Multicopter Design and Control Practice
—— A Series Experiments Based on MATLAB and Pixhawk

Lesson 06 Dynamic Modeling Experiment

Quan Quan, Associate Professor, qq_buaa@buaa.edu.cn
School of Automation Science and Electrical Engineering,
BeihangUniversity, Beijing 100191, China.

Outline

1. Preliminary
2. Basic Experiment
3. Analysis Experiment
4. Design Experiment
5. Summary

Preliminary

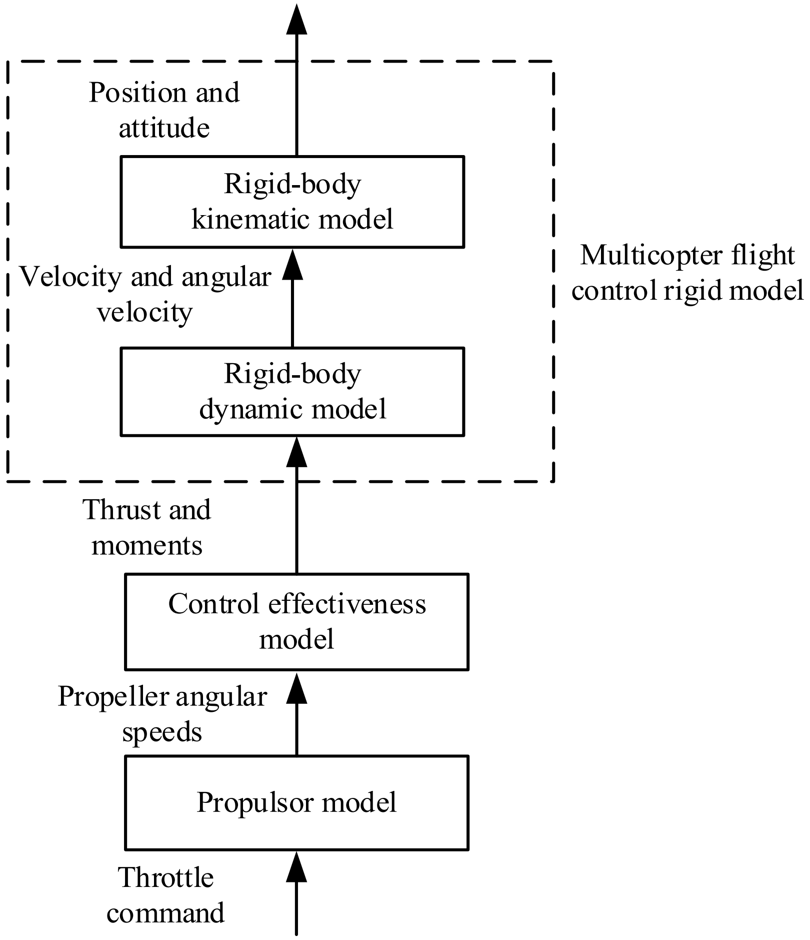


Figure. Architecture of multicopter modeling

（1）**Rigid-body kinematic model.** Kinematics are independent of the mass and force. It only examines variables such as position, velocity, attitude and angular velocity.

（2）**Rigid-body dynamic model.** Dynamics involve both the movement and the force. They are related to the object’s mass and moments of inertia. For the multicopter dynamic model, the inputs include thrust and moments (pitching moment, rolling moment, and yawing moment), and the outputs include velocity and angular velocity.

Preliminary

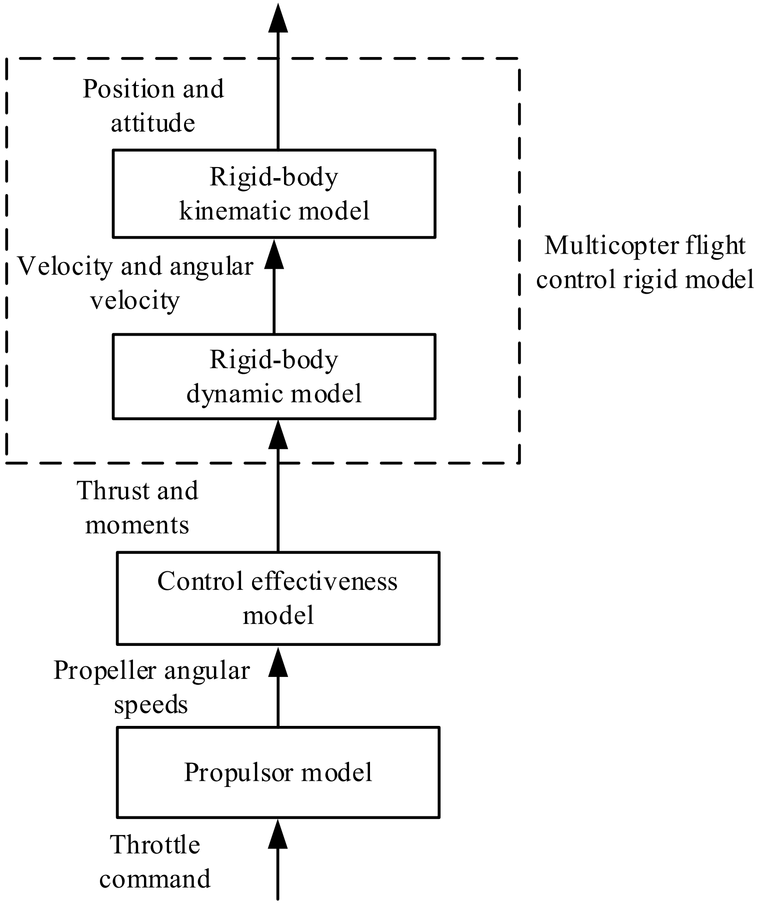


Figure. Architecture of multicopter modeling

（3）**Control effectiveness model.** The inputs include propeller angular speeds, and the outputs include thrust and moments. For either a quadcopter or a hexacopter, the thrust and moments are all generated by propellers. Given the propeller angular speeds, the thrust and moments can be calculated by using control effectiveness model.

（4）**Propulsor model.** The propulsor model is a whole power mechanism that includes a brushless Direct Current (DC) motor, an Electronic Speed Controller (ESC), and a propeller. The input is a throttle command between 0 and 1 and the outputs are propeller angular speeds.

Preliminary

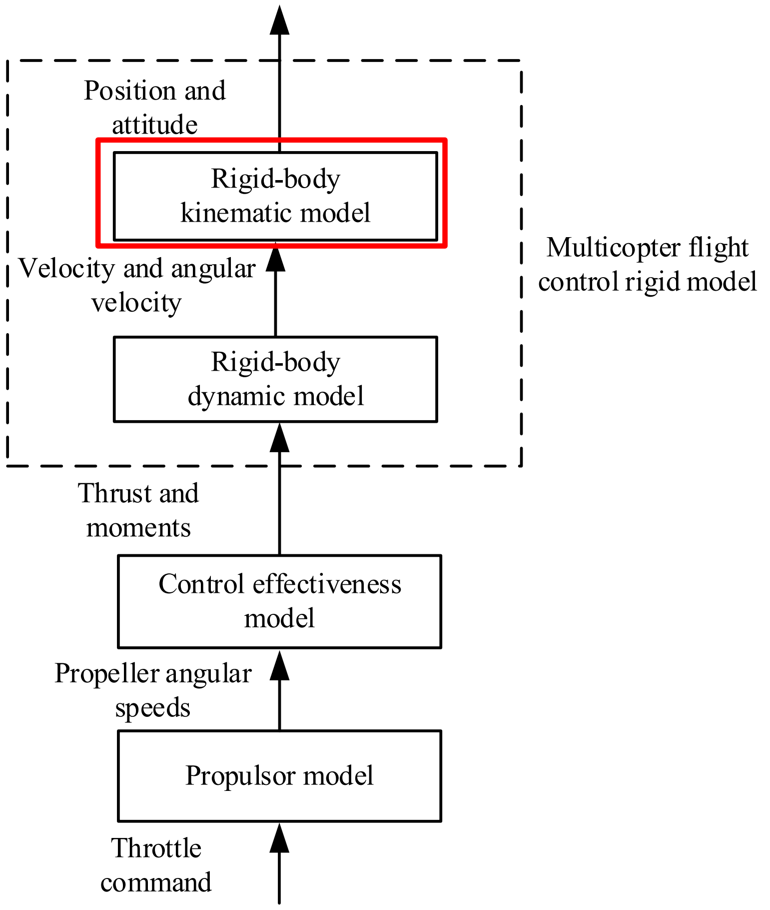


Figure. Architecture of multicopter modeling

- **Euler angle model**

$$ {}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} $$
$$ \dot{\boldsymbol{\Theta}} = \mathbf{W}\,{}^{\mathrm{b}}\boldsymbol{\omega} $$

- **Rotation matrix model**

$$ {}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} $$
$$ \dot{\mathbf{R}} = \mathbf{R}\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times} $$

- **Quaternions model**

$$ {}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} $$
$$ \dot{q}_0 = -\frac{1}{2}\mathbf{q}_{\mathrm{v}}^{\mathrm{T}} \cdot {}^{\mathrm{b}}\boldsymbol{\omega} $$
$$ \dot{\mathbf{q}}_{\mathrm{v}} = \frac{1}{2}\left(q_0\mathbf{I}_3 + \left[\mathbf{q}_{\mathrm{v}}\right]_{\times}\right){}^{\mathrm{b}}\boldsymbol{\omega} $$

Preliminary

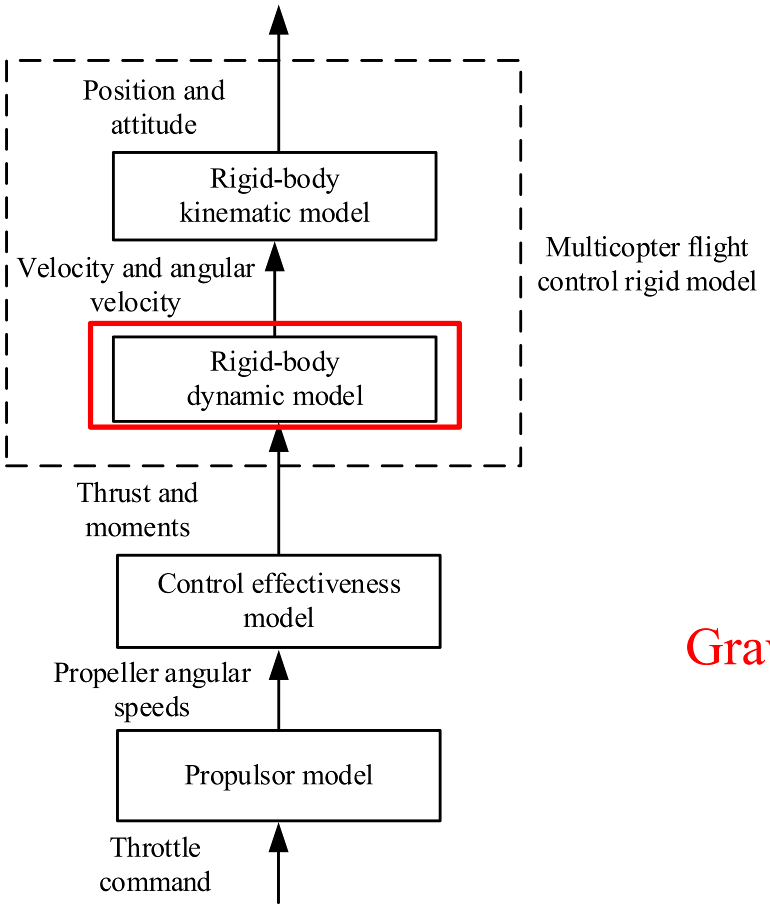


Figure. Architecture of multicopter modeling

■ Position dynamic model

$${}^{\mathrm{e}}\dot{\mathbf{v}} = {}^{\mathrm{e}}\mathbf{F}/m$$

The total force ${}^{\mathrm{e}}\mathbf{F}$ is composed of gravity, propeller control force, and aerodynamic force.

where

$${}^{\mathrm{e}}\mathbf{F} = m\mathbf{G} + \mathbf{R}\left({}^{\mathrm{b}}\mathbf{T} + {}^{\mathrm{b}}\mathbf{F}_{\mathrm{d}}\right)$$

Aerodynamic force

$$\mathbf{G} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^{\mathrm{T}} = g\mathbf{e}_3$$

Gravity

$${}^{\mathrm{b}}\mathbf{T} = \begin{bmatrix} 0 & 0 & -f \end{bmatrix}^{\mathrm{T}} = -f\mathbf{b}_3$$

f represents the magnitude of the total propeller thrust and the thrust is unidirectional (the situation of negative thrust caused by variable-pitch propellers is not considered here).

Preliminary

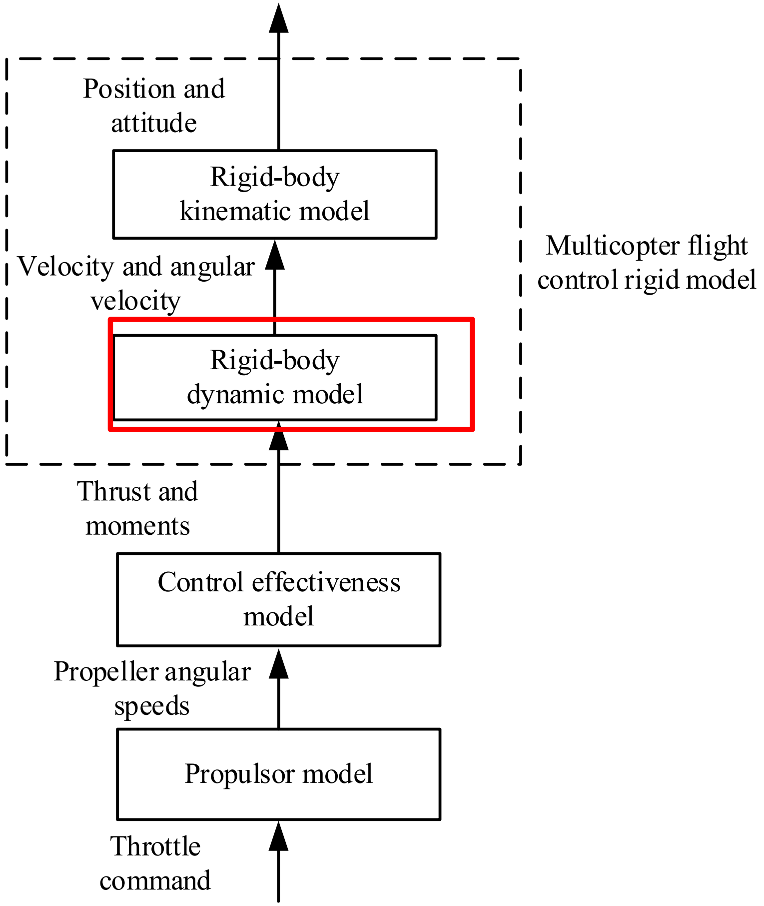


Figure. Architecture of multicopter modeling

■ Attitude dynamic model

The attitude dynamic equation in the ABCF is established as follows

$$\mathbf{J} \cdot {}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = -{}^{\mathrm{b}}\boldsymbol{\omega} \times (\mathbf{J} \cdot {}^{\mathrm{b}}\boldsymbol{\omega}) + {}^{\mathrm{b}}\mathbf{M}$$

$${}^{\mathrm{b}}\mathbf{M} = \mathbf{G}_{\mathrm{a}} + \boldsymbol{\tau} + {}^{\mathrm{b}}\mathbf{M}_{\mathrm{d}}$$

where $\boldsymbol{\tau} \triangleq \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ denotes the moments generated by the propellers in the body axes; $\mathbf{J} \in \mathbb{R}^{3\times 3}$ denotes the multicopter moment of inertia; and ${}^{\mathrm{b}}\mathbf{M}_{\mathrm{d}} \in \mathbb{R}^3$ denotes the aerodynamic moment acting on the body; for a multicopter, $\mathbf{G}_{\mathrm{a}} \triangleq \begin{bmatrix} G_{\mathrm{a},\phi} & G_{\mathrm{a},\theta} & G_{\mathrm{a},\psi} \end{bmatrix} \in \mathbb{R}^3$ denotes the gyroscopic torques.

