[AE450] Lec10: Simulating Quad(+)copter Dynamics and PID Attitude Controller

I. Disclaimer

> 1. This document may contain some typo and errors. If you find any problems in this document, please let me know through email SKYnSPACE (skynspace@kaist.ac.kr), or the GitHub page GitHub/SKYnSPACE/ to help us share bettter knowledge.
> 2. Every content in this document is not following the academic and/or industrial standards. Additional procedures may be required if you need any sort of certifications with this knowledge.
> 3. In no event shall authors or copyright holders of this document be liable for any damages, infringement of intellectual property rights, breach of contract, etc., caused by direct or indirect use of the contents contained in the document and source code. You must be aware that end users are responsible for them.

[SAFETY NOTES]

> - The contents of this document may be directly related to the various hazards caused by mechanical and electrical systems. Please be sure to thoroughly understand the system and safety rules regarding the systems beforehand.
> - In particular, if you have any doubt on your system you built, please be sure to remove the propellers attached to the motors before any tests.

II. Version Control

- [Ver: 1.0 @01/JUN/2018] Published the first edition for 2018' AE Dept. quick tutorial on the drone making. (SKYnSPACE)
- [Ver: 1.1 @13/NOV/2018] Edited the document for 2018' AE450 Flight dynamics and control lecture. (SKYnSPACE)
- [Ver: 1.11 @30/NOV/2019] Corrected typos before the 2019' AE450 Flight dynamics and control lecture. (SKYnSPACE)
- [Ver: 1.12 @15/DEC/2019] Corrected typos regarding k_T and k_Q coefficients. Thank you Timothy :)

