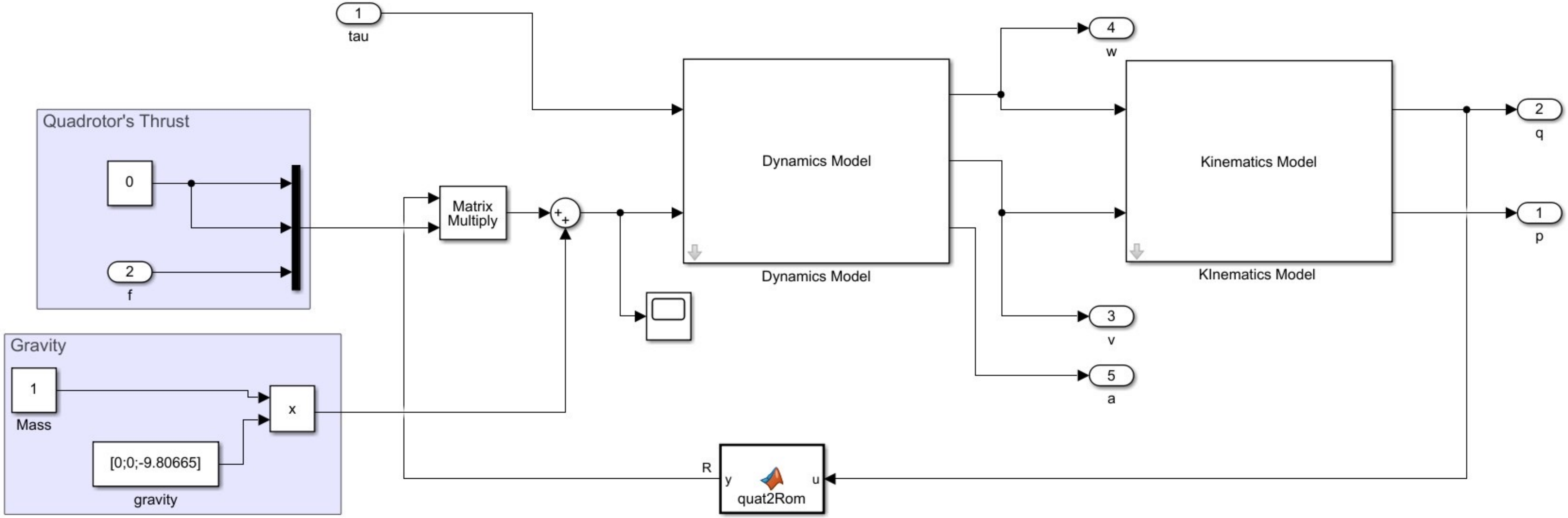


Figure 2: the overview of the quadrotor model

II Quadrotor Model & Control System (2.2)

- Orientation Dynamics

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

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

$\rightarrow$ $\boxed{\text{Integrate}}$ $\rightarrow$ $\boxed{\omega}$

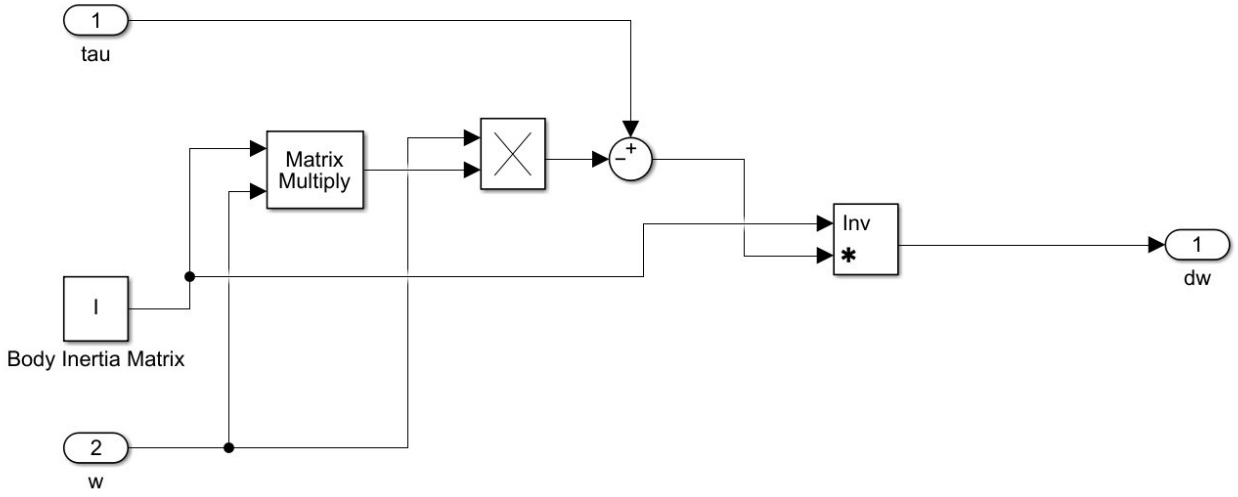


Figure 3: the Orientation Dynamics block

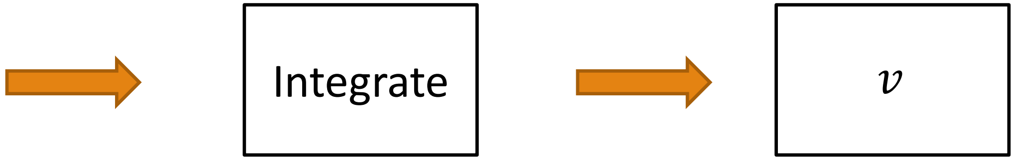
II Quadrotor Model & Control System (2.2)