Preliminary

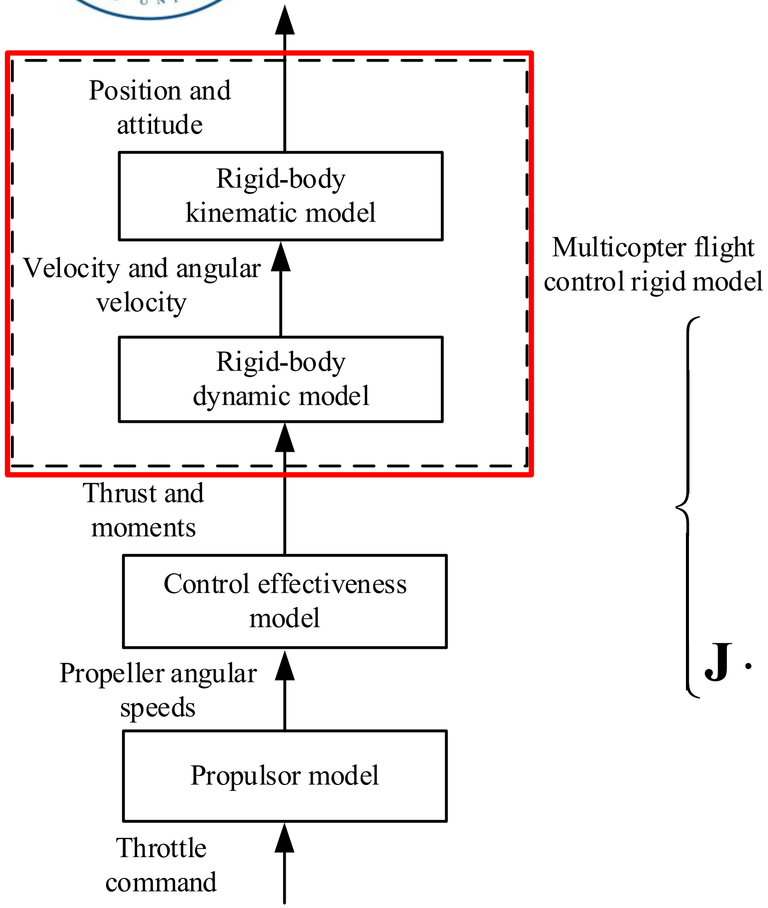


Figure. Architecture of multicopter modeling

- **Euler angle model**

$$\begin{cases} {}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} = \mathbf{R}\cdot{}^{\mathrm{b}}\mathbf{v} \\ {}^{\mathrm{b}}\dot{\mathbf{v}} = -\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\cdot{}^{\mathrm{b}}\mathbf{v} + {}^{\mathrm{b}}\mathbf{F}/m \\ \dot{\boldsymbol{\Theta}} = \mathbf{W}\cdot{}^{\mathrm{b}}\boldsymbol{\omega} \\ \mathbf{J}\cdot{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = -{}^{\mathrm{b}}\boldsymbol{\omega}\times(\mathbf{J}\cdot{}^{\mathrm{b}}\boldsymbol{\omega}) + {}^{\mathrm{b}}\mathbf{M} \end{cases}$$

- **Rotation matrix model**

$$\begin{cases} {}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} = \mathbf{R}\cdot{}^{\mathrm{b}}\mathbf{v} \\ {}^{\mathrm{b}}\dot{\mathbf{v}} = -\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\cdot{}^{\mathrm{b}}\mathbf{v} + {}^{\mathrm{b}}\mathbf{F}/m \\ \dot{\mathbf{R}} = \mathbf{R}\cdot\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times} \\ \mathbf{J}\cdot{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = -{}^{\mathrm{b}}\boldsymbol{\omega}\times(\mathbf{J}\cdot{}^{\mathrm{b}}\boldsymbol{\omega}) + {}^{\mathrm{b}}\mathbf{M} \end{cases}$$

- **Quaternions model**

$$\begin{cases} {}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} = \mathbf{R}\cdot{}^{\mathrm{b}}\mathbf{v} \\ {}^{\mathrm{b}}\dot{\mathbf{v}} = -\left[{}^{\mathrm{b}}\boldsymbol{\omega}\right]_{\times}\cdot{}^{\mathrm{b}}\mathbf{v} + {}^{\mathrm{b}}\mathbf{F}/m \\ \dot{q}_0 = -\frac{1}{2}\mathbf{q}_{\mathrm{v}}^{\mathrm{T}}\cdot{}^{\mathrm{b}}\boldsymbol{\omega} \\ \dot{\mathbf{q}}_v = \frac{1}{2}\left(q_0\mathbf{I}_3 + [\mathbf{q}_{\mathrm{v}}]_{\times}\right){}^{\mathrm{b}}\boldsymbol{\omega} \\ \mathbf{J}\cdot{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = -{}^{\mathrm{b}}\boldsymbol{\omega}\times(\mathbf{J}\cdot{}^{\mathrm{b}}\boldsymbol{\omega}) + {}^{\mathrm{b}}\mathbf{M} \end{cases}$$

Preliminary

■ Two configurations of quadcopters

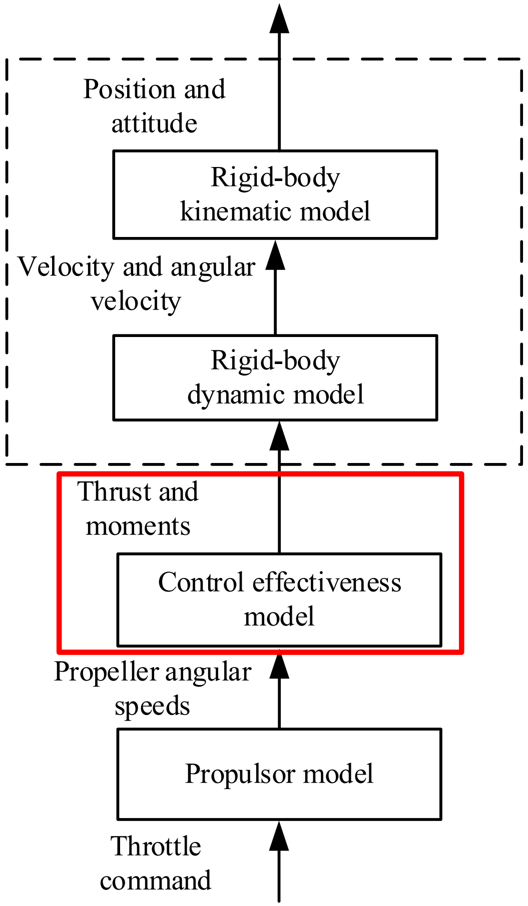
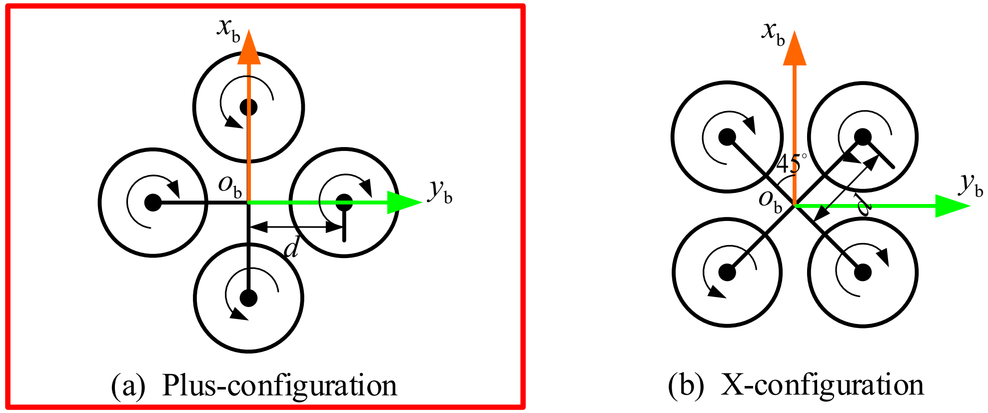


Figure. Architecture of multicopter modeling

The total thrust that acts on the quadcopter is as follows

Multicopter flight control rigid model

$$f = \sum_{i=1}^{4} T_i = c_\mathrm{T}\left(\varpi_1^2 + \varpi_2^2 + \varpi_3^2 + \varpi_4^2\right)$$



For a plus-configuration quadcopter, the moments produced by propellers are as follows

$$\tau_x = d c_\mathrm{T}\left(-\varpi_2^2 + \varpi_4^2\right)$$

$$\tau_y = d c_\mathrm{T}\left(\varpi_1^2 - \varpi_3^2\right)$$

$$\tau_z = c_\mathrm{M}\left(\varpi_1^2 - \varpi_2^2 + \varpi_3^2 - \varpi_4^2\right)$$

where $c_\mathrm{T} = \dfrac{1}{4\pi^2}\rho D_\mathrm{p}^4 C_\mathrm{T}, c_\mathrm{M} = \dfrac{1}{4\pi^2}\rho D_\mathrm{p}^5 C_\mathrm{M}$

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_\mathrm{T} & c_\mathrm{T} & c_\mathrm{T} & c_\mathrm{T} \\ 0 & -dc_\mathrm{T} & 0 & dc_\mathrm{T} \\ dc_\mathrm{T} & 0 & -dc_\mathrm{T} & 0 \\ c_\mathrm{M} & -c_\mathrm{M} & c_\mathrm{M} & -c_\mathrm{M} \end{bmatrix}}_{\mathbf{M}_4} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix}$$

■ Two configurations of quadcopters

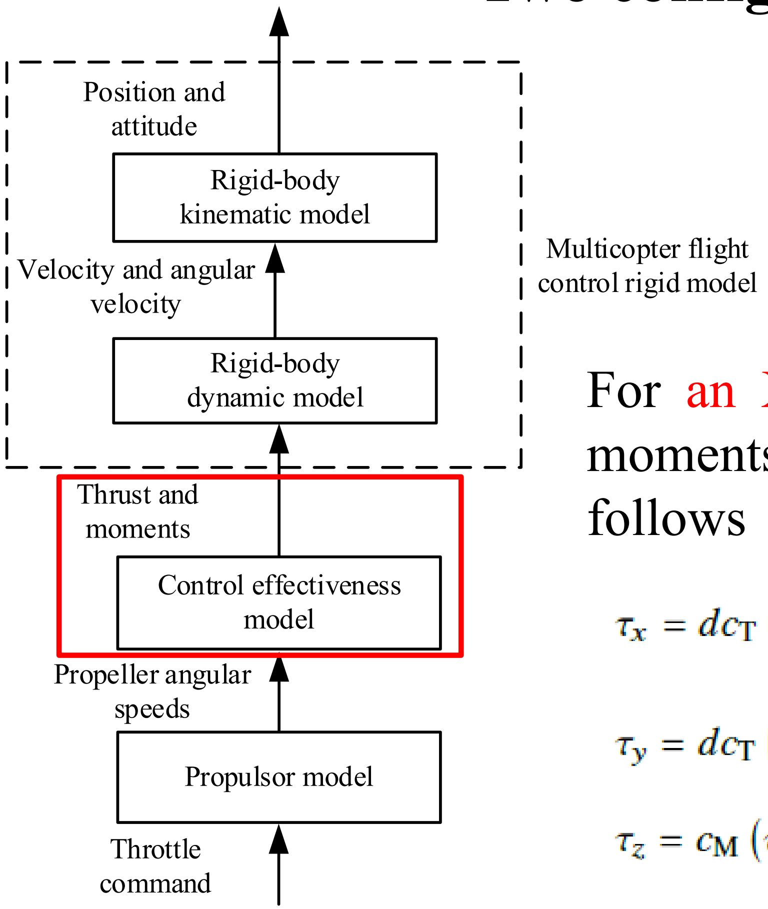
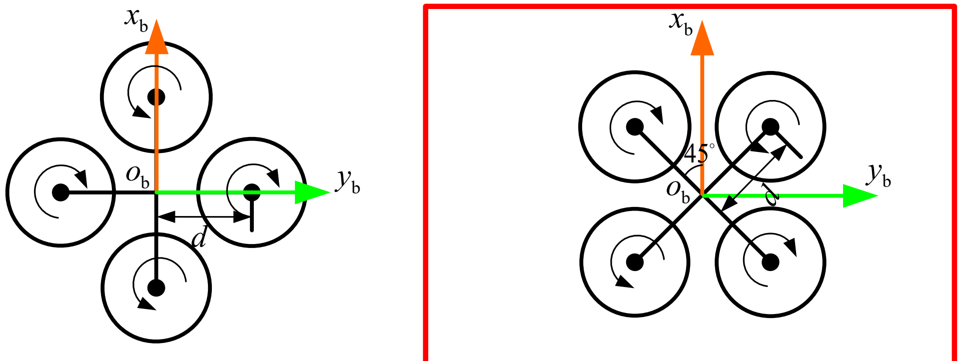


Figure. Architecture of multicopter modeling

For an X-configuration quadcopter, the total thrust produced by propellers is still

$$f=\sum_{i=1}^{4}T_i=c_{\mathrm{T}}\left(\varpi_1^2+\varpi_2^2+\varpi_3^2+\varpi_4^2\right)$$