III. Nomenclature

- Configurations

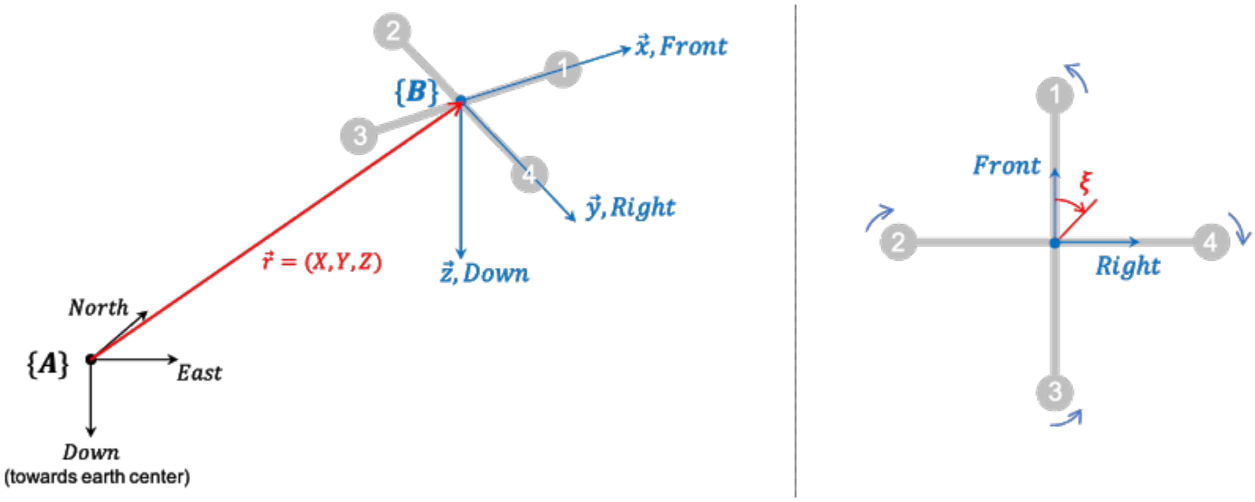

Figure 1.Coordinate Frames and Quad(+) Configuration

- Variables

Expressions	Descriptions	Notes
$\{\overrightarrow{\mathbf{a_1}}, \overrightarrow{\mathbf{a_2}}, \overrightarrow{\mathbf{a_3}}\}$	Set of unit vectors describing the inertial frame $\{\mathbf{A}\}$ $\overrightarrow{\mathbf{a_1}} = (1,0,0)^T, \overrightarrow{\mathbf{a_2}} = (0,1,0)^T, \overrightarrow{\mathbf{a_3}} = (0,0,1)^T$	NED: North East Down Right-hand rule
$\{\overrightarrow{\mathbf{b_1}}, \overrightarrow{\mathbf{b_2}}, \overrightarrow{\mathbf{b_3}}\}$	Set of unit vectors describing the body fixed frame for the airframe $\{\mathbf{B}\}$ expressed in the inertial frame $\{\mathbf{A}\}$, contating attitude of an aircraft.	FRU: Front Right Down Right-hand rule
$\{\vec{\mathbf{x}}, \vec{\mathbf{y}}, \vec{\mathbf{z}}\}$	Set of unit vectors in the airframe $\{\mathbf{B}\}$ itself. $\vec{\mathbf{x}} = (1,0,0)^T, \vec{\mathbf{y}} = (0,1,0)^T, \vec{\mathbf{z}} = (0,0,1)^T$	FRU: Front Right Down Right-hand rule
(X,Y,Z)	Location of an aircraft expressed in the inertial frame [1]. Mostly for the navigational purposes (e.g. path planning, guidance, etc.).	$(X,Y,Z) \in \{A\}$
(p,q,r)	Angular rates in the body frame $\{\mathbf{B}\}$	$(p,q,r) \in \{B\}$
(u,v,w)	Velocity components in the body frame $\{\mathbf{B}\}$	$(u,v,w) \in \{B\}$
(x,y,z)	Positions in the body frame $\{\mathbf{B}\}$ Used to express the location of motors, sensors, etc, with respect to the C.G.	$(x,y,z) \in \{B\}$
$\vec{\mathbf{H}}$	Angular momentum, $\vec{\mathbf{H}} = \mathbf{I} \cdot \vec{\omega}$	$\vec{\mathbf{H}} \in \mathbb{R}^3$
$\mathbf{I}$	Inertia matrix expressed in the body fixed frame $\{B\}$	$\mathbf{I} \in \mathbb{R}^{3\times3}$
$\vec{\mathbf{M}}$	Total moments acting on the body fixed frame $\{B\}$	$\vec{\mathbf{M}} \in \mathbb{R}^3$
M_i	Moment acting along $\vec{\mathbf{b}}_i$ axis	$M_i \in \mathbb{R}$
Q_i	Torque from i^{th} rotor	$Q_i \in \mathbb{R}$
$\mathbf{R}$	Rotation matrix	In this document, $\mathbf{R} = {}^A\mathbf{R}_B$ $= \{\overrightarrow{\mathbf{b_1}}, \overrightarrow{\mathbf{b_2}}, \overrightarrow{\mathbf{b_3}}\} \in SO(3)$ mostly.
${}^A\mathbf{R}_B$	Rotation matrix from body fixed frame $\{B\}$ to the inertial frame $\{A\}$.	${}^A\mathbf{R}_B = ({}^B\mathbf{R}_A)^T$
${}^B\mathbf{R}_A$	Rotation matrix from the inertial frame $\{A\}$ to body fixed frame $\{B\}$.	3-2-1 (Z-Y-X, Yaw-Pitch-Roll) Order (→APPENDIX A.)
$SO(3)$	Three dimensional special orthogonal group [2]	
T_i	Thrust from i^{th} rotor along $-\vec{\mathbf{b}}_3$ direction	$F_i \in \mathbb{R}$
T_Σ	Total sum fo thrusts along $-\vec{\mathbf{b}}_3$ direction, $T_\Sigma = \Sigma_{i=1}^N T_i$	for quadcopter, $N = 4$
V_{tip}	Rotor blade tip velocity	$V_{tip} = \Omega R$
$\vec{\mathbf{r}}$	$\vec{\mathbf{r}} = (X,Y,Z)^T \in \{A\}$, Position of an airframe.	
$\vec{\mathbf{u}}$	Control variables	
$\vec{\mathbf{v}}$	Linear velocity of $\{B\}$ with respect to $\{A\}$, expressed in $\{A\}$	
v_h	Hover induced velocity, $\sqrt{T/2\rho A}$	
v_i	Induced velocity	
$\vec{\mathbf{x}}$	State variables	
$\vec{\mathbf{y}}$	Measurements	
Ω (Omega)	Rotational frequency of rotor	$[rad/s]$
θ (theta)	Pitch (Euler) angle	
ϕ (phi)	Roll (Euler) angle	
ψ (psi)	Yaw (Euler) angle	
$\vec{\omega}$ (omega)	$\vec{\omega} = (p,q,r) \in \{\mathbf{B}\}$, Angular velocity of $\{B\}$ with respect to $\{A\}$, expressed in $\{B\}$	

- Parameters

Expressions	Descriptions	Notes
A	Rotor disk area	$[m^2]$
A_b	Rotor blade area	$[m^2]$
$\mathbf{I}$	Inertia matrix expressed in the body fixed frame $\{B\}$	$\mathbf{I} \in \mathbb{R}^{3\times3}$
I_{xx}	x moment of inertia	
I_{xy}	xy product of inertia	≈ 0
I_{yy}	y moment of inertia	
I_{yz}	yz product of inertia	≈ 0
I_{zz}	z moment of inertia	
I_{zx}	zx product of inertia	≈ 0
N_b	Number of propeller blades	$= 2$
O_i	Rotating direction indicator for the i^{th} rotor	($+1$:CCW, -1:CW)
R	Rotor radius	$[m]$
c_b	Rotor blade chord length	$[m]$
l	Arm length	$[m]$
m	Body mass	$[kg]$
θ_{b0} (theta)	Blade collective pitch	$[rad]$
ξ_i (xi)	i^{th} rotor attachment angle	
σ (sigma)	Rotor solidity	$\frac{\text{Blade area}}{\text{Disk area}} = \frac{A_b}{A} = \frac{N_b cR}{\pi R^2} = \frac{N_b c}{\pi R}$