- Position Dynamics

Newton’s second Law of motion

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

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



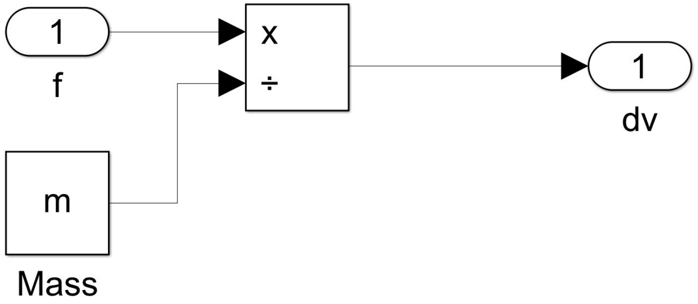


Figure 4: the Position Dynamics block

II Quadrotor Model & Control System (2.3)

2.3 Dynamics Model of Quadrotor

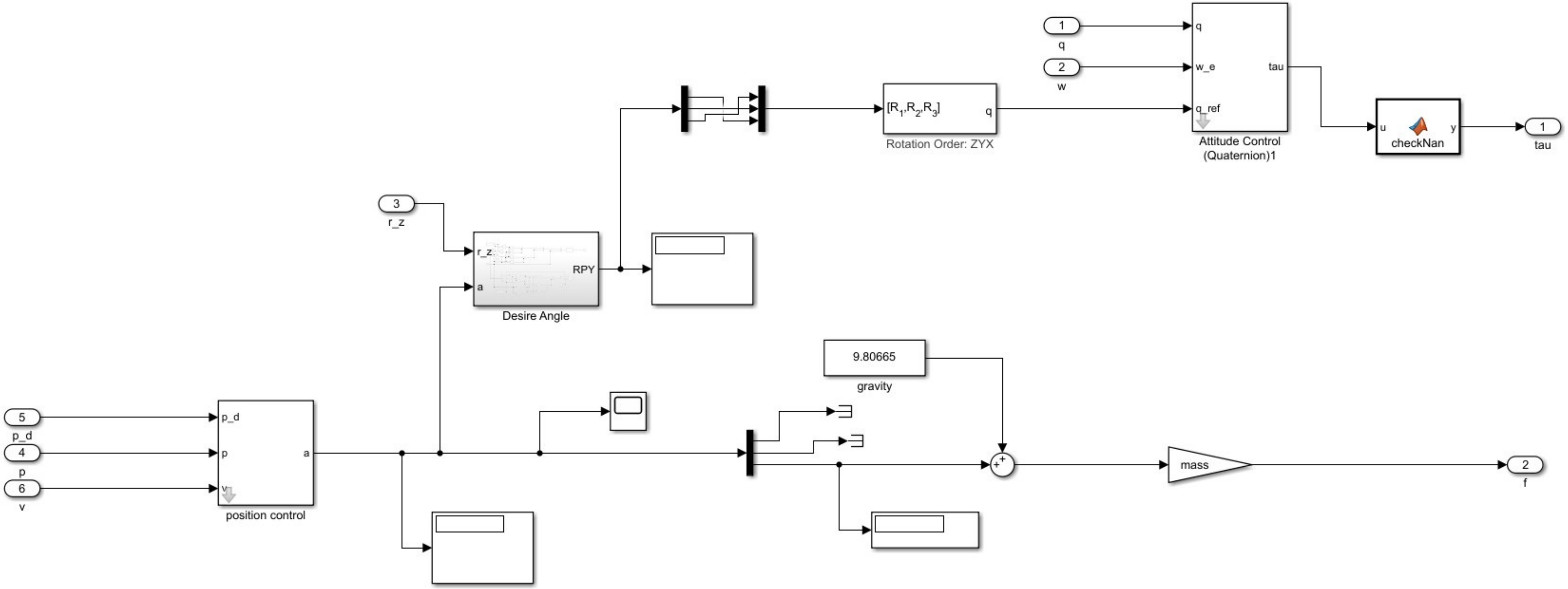


Figure 5: the overview of the hover control

II Quadrotor Model & Control System (2.3)