(a) Plus-configuration (b) X-configuration

For an X-configuration quadcopter, the moments produced by propellers are as follows

$$\tau_x=dc_{\mathrm{T}}\left(-\frac{\sqrt{2}}{2}\varpi_1^2+\frac{\sqrt{2}}{2}\varpi_2^2+\frac{\sqrt{2}}{2}\varpi_3^2-\frac{\sqrt{2}}{2}\varpi_4^2\right)$$

$$\tau_y=dc_{\mathrm{T}}\left(\frac{\sqrt{2}}{2}\varpi_1^2-\frac{\sqrt{2}}{2}\varpi_2^2+\frac{\sqrt{2}}{2}\varpi_3^2-\frac{\sqrt{2}}{2}\varpi_4^2\right)$$

$$\tau_z=c_{\mathrm{M}}\left(\varpi_1^2+\varpi_2^2-\varpi_3^2-\varpi_4^2\right).$$

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}=\begin{bmatrix} c_{\mathrm{T}} & c_{\mathrm{T}} & c_{\mathrm{T}} & c_{\mathrm{T}} \\ -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T \\ \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T & \frac{\sqrt{2}}{2}dc_T & -\frac{\sqrt{2}}{2}dc_T \\ c_{\mathrm{M}} & c_{\mathrm{M}} & -c_{\mathrm{M}} & -c_{\mathrm{M}} \end{bmatrix}\begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \varpi_3^2 \\ \varpi_4^2 \end{bmatrix}$$

Preliminary

■ **For a multicopter($n_{\mathrm{r}} \geq 5$)**

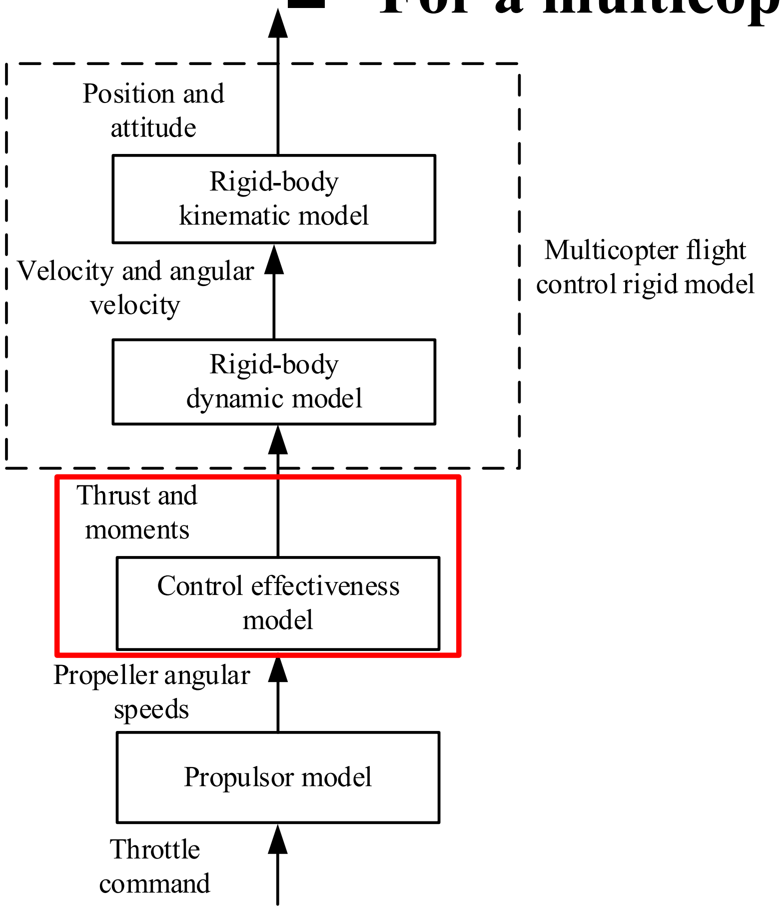
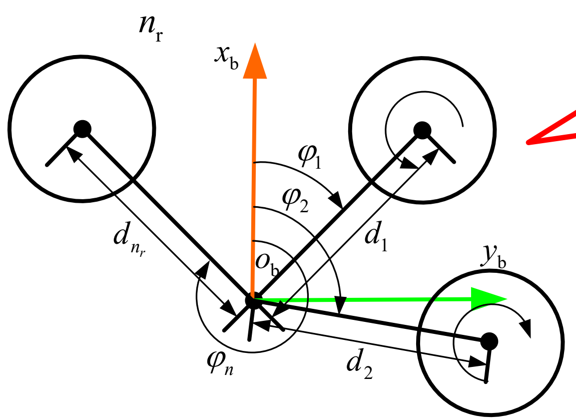


Figure. Architecture of multicopter modeling



Odd numbered propellers are marked in counterclockwise, and even numbered propellers are marked in clockwise.

Figure. Airframe geometry parameters of a multicopter

$$\begin{bmatrix} f \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} c_{\mathrm{T}} & c_{\mathrm{T}} & \cdots & c_{\mathrm{T}} \\ -d_1 c_{\mathrm{T}} \sin\varphi_1 & -d_2 c_{\mathrm{T}} \sin\varphi_2 & \cdots & -d_{n_r} c_{\mathrm{T}} \sin\varphi_{n_{\mathrm{r}}} \\ d_1 c_{\mathrm{T}} \cos\varphi_1 & d_2 c_{\mathrm{T}} \cos\varphi_2 & \cdots & d_{n_r} c_{\mathrm{T}} \cos\varphi_{n_{\mathrm{r}}} \\ c_{\mathrm{M}}\delta_1 & c_{\mathrm{M}}\delta_2 & \cdots & c_{\mathrm{M}}\delta_{n_{\mathrm{r}}} \end{bmatrix}}_{\mathbf{M}_{n_{\mathrm{r}}}} \begin{bmatrix} \varpi_1^2 \\ \varpi_2^2 \\ \vdots \\ \varpi_{n_{\mathrm{r}}}^2 \end{bmatrix}$$

where $\mathbf{M}_{n_{\mathrm{r}}} \in \mathbb{R}^{4\times n_r}, \delta_i = (-1)^{i+1}, i = 1, \cdots, n_{\mathrm{r}}$

Preliminary

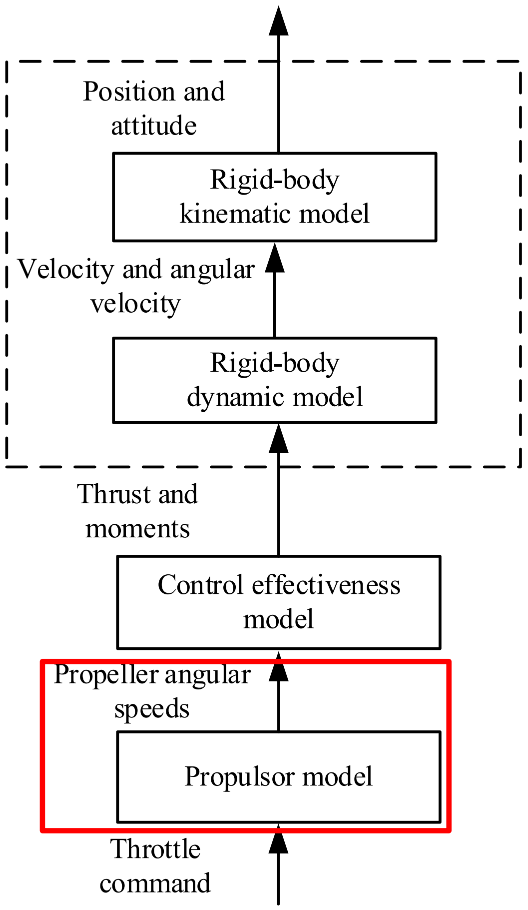


Figure. Architecture of multicopter modeling

■ Propulsor Model

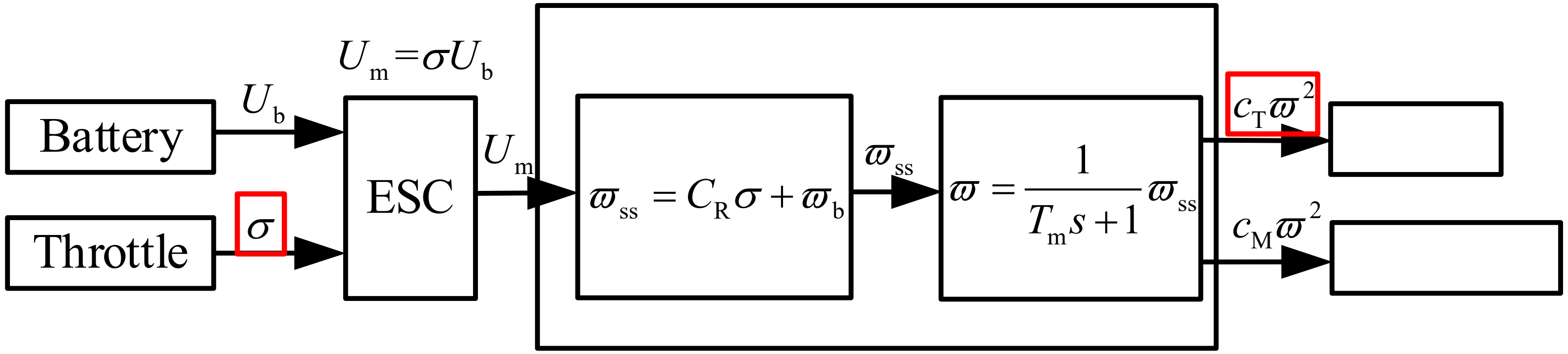


Figure. Signal flow of propulsor model

Generally, the dynamics of a brushless DC motor can be simplified as a first-order low-pass filter. Its transfer function is expressed as follows

$$\varpi = \frac{1}{T_m s+1}(C_R\sigma + \varpi_b)$$

where the input is the throttle command σ and the output is motor speed ϖ .

This time constant T_m denoted by determines the dynamic response.

Preliminary

In order to make this chapter self-contained, the preliminary is from Chapter. 5 and 6 of "**Quan Quan. *Introduction to Multicopter Design and Control*. Springer, Singapore, 2017**".

Basic Experiment

□ Experimental Objective

■ Things to prepare

Software: MATLAB R2017b or above, Instruction Package “e2.1”

（ https://rflysim.com/course).

■ Objectives

Analyze the flight performance with respect to the total mass, moment of inertia

matrix, and propeller parameters of a multicopter.

Basic Experiment

□ Analysis Procedure

（1）Step1： Flight state with respect to total mass changing

1) Open the file "e2\e2.1\e2_1.slx", open the file "Init_control.m" to initialize the parameters in the file.

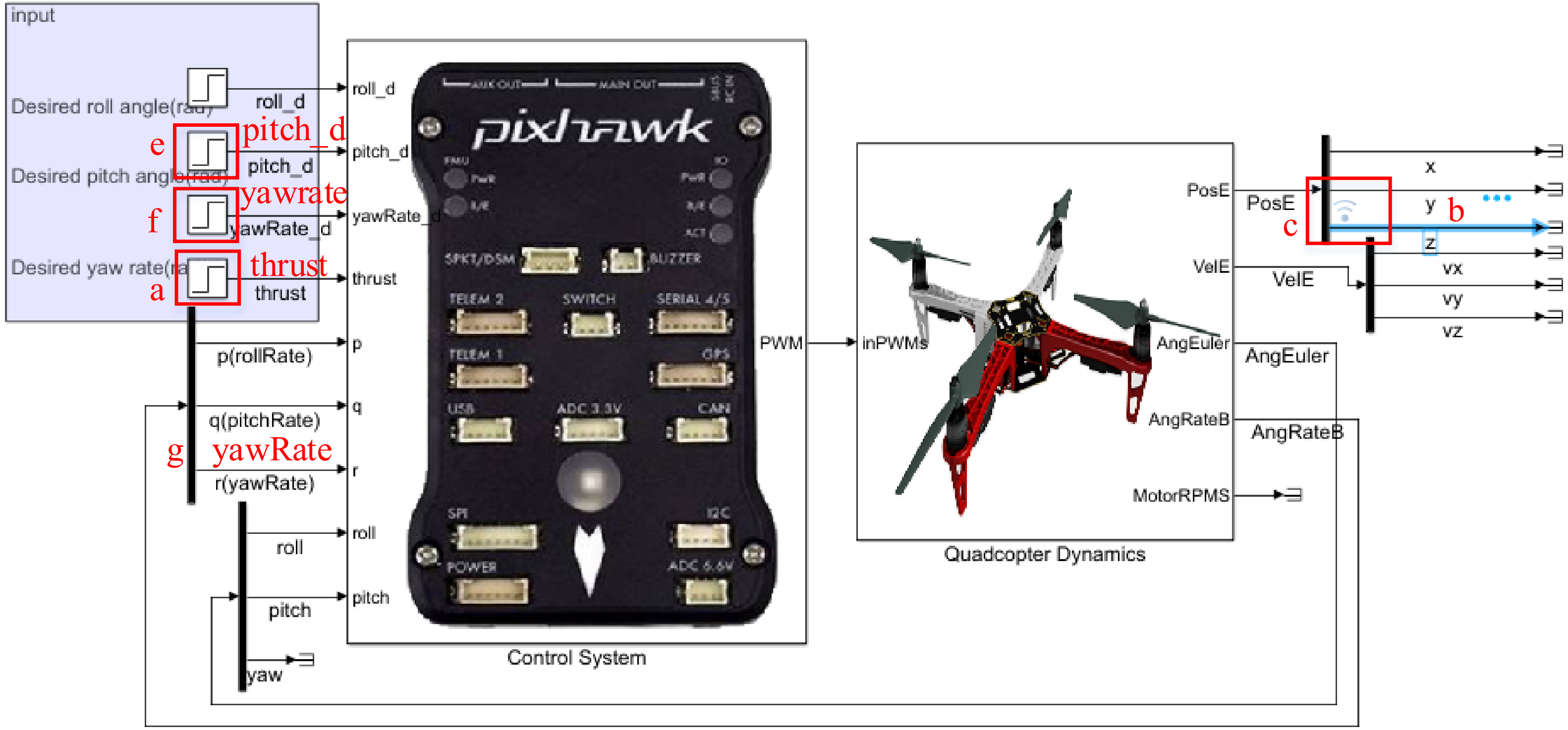


Figure. SIL test for flight state with respect to total mass changing, Simulink model "e2_1.slx"

Basic Experiment

□ Analysis Procedure

（1）Step1：The flight state with respect to the total mass

2）Altitude response with respect to the hover throttle. Select “z” signal line on “PosE” output as “Enable Data Logging”. When the mass is 1.4kg (“ModelParam_uavMass” in file “Init_control.m” is set to 1.4) and “Throttle command” is 0.6085 (60.85% throttle percentage) in Simulink, the multicopter can keep hovering. As shown in the right figure.