- Lateral Flight

Trajectory plan : Quintic Polynomial

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

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

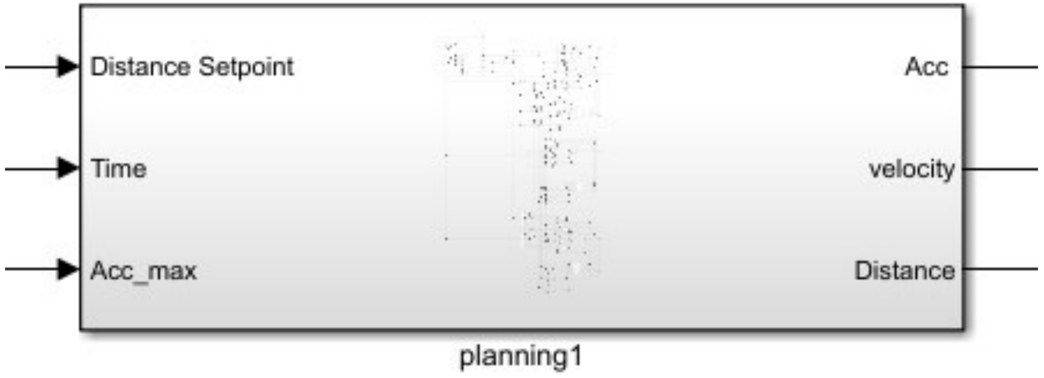
$$\frac{d}{dt} v(t) = a(t) = 2C_2 + 6C_3 t + 12C_4 t^2 + 20C_5 t^3$$

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

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

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

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



II Quadrotor Model & Control System (2.3)

- Altitude Control

To control force → acceleration → PID Controller

$$\Delta\vec{p} = \vec{p}_d - \vec{p}.$$

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

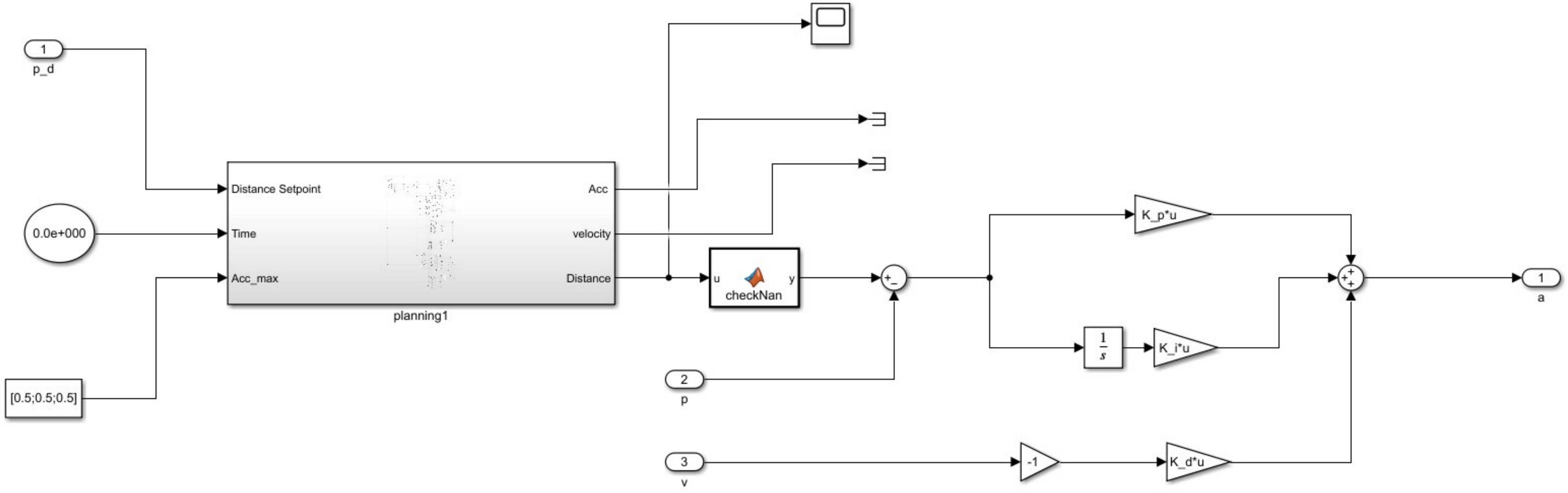


Figure 6: the overview of Altitude Control block

II Quadrotor Model & Control System (2.3)

- Attitude Control

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

$$m\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} c_\psi c_\theta - s_\phi s_\psi s_\theta & -c_\phi s_\psi & c_\psi s_\theta + c_\theta s_\phi s_\psi \\ s_\psi c_\theta - s_\phi c_\psi s_\theta & c_\phi c_\psi & s_\psi s_\theta - c_\theta s_\phi c_\psi \\ -c_\phi s_\theta & s_\phi & c_\theta c_\phi \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

$\Rightarrow$

Linearization

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

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

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

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

II Quadrotor Model & Control System (2.3)

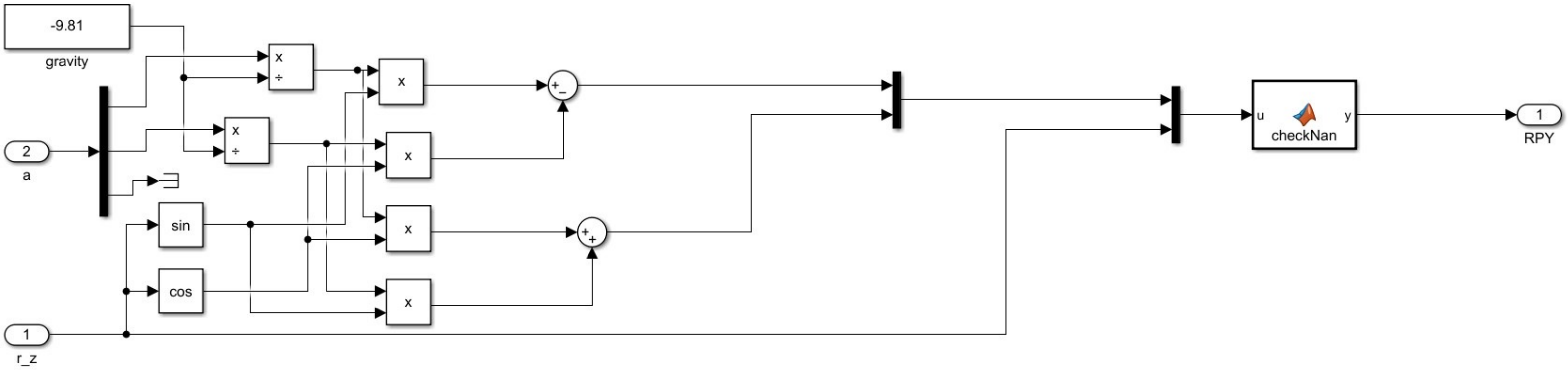


Figure 7: the overview of Desire Angle block

II Quadrotor Model & Control System (2.3)