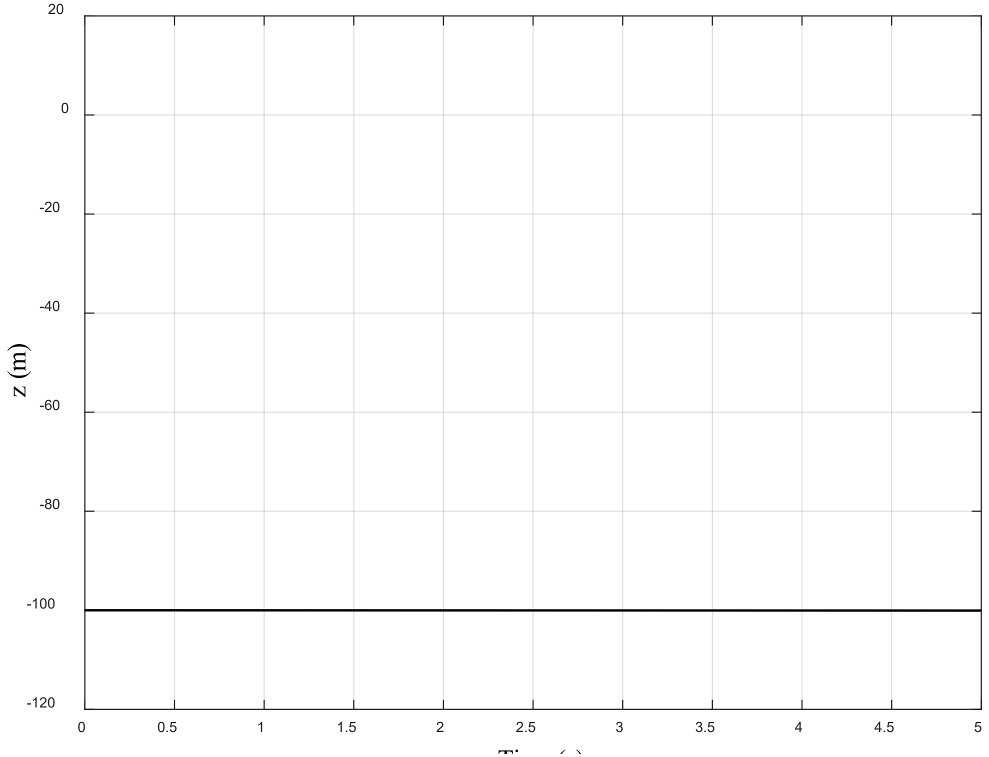


Figure. Altitude response when mass is 1.4kg

Basic Experiment

□ Analysis Procedure

（1）Step1： The flight state with respect to the total mass

3）Altitude response with respect to mass.

Change “ModelParam_uavMass” to 2.0, namely the mass is changed to 2.0 kg. As shown in the right figure, the conclusion is that the altitude of the multicopter decreases when “Throttle command” is constant. Given the increased weight, the same input throttle is unable to provide sufficient thrust to keep the multicopter hovering.When the mass is 2 kg, the “Throttle command” is set to 0.7032 (70.32% throttle percentage) to keep the multicopter hovering.

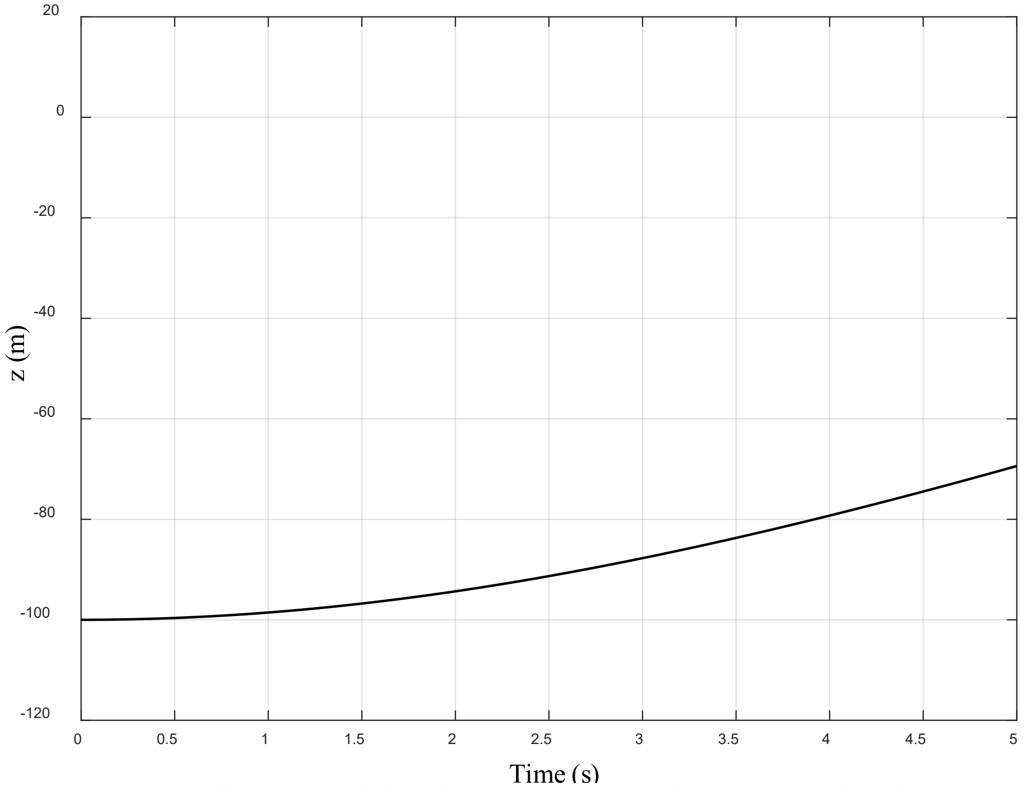


Figure. Altitude response when mass is 2kg

Basic Experiment

（1）Step1： The flight state with respect to the total mass

4）Attitude control performance with respect to mass.

In file “Init_control.m”, the mass is set to 1.4kg or 2.0kg. And, correspondingly, the “Throttle command” is set to 0.6085 or 0.7032. Besides that, “pitch_d” is set to 0.2rad and the output pitch angle is observed with “Scope” in Simulink.

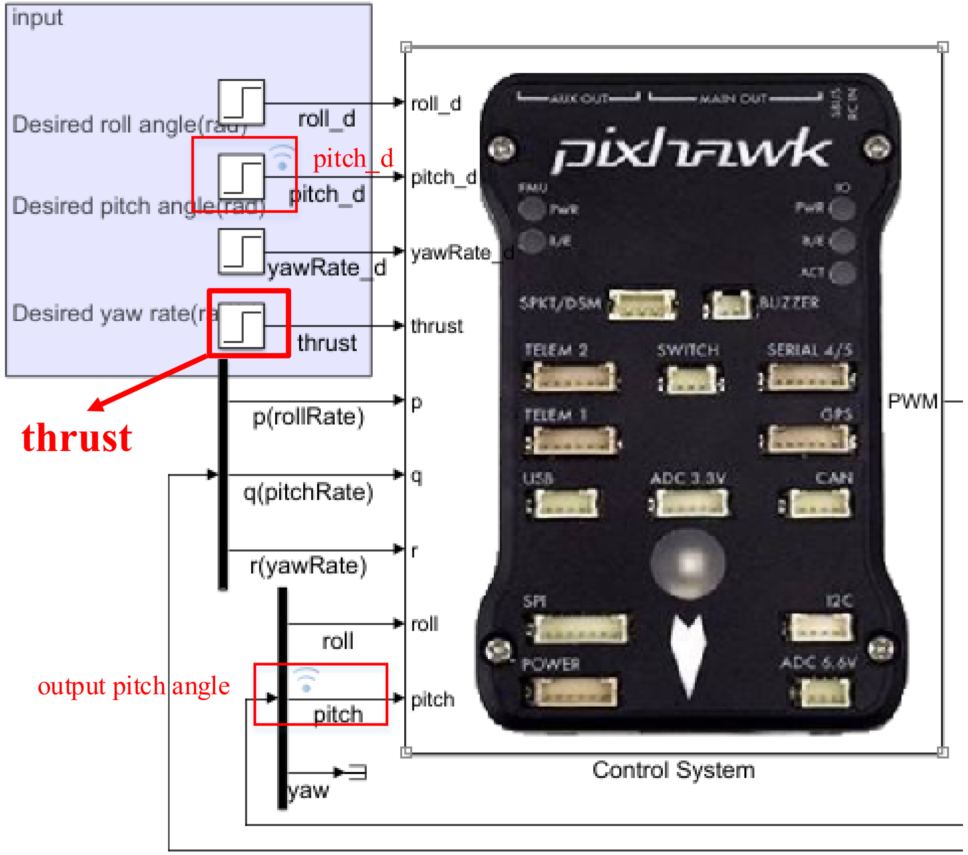


Figure. Set input and observation signal in Simulink model “e2_1.slx”

Basic Experiment

（1）Step1：The flight state with respect to the total mass

4）Attitude control performance with respect to mass

Run the Simulink model to obtain the results shown in right figure. A conclusion is drawn that the attitude response is almost independent of the mass when the remaining parameters such as the moment of inertia are constant.

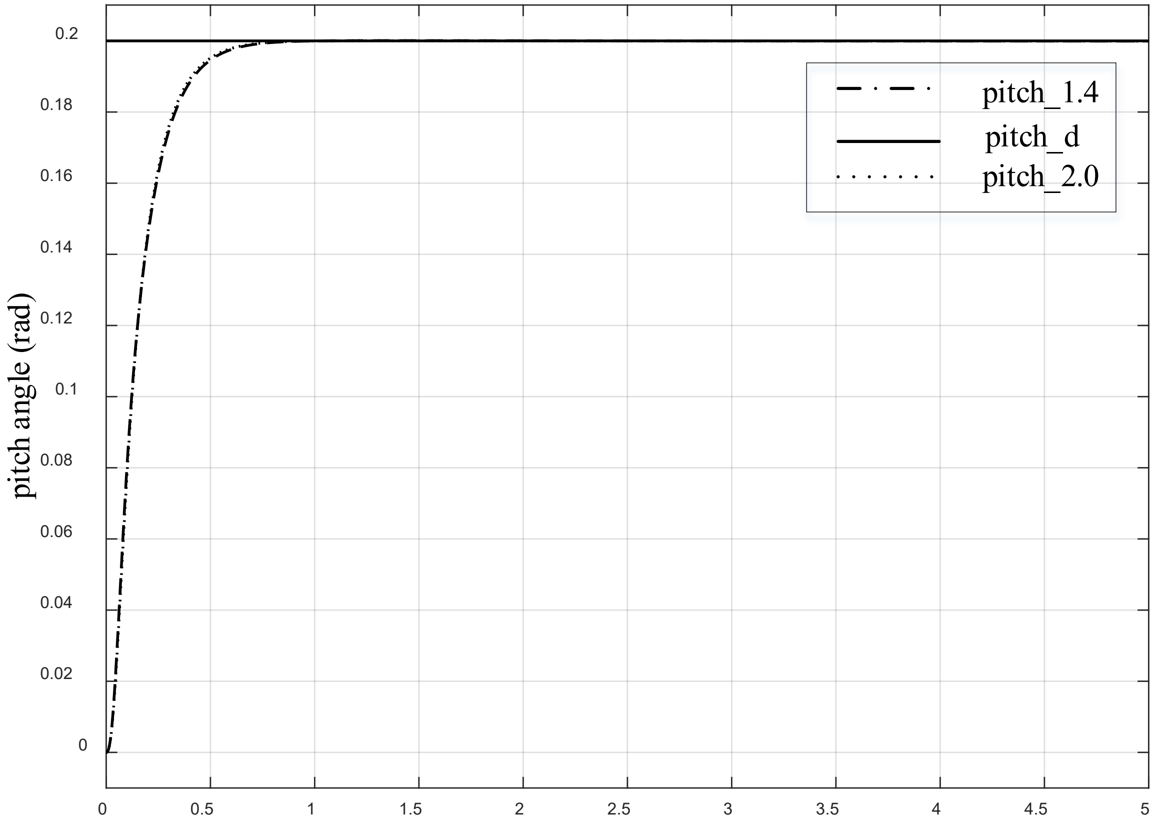


Figure. Pitch angle response with respect to mass

Basic Experiment

（2）Step2：Yaw rate with respect to moment of inertia

Modify “ModelParam_uavJzz” in file “Init_control.m” to double the moment of inertia about the obzb axis (“ModelParam_uavJzz=0.0732”), and the results shown in the right figure.

Parameter “yawrate_d” denotes the desired yaw rate, “r_1” denotes the yaw rate when “ModelParam_uavJzz” is the initial value (“ModelParam_uavJzz=0.0366”), and “r_2” denotes the yaw rate when “ModelParam_uavJzz” is doubled (“ModelParam_uavJzz=0.0732”).

The conclusion is drawn that the system yaw rate response becomes slower when the moment of inertia about the obzb axis increases.

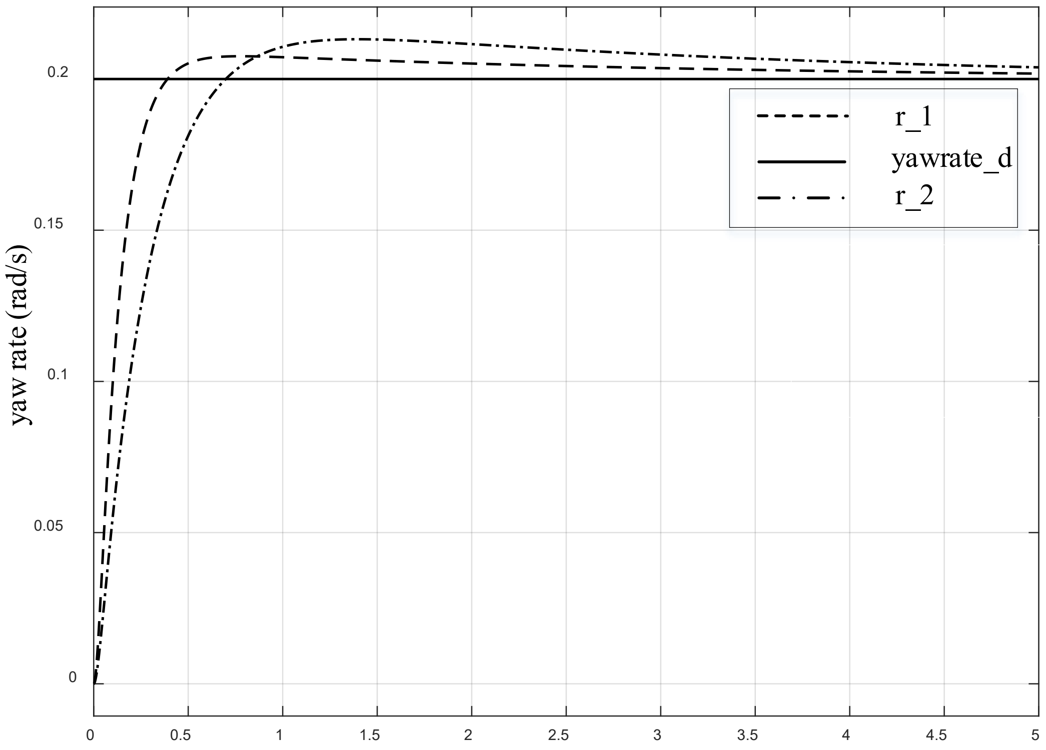


Figure. Yawrate response with respect to the moment of inertia about the o_bz_b axis.