- The quaternion-based attitude control

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

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

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

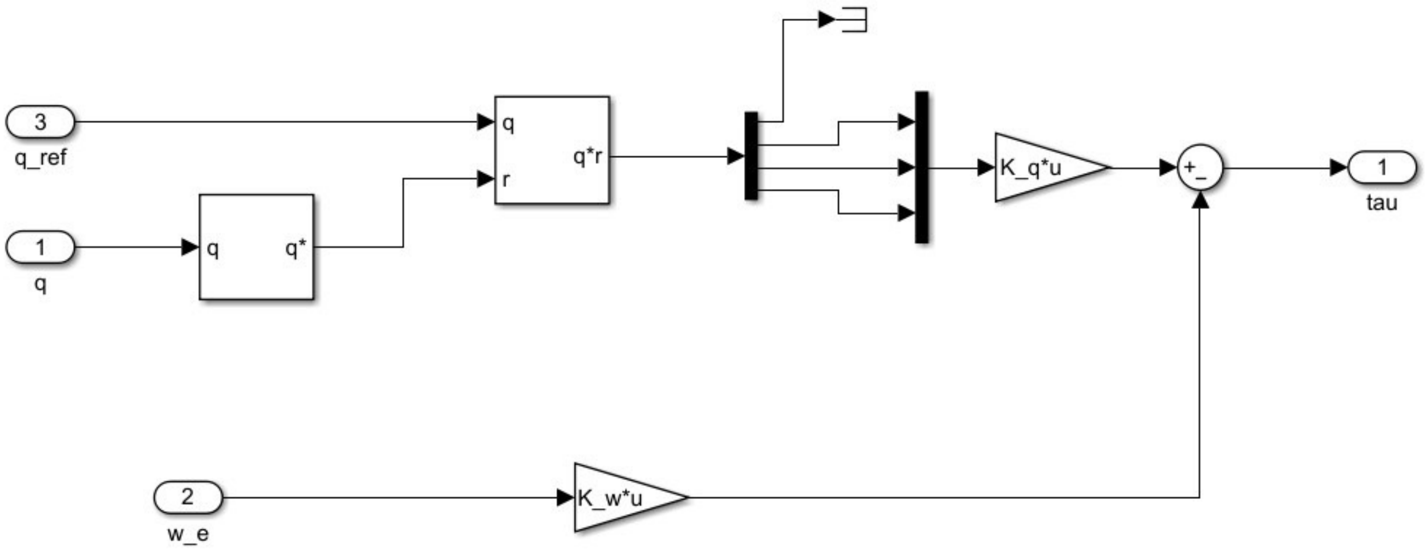
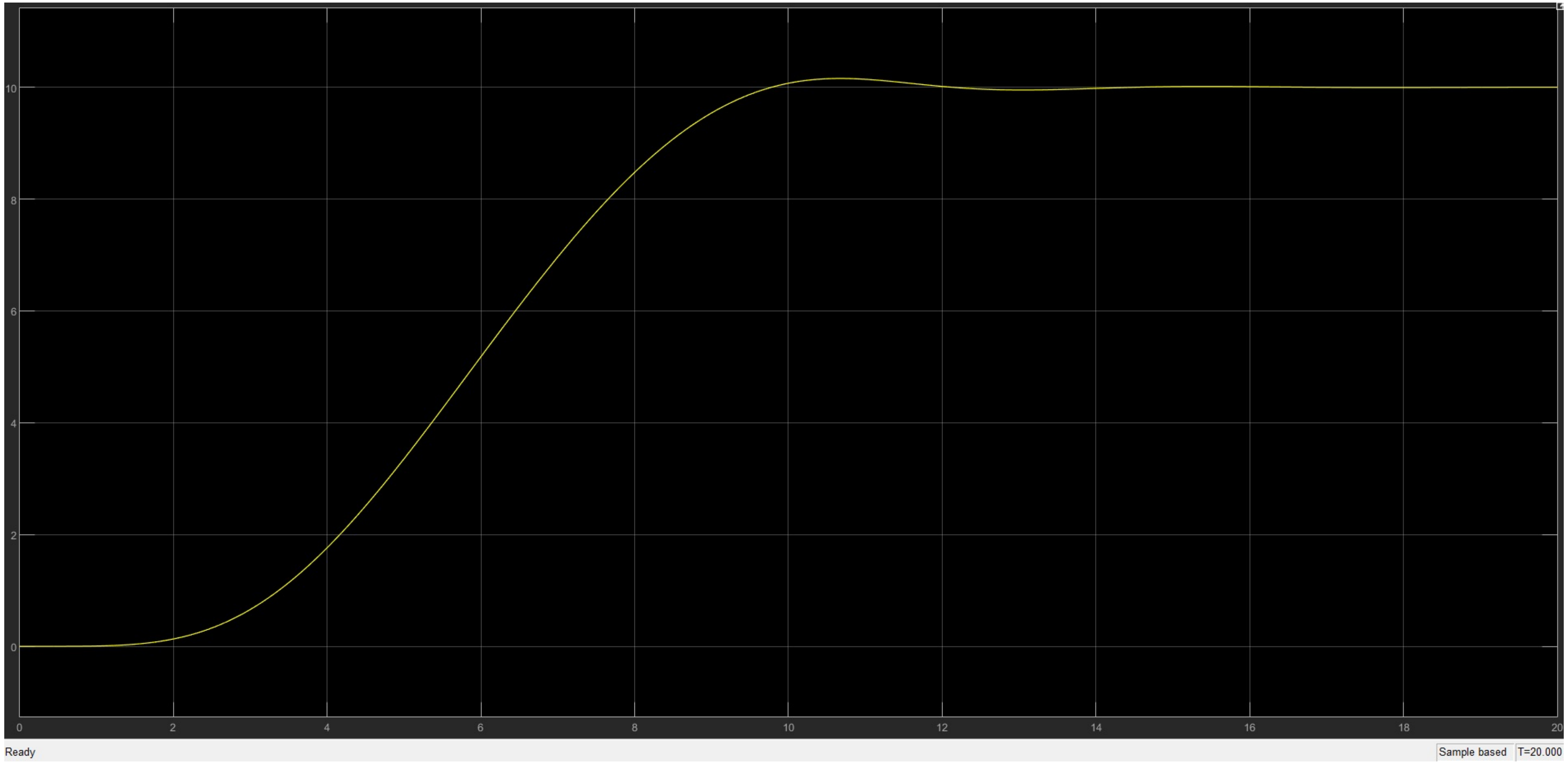


Figure 8: the overview of Attitude Control Block

II Quadrotor Model & Control

The result from Model & Control System



III. Sensor Model & State-Space Estimation

III. Sensor Model & State-Space Estimation

3.1 Sensor Model

3.2 Estimator

3.3 Implementation

III. Sensor Model & State-Space Estimation (3.1)

3.1 Sensor Model ➡ 6-axis IMU (3-axis accelerometer, 3-axis gyroscope) and range sensor (ultrasonic)

- Ultrasonic Sensor (Z_u)

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

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

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

$$\boxed{\boldsymbol{Z_u = \frac{p_z}{c_\theta c_\emptyset}}}$$

III. Sensor Model & State-Space Estimation (3.1)

- Accelerometer (Z_a)

Sensor read value

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

$\rightarrow$ Newton's second laws $\rightarrow$

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

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

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

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

$$\boxed{\boldsymbol{Z_a = \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix}}}$$

III. Sensor Model & State-Space Estimation (3.1)

- Gyroscope Sensor

$$\boxed{Z_g = \omega}$$

III. Sensor Model & State-Space Estimation (3.1)

$$\begin{bmatrix} Z_u \\ Z_a \\ Z_g \end{bmatrix} = \begin{bmatrix} \dfrac{\vec{P}_z}{\mathrm{c}_\theta \mathrm{c}_\emptyset} \\ \dfrac{1}{m}\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \\ \omega \end{bmatrix}$$

III. Sensor Model & State-Space Estimation (3.2)

3.2 Estimator

- Position Estimator → Kalman Filter

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

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

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

$$A = \begin{bmatrix} I_3 & \Delta t \cdot I_3 & 0_{3x3} \\ 0_{3x3} & I_3 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & 0_{3x3} \end{bmatrix}$$

$$B = \begin{bmatrix} I_9 & 0_{9x4} \end{bmatrix}$$

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

$$D = \begin{bmatrix} 0_{4x9} & I_4 \end{bmatrix}$$

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

III. Sensor Model & State-Space Estimation (3.2)

- Orientation Estimator → Extended Kalman Filter

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

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

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

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

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

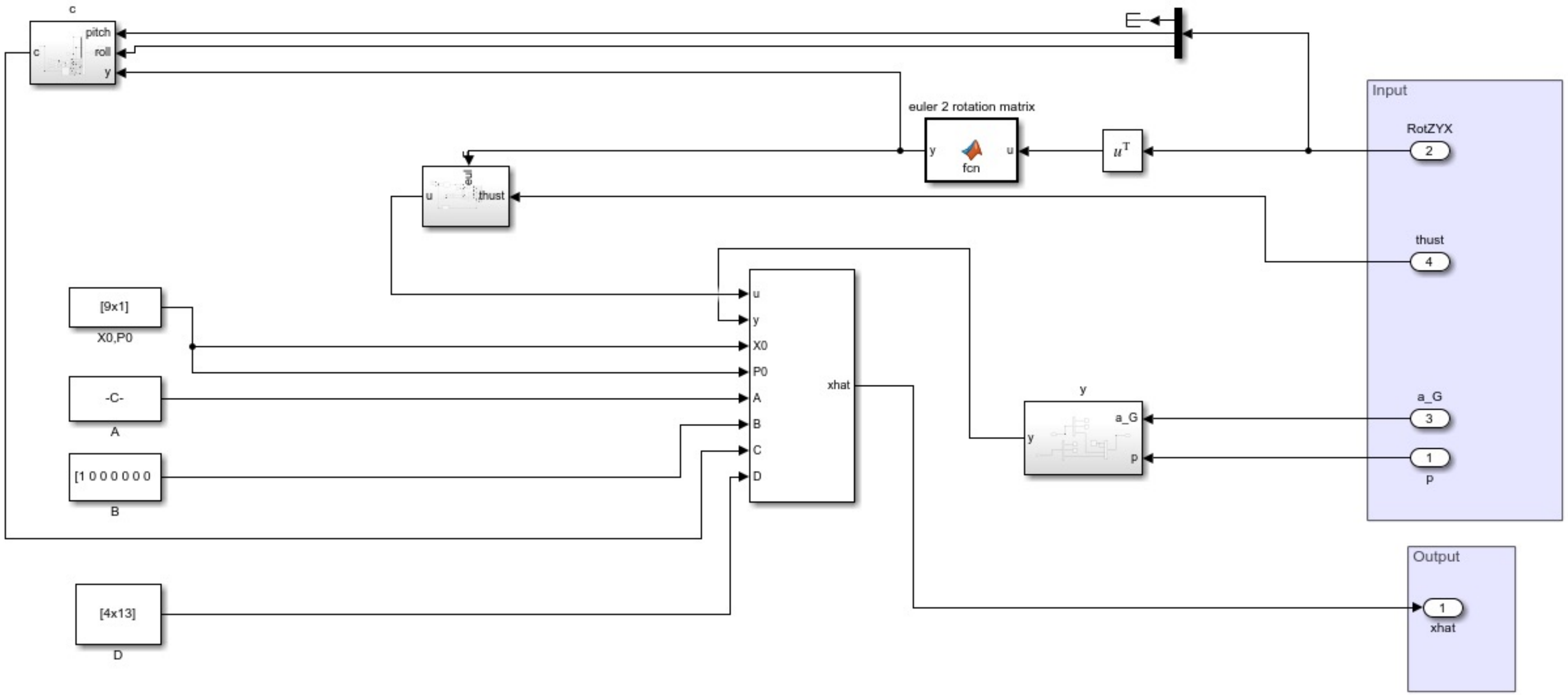
* ($\dot{\vec{\alpha}}$) is 0 mean gaussian