Basic Experiment

（3）Step3：Altitude with respect to propeller thrust coefficient

The propeller thrust coefficient “ModelParam_rotorCt” is doubled (“ModelParam_rotorCt=2.21e-05”) and other parameters correspond to the initial value.

Evidently, with the propeller thrust parameter increased, the thrust provided by propellers is increased under the same throttle command. The altitude response is shown right figure. In order to maintain the multicopter hovering, the “Throttle command” is set to 0.3042 (30.42% throttle percentage).

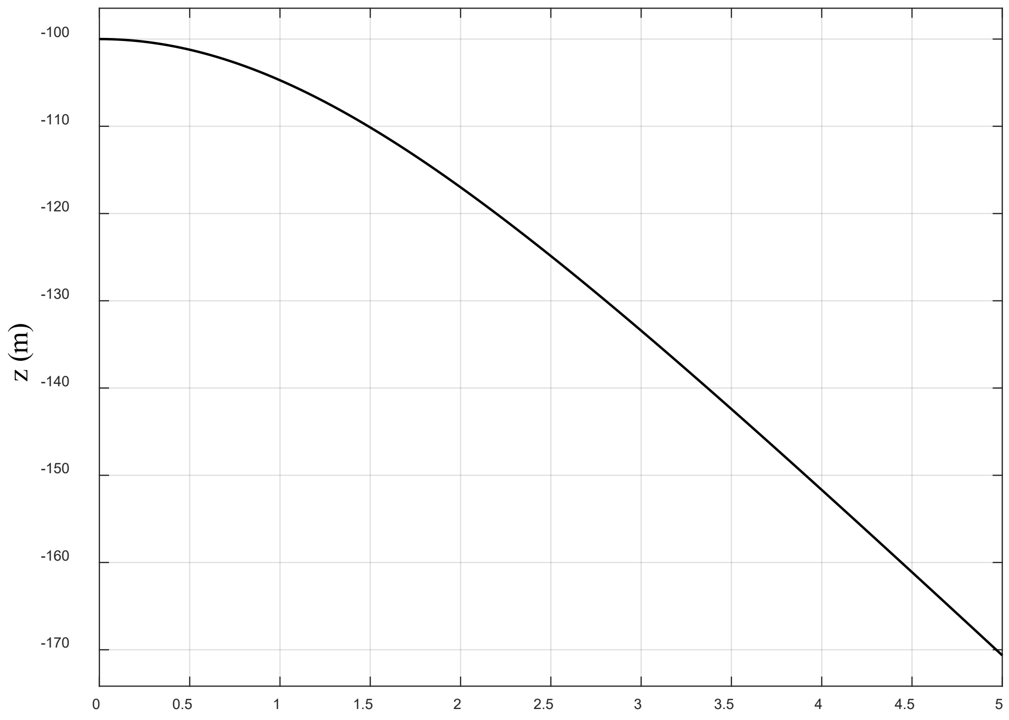


Figure. Altitude response with respect to propeller thrust coefficient

Basic Experiment

（4）Step4：Yaw rate with respect to propeller torque

The result of yaw rate output is shown in the right figure. Parameter “yawrate_d” denotes the desired yaw rate, “r_1” denotes the yaw rate when “ModelParam.rotorCm” is the initial value (“ModelParam.rotorCm=1.779e-07”), and “r_2” denotes the yaw rate when “ModelParam.rotorCm” is doubled (“ModelParam.rotorCm=3.558e-07”).

It is concluded that the larger torque coefficient is, the faster yaw rate response is.

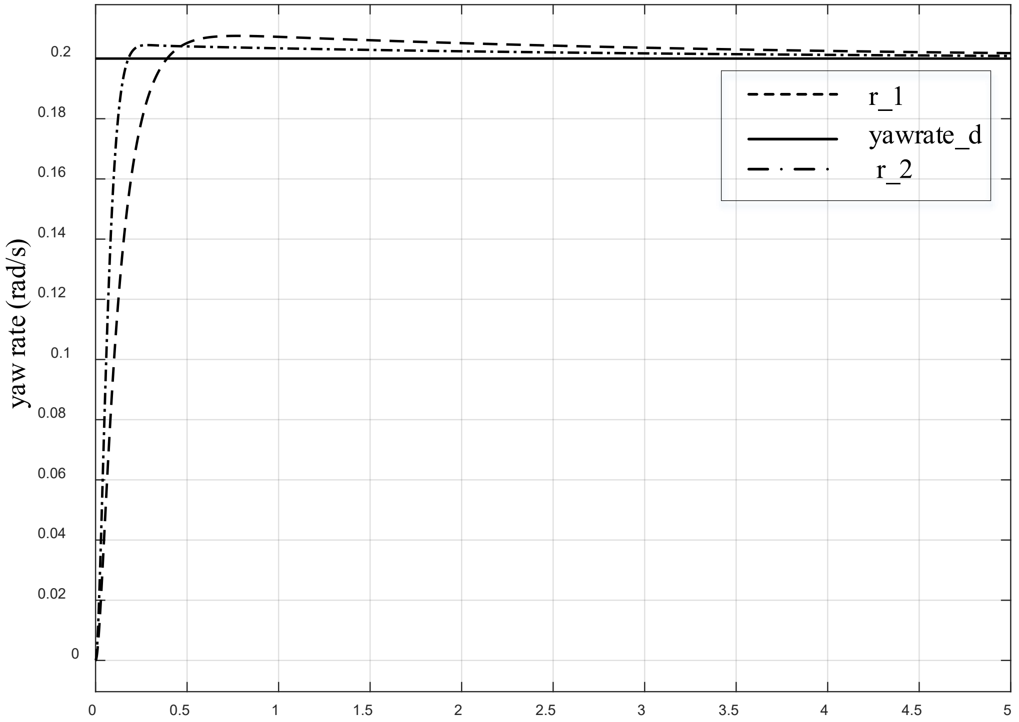


Figure. Yaw rate response with respect to the torque coefficient.

Basic Experiment

☐ Remarks

When the altitude changes, change the value of the throttle to hover and observe the pitch response of the multicopter.

Analysis Experiment

□ Experimental Objective

When an X-configuration quadcopter is hovering, calculate the equilibrium point of the dynamic system in the following

$$\begin{cases} {}^{e}\dot{\mathbf{p}} = {}^{e}\mathbf{v} \\ {}^{e}\dot{\mathbf{v}} = {}^{e}\mathbf{F}/m \\ \dot{\mathbf{\Theta}} = \mathbf{W} \cdot {}^{b}\boldsymbol{\omega} \\ \mathbf{J} \cdot {}^{b}\dot{\boldsymbol{\omega}} = -{}^{b}\boldsymbol{\omega} \times (\mathbf{J} \cdot {}^{b}\boldsymbol{\omega}) + {}^{b}\mathbf{M} \end{cases}$$

We express the linearization model at the equilibrium point by considering the motor dynamics. Subsequently, compare and analyze the conclusions from the basic experiment and the analysis experiment.

Analysis Experiment

□ Calculation Procedure

（1）Step1： Calculate the hover throttle command

When a multicopter is hovering, the weight satisfies the following expression

$$\varpi^* = \sqrt{\frac{mg}{4c_T}} = \sqrt{\frac{1.4\times 9.8}{4\times 1.105\text{e-}05}} \approx 557.14\text{RPM}$$

it further obtains the following expression

$$\sigma^* = \frac{\varpi - \varpi_b}{C_R} = \frac{557.14-(-141.4)}{1148} = 0.6085$$

thereby the hover throttle command is 0.6085.

Analysis Experiment

Calculation Procedure

（2）Step2： Calculate the linearization model of the equilibrium point

1）Linear model simplification

For multicopters in this experiment, assuming ${}^{\mathrm{e}}\mathrm{v} \approx 0,\ {}^{\mathrm{b}}\omega \approx 0$ and ${}^{\mathrm{e}}\mathrm{w} \approx 0$ with the small perturbation hypothesis around the hovering mode, one has

$$
\begin{aligned}
-{}^{\mathrm{b}}\omega \times \left(\mathbf{J}\,{}^{\mathrm{b}}\omega\right) &\approx \mathbf{0} \\
\mathbf{G}_{\mathrm{a}} &\approx \mathbf{0} \\
{}^{\mathrm{b}}\mathbf{M}_{\mathrm{d}} &\approx \mathbf{0} \\
{}^{\mathrm{b}}\mathbf{F}_{\mathrm{d}} &\approx \mathbf{0}
\end{aligned}
$$

The simplified model at the equilibrium point is

$$
\begin{cases}
{}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} \\
{}^{\mathrm{e}}\dot{\mathbf{v}} = {}^{\mathrm{e}}\mathbf{F}/m \\
\dot{\mathbf{\Theta}} = \mathbf{W}\cdot{}^{\mathrm{b}}\boldsymbol{\omega} \\
\mathbf{J}\cdot{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = -{}^{\mathrm{b}}\boldsymbol{\omega}\times(\mathbf{J}\cdot{}^{\mathrm{b}}\boldsymbol{\omega}) + {}^{\mathrm{b}}\mathbf{M}
\end{cases}
$$

⇓

$$
\begin{cases}
{}^{\mathrm{e}}\dot{\mathbf{p}} = {}^{\mathrm{e}}\mathbf{v} \\
{}^{\mathrm{e}}\dot{\mathbf{v}} = g\mathbf{e}_3 - \dfrac{f}{m}\mathbf{R}\mathbf{e}_3 \\
\dot{\mathbf{\Theta}} = \mathbf{W}\cdot{}^{\mathrm{b}}\boldsymbol{\omega} \\
\mathbf{J}\cdot{}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}
\end{cases}
$$

Analysis Experiment

□ Calculation Procedure

Given that the pitch and roll angles at the equilibrium point are both very small, and the total thrust is approximately equal to the gravity of the multicopte, the assumptions are further expressed as

$$\sin\phi \approx \phi, \cos\phi \approx 1, \sin\theta \approx \theta, \cos\theta \approx 1, \boldsymbol{\tau} \approx \mathbf{0}.$$

Thus, $\mathbf{Re}_3$ is simplified as follows

$$\mathbf{Re}_3 \approx \begin{bmatrix} \theta\cos\psi + \theta\sin\psi \\ \theta\sin\psi - \phi\cos\psi \\ 1 \end{bmatrix}$$

So, the original model is decoupled into three linear models, namely horizontal position channel model, altitude channel model, and attitude model. They are introduced as follows.

Analysis Experiment

□ Calculation Procedure

（a）Horizontal position channel model

$$\dot{\mathbf{p}}_{\mathrm{h}} = \mathbf{v}_{\mathrm{h}}$$
$$\dot{\mathbf{v}}_{\mathrm{h}} = -g\mathbf{A}_{\psi}\mathbf{\Theta}_{\mathrm{h}}$$

where $\mathbf{p}_{\mathrm{h}} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$, $\mathbf{A}_{\psi} = \begin{bmatrix} \sin\psi & \cos\psi \\ -\cos\psi & \sin\psi \end{bmatrix}$, $\mathbf{\Theta}_{\mathrm{h}} = \begin{bmatrix} \phi \\ \theta \end{bmatrix}$.

In the horizontal position channel model, $\mathbf{\Theta}_{\mathrm{h}}$ is viewed as the input. Furthermore, $-g\mathbf{A}_{\psi}$ can be obtained. So, $-g\mathbf{A}_{\psi}\mathbf{\Theta}_{\mathrm{h}}$ can be viewed as the input and $\mathbf{p}_{\mathrm{h}}$ is viewed as the output.

（b）Altitude channel model

$$\dot{p}_{\mathrm{z}} = v_{\mathrm{z}}$$
$$\dot{v}_{\mathrm{z}} = g - \frac{f}{m}$$

（c）Attitude model

$$\dot{\mathbf{\Theta}} = {}^{\mathrm{b}}\boldsymbol{\omega}$$
$$\mathbf{J} \cdot {}^{\mathrm{b}}\dot{\boldsymbol{\omega}} = \boldsymbol{\tau}$$

Analysis Experiment

□ Calculation Procedure

1）Hover state linearization

When a multicopter is hovering, the angular speed of the ith propeller at the equilibrium point is $\varpi_i^* = \varpi^*$, throttle command at the equilibrium point is $\sigma_i^* = \sigma^*$, reaction torque at the equilibrium point is , $M_i^* = M^*$ $i = 1,2,3,4$ Furthermore, the attitude angle and speed at the equilibrium point are zero.

Any variable can be decomposed into a sum of the value at the equilibrium point and the perturbation in the form as follows

$$\mathbf{\Theta} = \mathbf{0} + \Delta\mathbf{\Theta}$$

$$\boldsymbol{\omega} = \mathbf{0} + \Delta\boldsymbol{\omega}$$

$$\varpi_i = \varpi^* + \Delta\varpi_i$$

$$\sigma_i = \sigma^* + \Delta\sigma_i$$

$$M_i = M^* + \Delta M_i$$

$$T_i = T^* + \Delta T_i$$

where $\Delta\mathbf{\Theta}, \Delta\boldsymbol{\omega}$ are the changes of the Euler angle and speed; $\Delta\varpi_i, \Delta\sigma_i, \Delta M_i, \Delta T_i$ are the perturbations of propeller angular speed, throttle command, reaction torque, and thrust respectively.