III. Sensor Model & State-Space Estimation (3.3)

3.3 Implementation

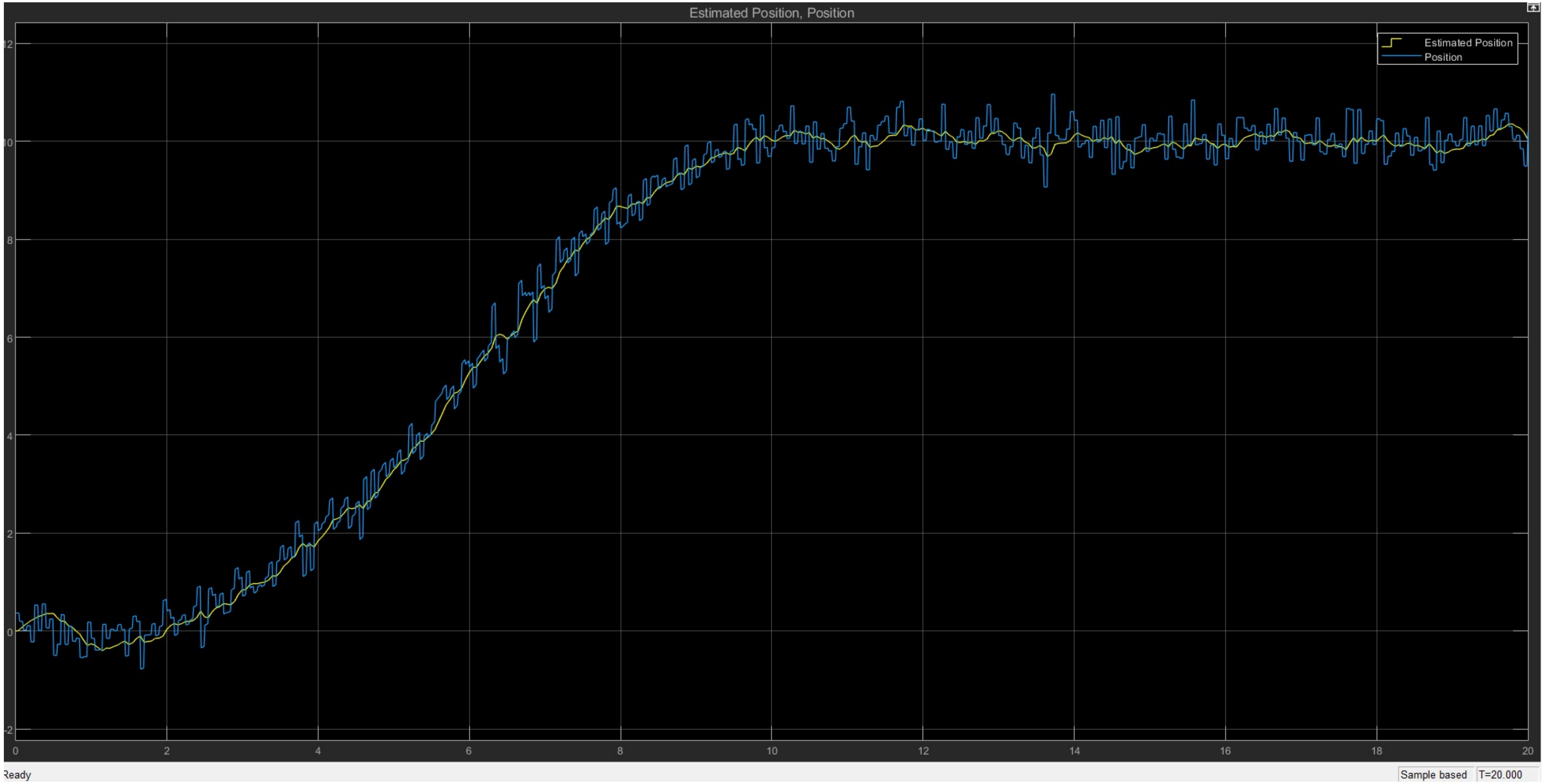
- Position Estimator

  In Simulink , Create matrix A ,B ,C ,D ,u ,y ,X0 and P0 and connect them to Kalman Filter block and use sample rate at 100 Hz



III. Sensor Model & State-Space Estimation (3.3)

This graph show ‘Position with noise ,time’ and ‘Position with noise (Kalman filter feedback) ,time’