Analysis Experiment

Calculation Procedure

According to propulsor model, we can get $\Delta\varpi_i = \frac{1}{T_{\mathrm{m}}s+1}C_{\mathrm{R}}\Delta\sigma_i$

Based on Newton's third law, the reaction torque is as large as the torque acting on the ith propeller. Subsequently, this is expressed as

$$M_i = c_{\mathrm{M}}\varpi_i^2 + J_{\mathrm{RP}}\dot{\varpi}_i$$

When a multicopter hovers without wind, the propeller thrust is expressed as

$$T_i = c_T\varpi_i^2$$

The perturbation of thrust and the perturbation of reaction torque are

$$\Delta T_i = 2c_{\mathrm{T}}\Delta\varpi_i\varpi^*$$

$$\Delta M_i = 2c_{\mathrm{M}}\varpi^*\Delta\varpi_i + J_{\mathrm{RP}}\Delta\dot{\varpi}_i$$

Furthermore $\Delta T_i = \frac{C_R 2c_{\mathrm{T}}\varpi_0^*}{T_m s+1}\Delta\sigma_i, \Delta M_i = \frac{C_R\left(2c_{\mathrm{M}}\varpi_0^* + J_{\mathrm{RP}}s\right)}{T_m s+1}\Delta\sigma_i$

The perturbation model of thrust and torque can be further written as

$$\Delta f = \frac{C_R 2c_{\mathrm{T}}\varpi_0^*}{T_m s+1}(\Delta\sigma_1+\Delta\sigma_2+\Delta\sigma_3+\Delta\sigma_4)$$

$$\Delta\boldsymbol{\tau}_x = \sqrt{2}d\frac{C_R c_{\mathrm{T}}\varpi_0^*}{T_m s+1}(\Delta\sigma_1-\Delta\sigma_2-\Delta\sigma_3+\Delta\sigma_4)$$

$$\Delta\boldsymbol{\tau}_y = \sqrt{2}d\frac{C_R c_{\mathrm{T}}\varpi_0^*}{T_m s+1}(\Delta\sigma_1+\Delta\sigma_2-\Delta\sigma_3-\Delta\sigma_4)$$

$$\Delta\boldsymbol{\tau}_y = \frac{C_R\left(2c_{\mathrm{M}}\varpi_0^* + J_{\mathrm{RP}}s\right)}{T_m s+1}(\Delta\sigma_1-\Delta\sigma_2+\Delta\sigma_3-\Delta\sigma_4)$$

Analysis Experiment

Calculation Procedure

The design of horizontal position channel model

Horizontal position channel model are linearized at the equilibrium point that

$$\begin{bmatrix}\Delta p_x \\ \Delta p_y\end{bmatrix} = \begin{bmatrix}-\dfrac{g}{s^2}\Delta\theta \\ \dfrac{g}{s^2}\Delta\phi\end{bmatrix}$$

So the transfer function is

$$\Delta p_x = -\sqrt{2}g\frac{dC_R c_T \varpi_0^*}{J_y}\frac{1}{s^4}\frac{1}{T_m s+1}\Delta\overline{\tau}_y$$

$$\Delta p_{y=}\sqrt{2}g\frac{dC_R c_T \varpi_0^*}{J_x}\frac{1}{s^4}\frac{1}{T_m s+1}\Delta\overline{\tau}_x$$

$$\Delta p_z = -\frac{2C_R c_T \varpi_0^*}{ms^2(T_m s+1)}\Delta\overline{\tau}$$

where $\Delta\overline{\tau} = \Delta\sigma_1 + \Delta\sigma_2 + \Delta\sigma_3 + \Delta\sigma_4$

Analysis Experiment

□ Calculation Procedure

(3) Step3： Compare theoretical analysis with conclusion from the basic experiment

1) Analyze the yaw rate response

the transfer function of yaw rate is as follow

$$\Delta\dot{\psi}(s)=\frac{C_R\left(2c_{\mathrm{M}}\varpi_0^{\ *}+J_{\mathrm{RP}}s\right)}{J_z}\frac{1}{s}\frac{1}{T_m s+1}\Delta\overline{\tau}_z$$

It is concluded that, with the slope of the linear relationship from the throttle command to the motor speed C_{R} , the torque coefficient c_M , and torque speed ϖ_0^* increased, and the yaw angle rate response gets faster. If the motor response time constant T_{m} and the moment of inertia J_z about the $o_b z_b$ axis increase, the yaw angle rate response decreases. These are consistent with the conclusions from Step2 and Step4 in the basic experiment.

Analysis Experiment

◻ Calculation Procedure

2) Analyze the altitude response

The transfer function of the altitude channel model is as follows

$$\Delta p_z = -\frac{2C_R c_T \varpi_0^*}{ms^2(T_m s+1)}\Delta\overline{\tau}$$

It is concluded that with the propeller thrust coefficient c_T and the slope of the linear relationship from the throttle command to the motor speed C_R increased, the altitude gets higher. If the mass m is increased, the altitude response gets slower. The conclusions can be verified by readers yourselves with corresponding simulation experiments.

Experimental Objective

Things to prepare

Software: MATLAB R2017b or above, a designed multicopter model in lesson 5, the model parameters provided on the multicopter performance evaluation website https://flyeval.com/paper/.

Objectives

1) Establish a control model for the designed quadcopter with its attitude control model based on quaternions, rotation matrix or Euler angles;
2) Add a quadcopter 3D model to FlightGear flight simulator.

Design Experiment

☐ Design Procedure of First Objective

(1) Step1： Propulsor model

For a complete propulsor model

$$\varpi = \frac{1}{T_{\mathrm{m}} s + 1} \varpi_{\mathrm{ss}}$$

Let the state variable $x = T_{\mathrm{M}} \varpi$, output $y = \varpi$ and input $u = \varpi_{\mathrm{ss}}$. Then

$$\begin{cases} \dot{x} = -\dfrac{1}{T_{\mathrm{m}}} x + u \\ y = \dfrac{1}{T_{\mathrm{m}}} x \end{cases}$$

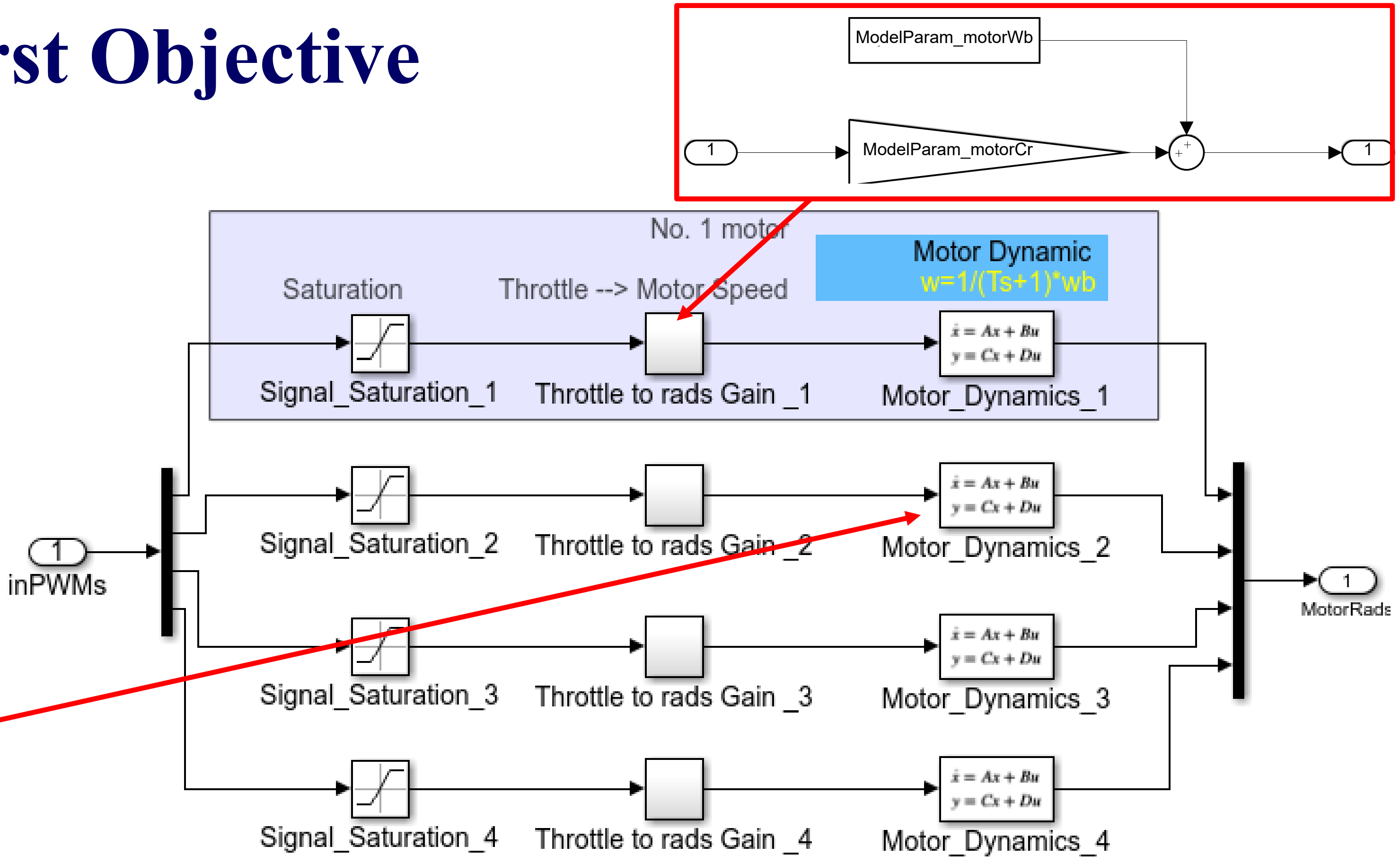


Figure. Propulsor model, Simulink model “dynamics.slx”

Design Experiment

☐ Design Procedure of First Objective

(2) Step2： Control effectiveness model

Based on equations

$$f=\sum_{i=1}^{4}T_i=c_{\mathrm{T}}\left(\varpi_1^2+\varpi_2^2+\varpi_3^2+\varpi_4^2\right)$$

$$\tau_x=dc_{\mathrm{T}}\left(-\frac{\sqrt{2}}{2}\varpi_1^2+\frac{\sqrt{2}}{2}\varpi_2^2+\frac{\sqrt{2}}{2}\varpi_3^2-\frac{\sqrt{2}}{2}\varpi_4^2\right)$$

$$\tau_y=dc_{\mathrm{T}}\left(\frac{\sqrt{2}}{2}\varpi_1^2-\frac{\sqrt{2}}{2}\varpi_2^2+\frac{\sqrt{2}}{2}\varpi_3^2-\frac{\sqrt{2}}{2}\varpi_4^2\right)$$

$$\tau_z=c_{\mathrm{M}}\left(\varpi_1^2+\varpi_2^2-\varpi_3^2+\varpi_4^2\right)$$

the force and moment of the propeller acting on the body are obtained; and The control effectiveness model is shown in the right figure.

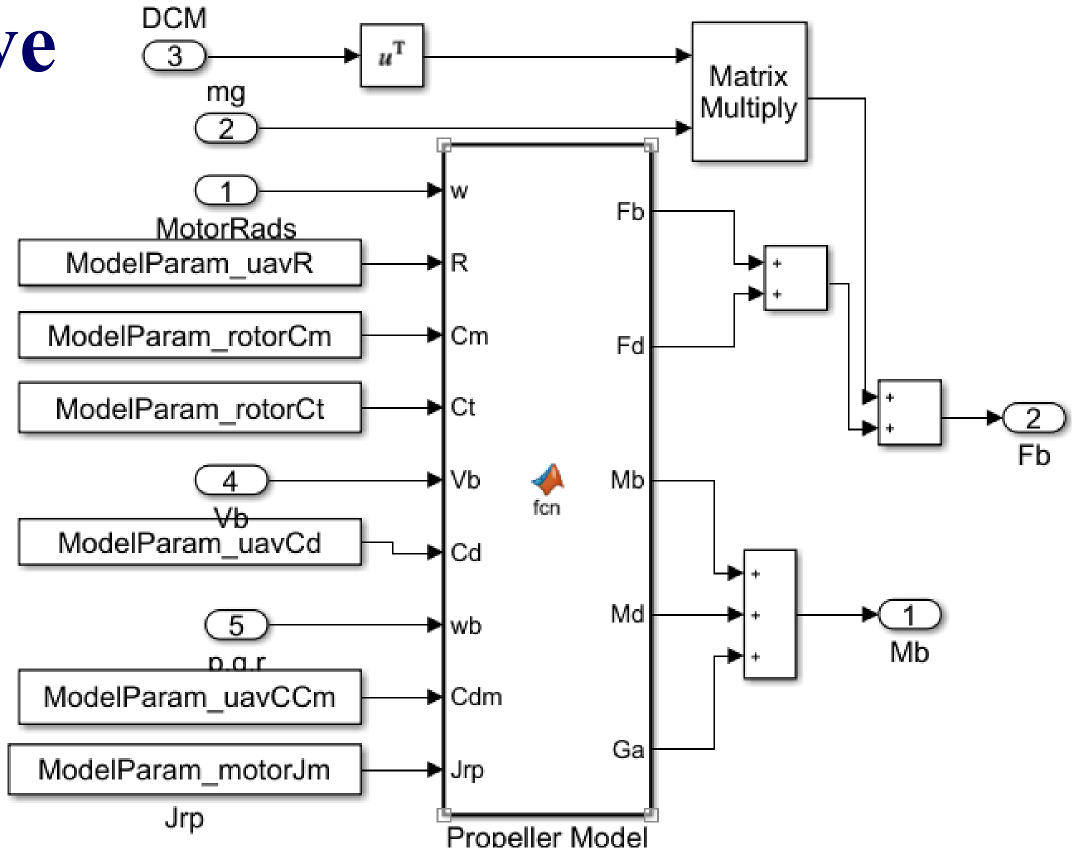


Figure. Control effectiveness model, Simulink model “dynamics.slx”

Design Experiment

□ Design Procedure of First Objective

(3) Step3: Rigid-body dynamic model

Based on equations

$$\mathbf{J}\cdot {}^{b}\dot{\boldsymbol{\omega}} = -{}^{b}\boldsymbol{\omega}\times\left(\mathbf{J}\cdot{}^{b}\boldsymbol{\omega}\right) + {}^{b}\mathbf{M}$$

$${}^{b}\dot{\mathbf{v}} = -\left[{}^{b}\boldsymbol{\omega}\right]_{\times} {}^{b}\mathbf{v} + {}^{b}\mathbf{F}/m$$

The position dynamic model and attitude dynamic model as shown in the right figure.

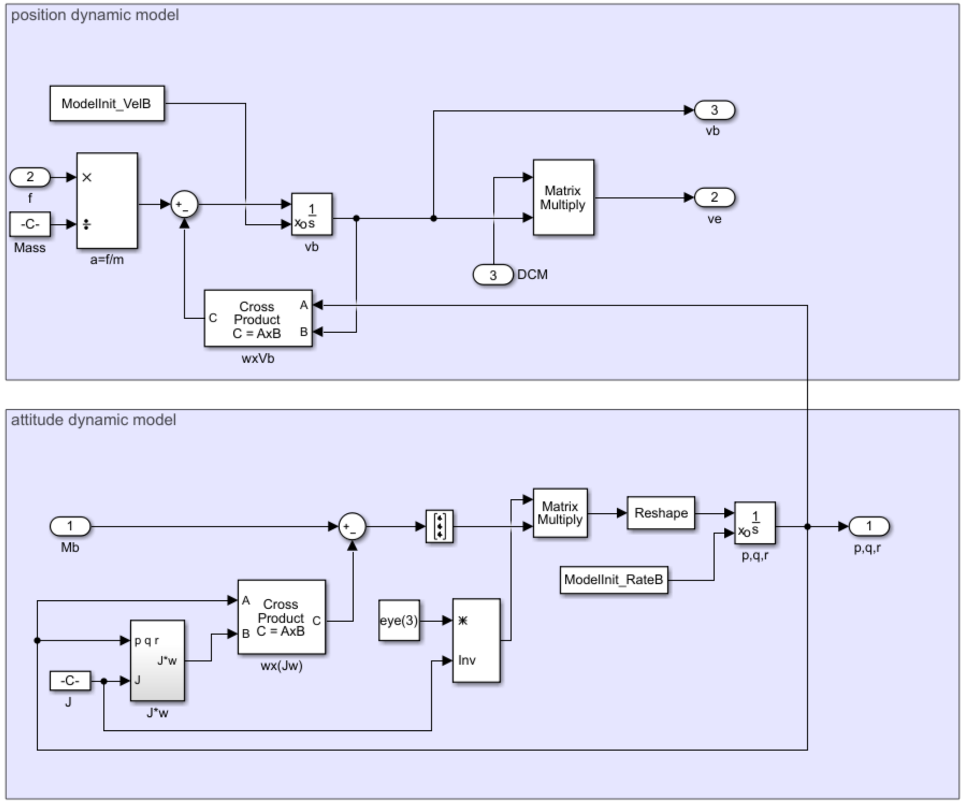


Figure. Position dynamic and attitude dynamic model , Simulink model “dynamics.slx”

Design Experiment

□ Design Procedure of First Objective

（4）Step4: Rigid-body kinematic model

Based on equations

$$^{e}\dot{\mathbf{p}} = {}^{e}\mathbf{v}$$

$$\dot{q}_0 = -\frac{1}{2}\mathbf{q}_v^T \cdot {}^{b}\boldsymbol{\omega}$$

$$\dot{\mathbf{q}}_v = \frac{1}{2}\left(q_0\mathbf{I}_3 + [\mathbf{q}_v]_\times\right){}^{b}\boldsymbol{\omega}$$

the kinematic model is established as shown in the right figure.

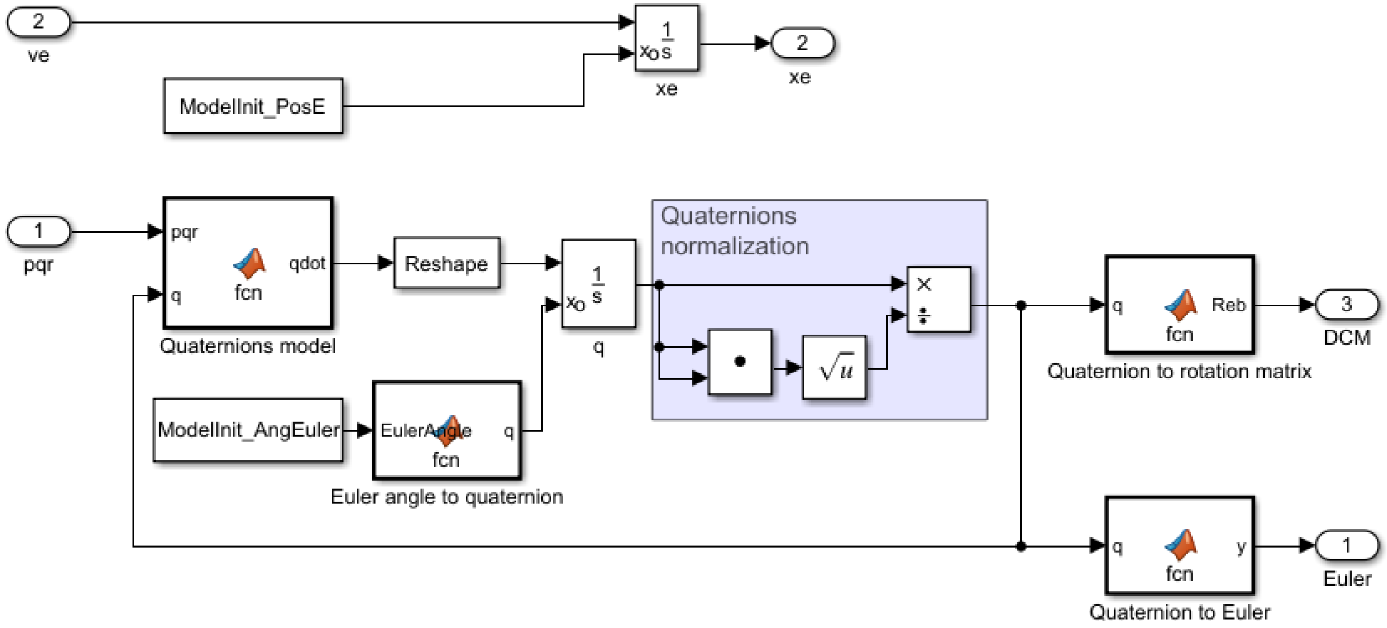


Figure. Attitude kinematic model, Simulink model “dynamics.slx”

Design Experiment

□ Design Procedure of First Objective

(5) Step5：Model connection

The connection between kinematics model and dynamic model is shown in the right figure, which consists of the multicopter flight control rigid-body model.

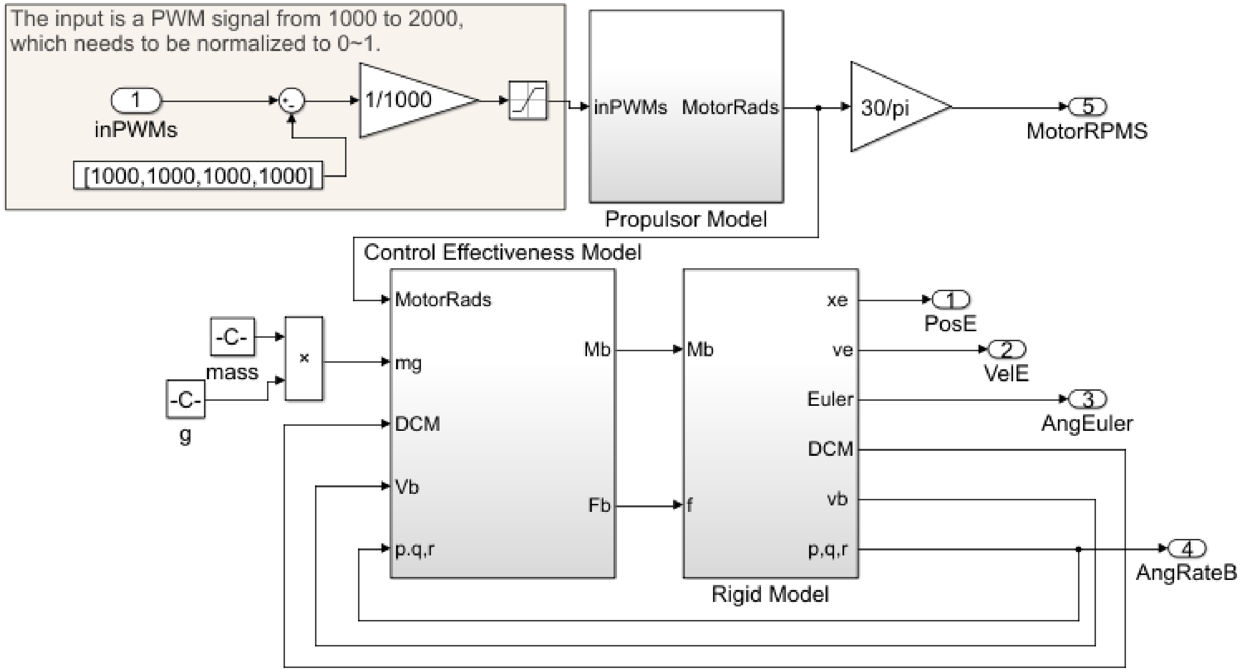


Figure. Quadcopter dynamics model, Simulink model “dynamics.slx”

Design Experiment

□ Design Procedure of Second Objective

（1）Step1： Prepare a quadcopter 3D model

- A simple quadcopter 3D model drawn by AC3D software is shown in the following figure.

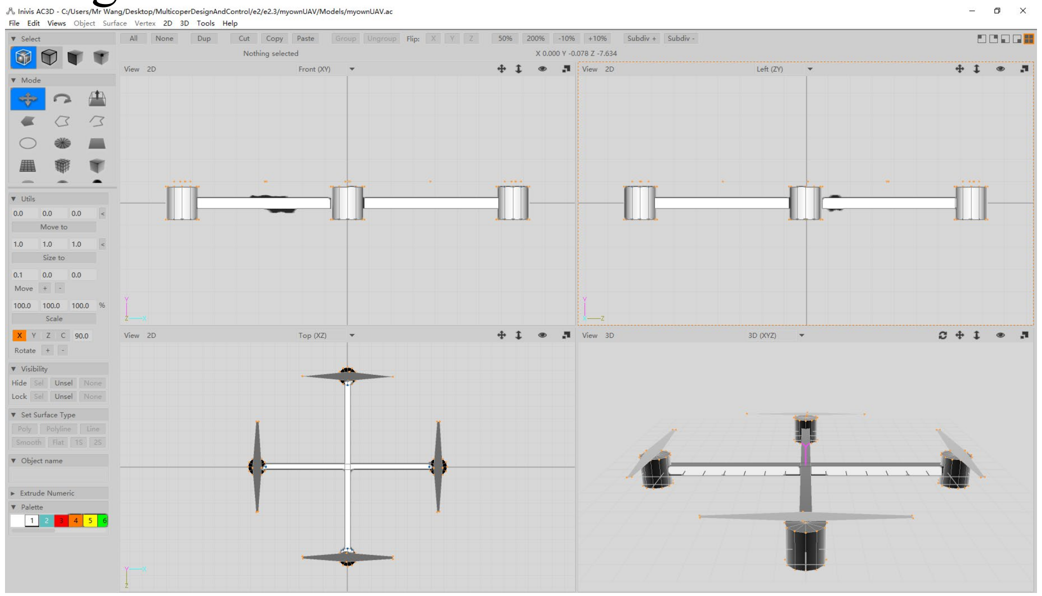


Figure. A simple quadcopter 3D model

- The parameters of the four rotors of the quadcopter model are shown in the following table.

Table. Actual coordinates of each propellers

Propeller Name	Location attribute		
	X	Y	Z
propeller1	0	7.5	1
propeller2	7.5	0	1
propeller3	0	-7.5	1
propeller4	-7.5	0	1

Design Experiment

□ Design Procedure of Second Objective

（2）Step2：Parameter configuration

The type of the main configuration file in FlightGear is XML, such as “myownUAV-set.xml”and myownUAV.xml in the design experiment.

myownUAV-set.xml

```
<?xml version="1.0"?>
<PropertyList>
 <sim>
  <description>myownUAV</description>
  <flight-model>network</flight-model>
  <model>
    <path>Aircraft/myownUAV/Models/myownUAV.xml</path>
  </model>
<chase-distance-m type="double"> -40</chase-distance-m>
    <current-view>
  <view-number type="int">2</view-number>
    </current-view>
 </sim>
</PropertyList>
```

myownUAV.xml

```
<?xml version="1.0"?>
<PropertyList>
 <path>myownUAV.ac</path>
<animation>
   <type>spin</type>
   <object-name>propeller1</object-name>
   <property>/engines/engine[0]/rpm</property>
   <factor>-1</factor>
   <center>
     <x-m>0</x-m>
     <y-m>7.5</y-m>
     <z-m>1</z-m>
   </center>
   <axis>
     <x>0.0</x>
     <y>0.0</y>
     <z>1.0</z>
   </axis>
</animation>
...
</PropertyList>
```

Design Experiment

□ Design Procedure of Second Objective

（3）Step3：Configuration files

New a secondary directory “myownUAV”

Models	2019/2/20 20:11	文件夹	
myownUAV-set.xml	2018/12/14 15:39	XML 文档	1 KB

myownUAV.ac	2018/10/11 15:45	AC3D geometry	14 KB
myownUAV.xml	2018/12/14 15:11	XML 文档	2 KB

Copy the file to the file “\data\Aircraft”.

Design Experiment

□ Design Procedure of Second Objective

（4）Step4：Drive FlightGear by MATLAB

Prior to running FlightGear, double-click the module “Generate Run Script” and, then open and set the script, which includes name, FlightGear position, model name, port, flight airport background and others. Subsequently, click “Generate Script” to generate a script in the current workspace of MATLAB. Open the script with a text editor and make the following changes.