III. Sensor Model & State-Space Estimation (3.3)

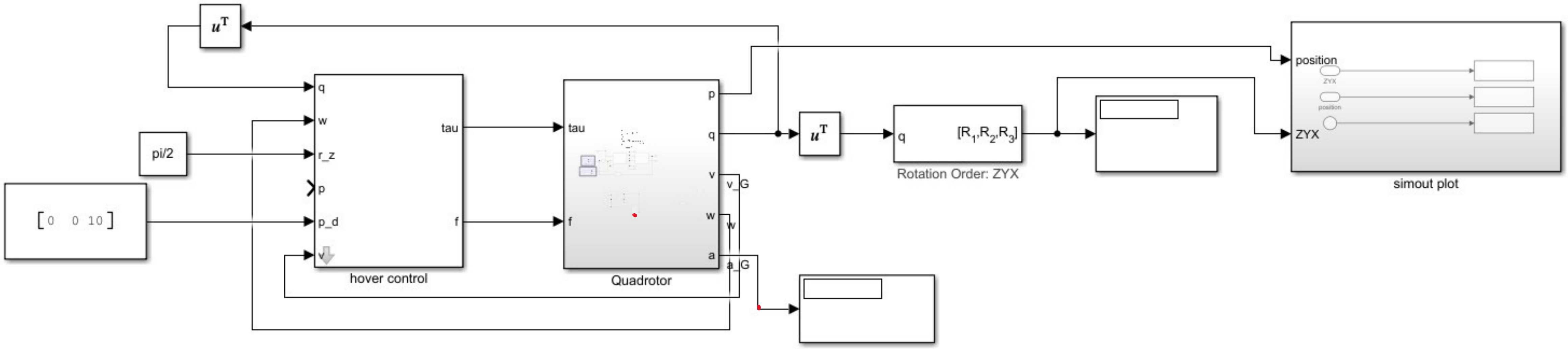
- Orientation Estimator

Implementation is in progress

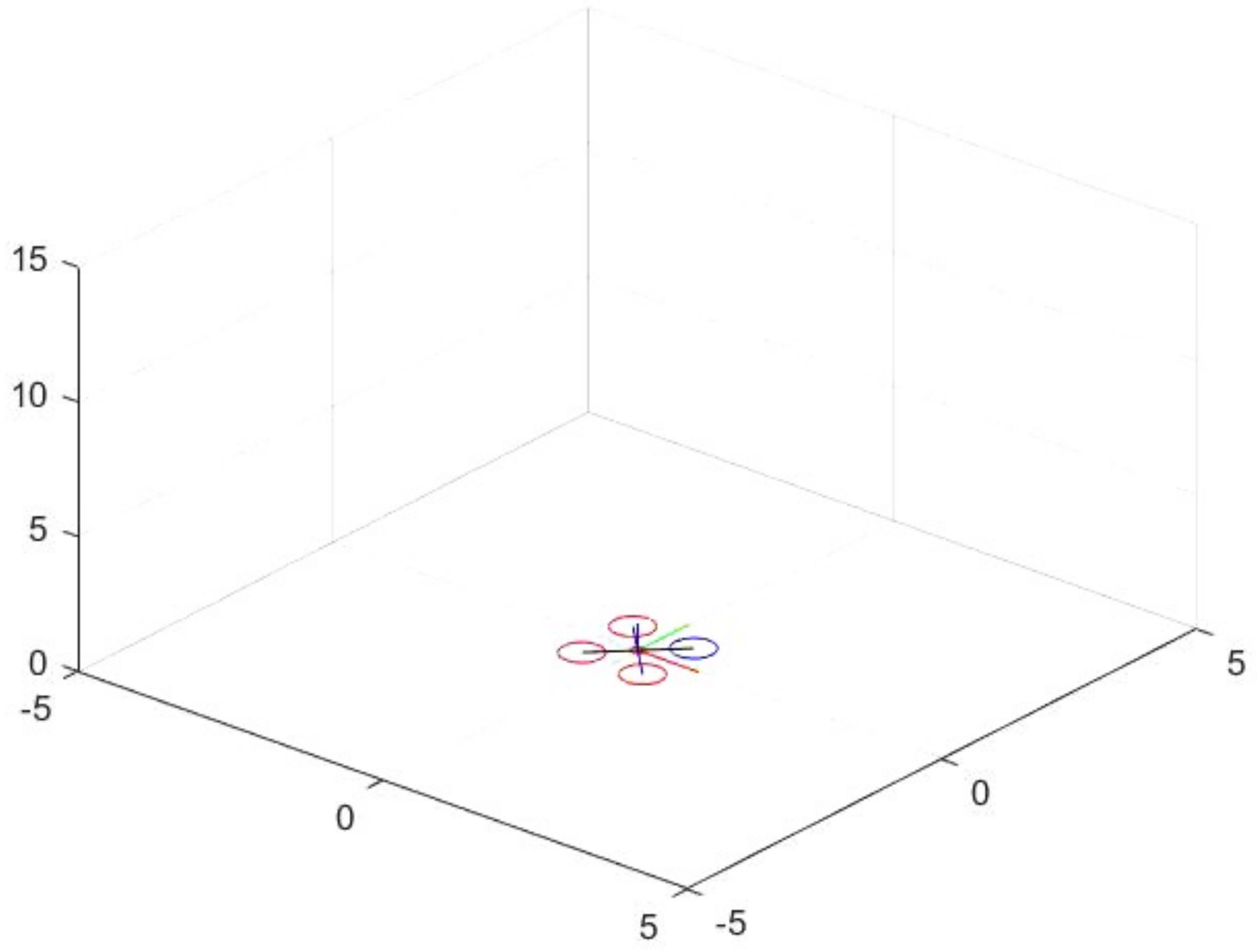
IV Vitualization

IV Vitualization

- Bring values from model to do 3D-plot



Vitualization of Quadrotor



Set point : (0,0,10)

THANK YOU