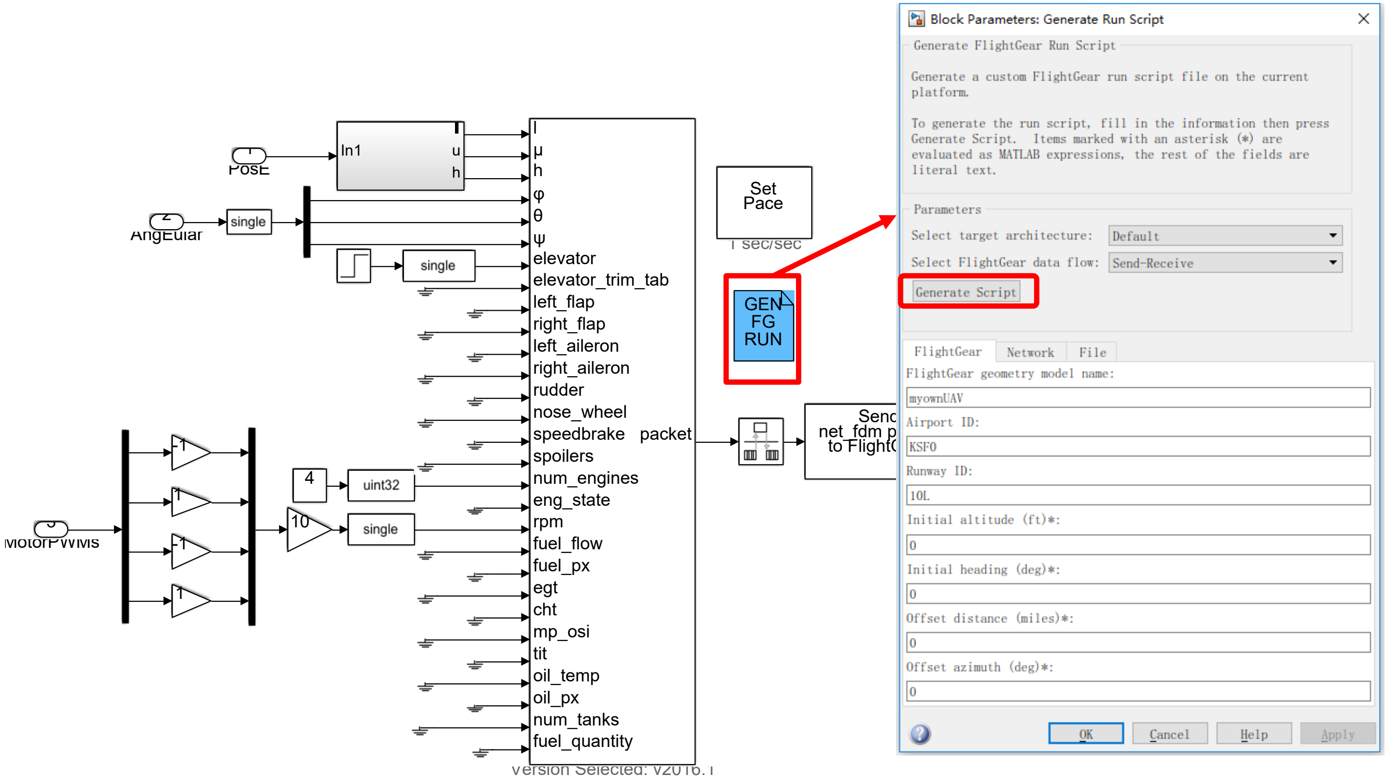


Figure. Data interface, Simulink model “dynamics.slx”

Design Experiment

□ Design Procedure of Second Objective

（4）Step4：Drive FlightGear by MATLAB

1) The “time” following “-start-date-lat” in the script is changed to 2004:06:01:01:00:00

2) Find “freeze” and then change “enable” to “disable”. Modify and save the script. In order todrive FlightGear by MATLAB, run the script and then click “Run” button in the Simulink to run the Simulink model.

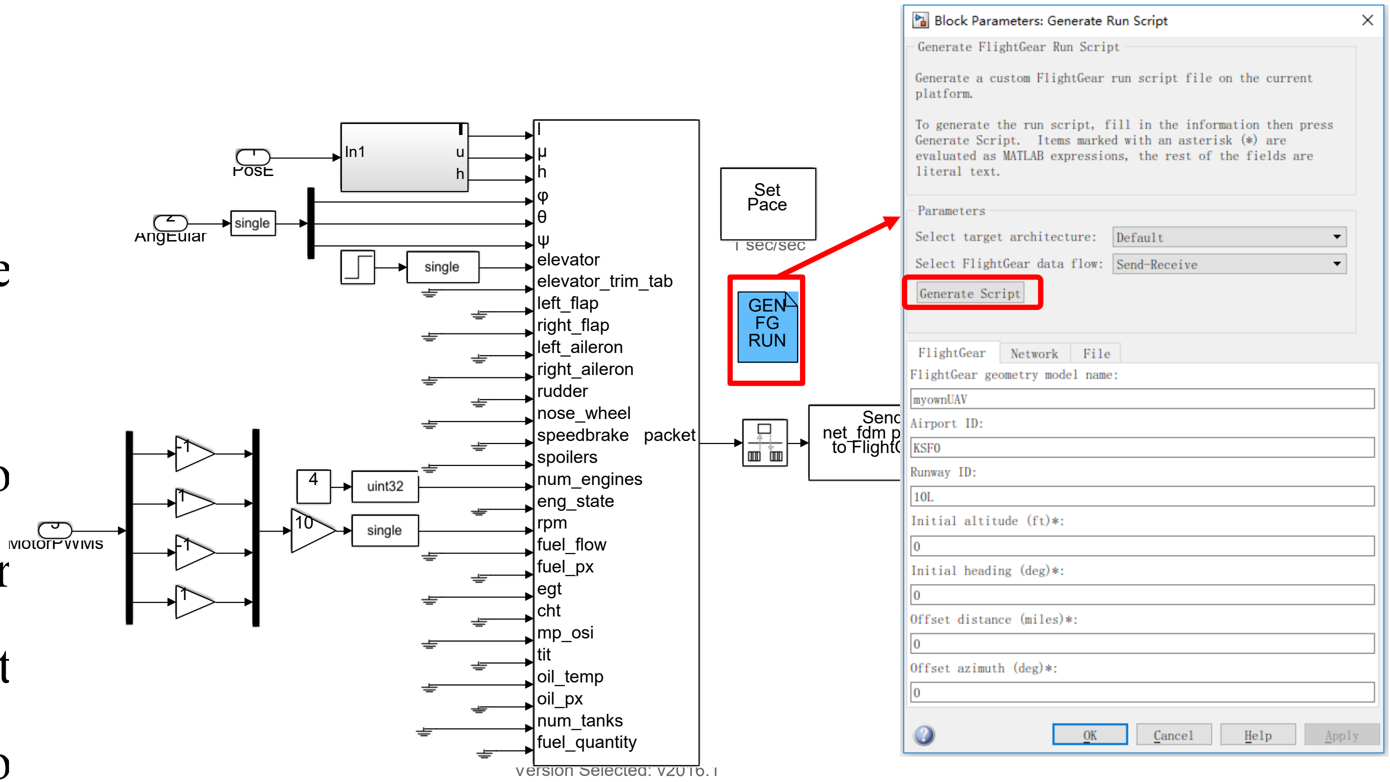


Figure. SIL test for data driving, Simulink model “dynamics.slx”

Design Experiment

☐ Remark

If the background in FlightGear is dim or if need to adjust the view position, then readers should set up the background in FlightGear. As shown in the right figure, readers can set environment time: select "Environment" - "Time Setting" - "noon".

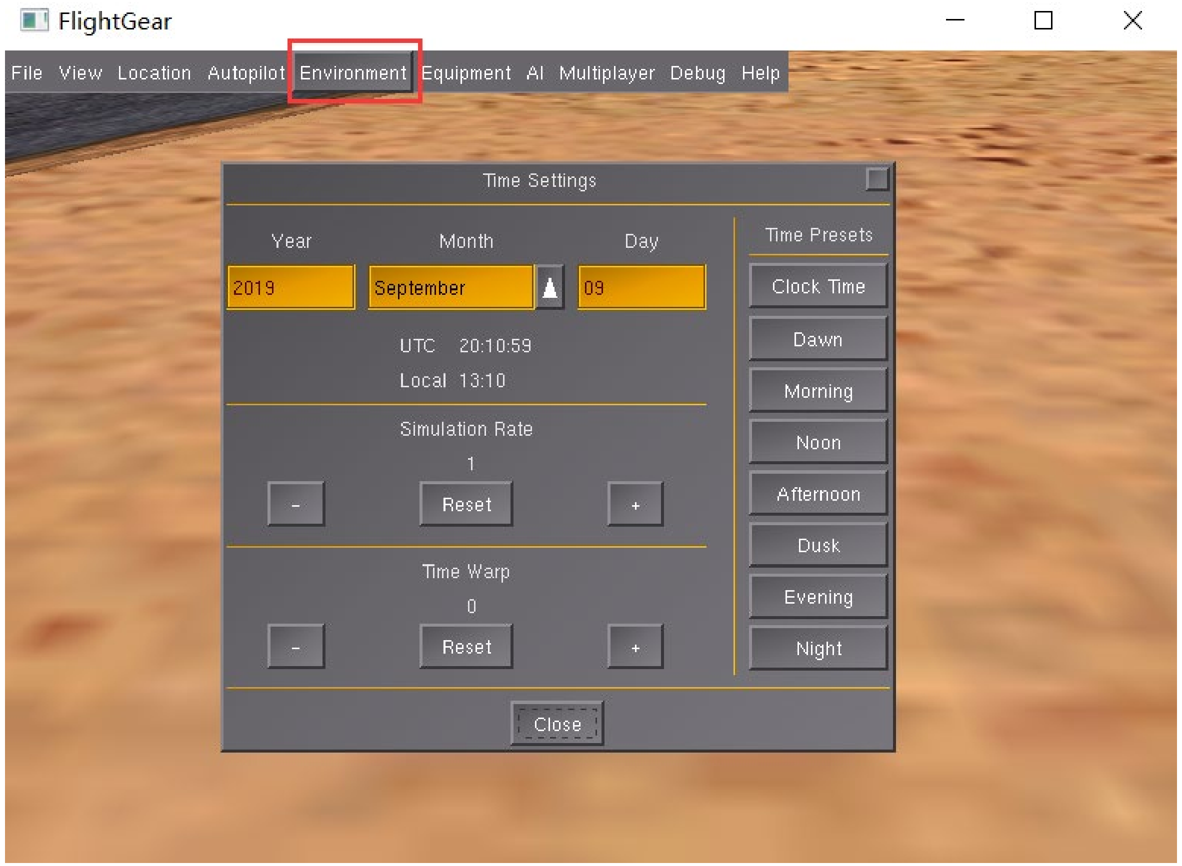


Figure. Setting environment time in FlightGear flight simulator

Design Experiment

☐ Remark

Select "View" - "Adjust View Position", as shown in the right figure. Adjust the angle and distance of the observation by adjusting the three markers, namely "Left/Right", "Down/Up", and "Fwd/Back".

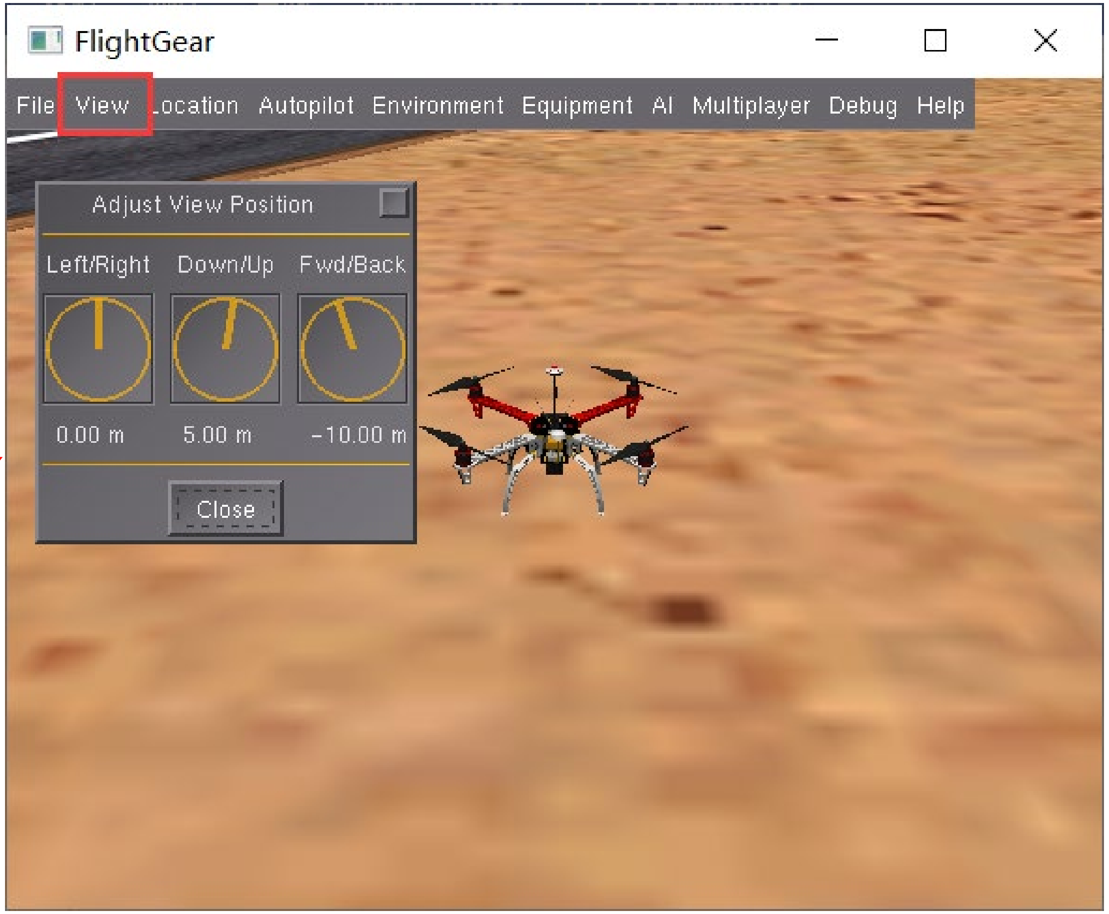


Figure. Setting view position in FlightGear flight simulator

Summary

(1) The flight control rigid model of multicopters includes rigid-body kinematic model, rigid-body dynamic model, control effectiveness model, and propulsor model.

(2) Kinematics is not directly related to mass and force, in which the inputs are only speed and angular velocity and the outputs include position and attitude. Dynamics modeling involves both force in ABCF and motions in EFCF based on Newton's second law and Euler's equations.

(3) At the equilibrium point where a multicopter is hovering, pitch and roll angles are often confined within a small range. In order to simplify the equations in the model, the whole nonlinear model is linearized. In the analysis experiment, yaw angle rate and altitude with respect to different parameters are considered wherein the conclusions are consistent with those from the basic experiment.

(4) The experiments in the following chapters, such as Chapters 9-12, are based on the model established in this chapter.

If you have any question, please go to https://rflysim.com for your information.

Resource

All course PPTs, videos, and source code will be released on our website

https://rflysim.com/en

For more detailed content, please refer to the textbook:

Quan Quan, Xunhua Dai, Shuai Wang. *Multicopter Design and Control Practice*. Springer, 2020

https://www.springer.com/us/book/9789811531378

If you encounter any problems, please post question at Github page

https://github.com/RflySim/RflyExpCode/issues

If you are interested in RflySim advanced platform and courses for rapid development and testing of UAV Swarm/Vision/AI algorithms, please visit:

https://rflysim.com/en/4_Pro/Advanced.html

Thanks