- Constants and Coefficients

Expressions	Descriptions	Notes
C_P	Rotor power coefficient	$C_P = \frac{P}{\rho A \Omega^3 R^3} \equiv C_Q \; (\because P = \Omega Q)$
C_Q	Rotor shaft torque coefficient	$C_Q = \frac{Q}{\rho A \Omega^2 R^3}$, 0.00224
C_T	Rotor thrust coefficient	$C_T = \frac{T}{\rho A \Omega^2 R^2}$, 0.0181
C_d	Section drag coefficient	
C_{d_0}	Section zero-lift drag coefficient	
C_l	Section lift coefficient	$C_l = L / \frac{1}{2} \rho U^2 c$
$C_{l_{max}}$	Maximum lift coefficient	
C_{l_α}	Section lift-curve slope	$= 2\pi$ (theoretical value)
g	Gravitational acceleration	$9.807 \, [m/s^2]$
k_M	Motor gain	$20[s^{-1}]$
k_Q	Lumped rotor torque coefficient	$k_Q = C_Q \rho A R^3$, 2.74×10^{-7} [$kg \cdot m^2$]
k_T	Lumped rotor thrust coefficient	$k_T = C_T \rho A R^2$, 1.75×10^{-5} [$kg \cdot m$]
ρ (rho)	Air density	$1.18 \, [kg/m^3]$, @ $25^\circ C, \; 1atm$
τ (tau)	Time constant	$1/k_M$

- Control Gains

Expressions	Descriptions	Notes
$K_{D,\dot{Z}}$	Z-velocity derivative gain	
$K_{D,\theta}$	Pitch angle derivative gain	
$K_{D,\phi}$	Roll angle derivative gain	
$K_{D,\psi}$	Yaw angle derivative gain	
$K_{I,\dot{Z}}$	Z-velocity integral gain	
$K_{I,\theta}$	Pitch angle integral gain	
$K_{I,\phi}$	Roll angle integral gain	
$K_{I,\psi}$	Yaw angle integral gain	
$K_{P,\dot{Z}}$	Z-velocity proportional gain	
$K_{P,\theta}$	Pitch angle proportional gain	
$K_{P,\phi}$	Roll angle proportional gain	
$K_{P,\psi}$	Yaw angle proportional gain	

- Abbreviations

Expressions	Descriptions	Notes
AOA	Angel of Attack	
$BEMT$	Blade Element Momentum Theory	
BET	Blade Element Theory	
FM	Figure of Merit	
$NACA$	National Advisory Committee for Aeronautics	now NASA
$NASA$	National Aeronautics and Space Administration	formally NACA
$VTOL$	Vertical Take-Off and Landing	

1. Quad(+) Airframe

1.1. Configuration

Distribute four motors along clockwise direction starting from the front arm. Must use the suitable propeller correspond to the rotating direction of each motor. To get the highest efficiency, you should also investigate the specification of motors to best select a propeller with a proper diameter and pitch.

Motor #	Location	Rotating direction	Notes
1	Front	Counter-clockwise	$O_1 = +1, \xi_1 = 0^\circ$
2	Left	Clockwise	$O_2 = -1, \xi_2 = 270^\circ$
3	Back	Counter-clockwise	$O_3 = +1, \xi_3 = 180^\circ$
4	Right	Clockwise	$O_4 = -1, \xi_4 = 90^\circ$

1.2. Specifications (nominal plant)

Enter the nominal specification to be used in the simulation environments. (optional) You can also set some uncertainties to each parameter to check the robustness of your controller.

Specifications	Value	Uncertainties (Units are same w/ the left col.)
Mass, m	1.25 $[kg]$	$\pm 10\%$
Arm length, l	0.265 $[m]$	± 0.01
x moment of inertia, I_{xx}	0.0232 $[kg \cdot m^2]$	± 0.01
y moment of inertia, I_{yy}	0.0232 $[kg \cdot m^2]$	± 0.01
z moment of inertia, I_{zz}	0.0468 $[kg \cdot m^2]$	± 0.01
-	-	-

2. Dynamic model of a quad(+)copter

2.1. State Variables

In this simulation, we take 12 state variables, i.e., position, velocity, and attitude of an airframe in the inertial frame; and angular rates.

$$\vec{\mathbf{x}} = [X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, \phi, \theta, \psi, p, q, r]^T$$

2.2. Measurements

In this simulation, we consider all state variables are measurable (observable) since most of the recent GPS/INS sensors provide above information. If you want to build your own system starting from raw sensor measurements, such as magnetometers, gyros, GPS antennas, etc, one can search for the Kalman Filter or Linear Quadratic Gaussian (LQG) controller for reference.

2.3. Sensor (noise) Model

In this simulation, we do not take into account the effects associated with signal processing, such as filtering to handle noises, which may be included in the sensor measurements.

2.4. Control Variables

We take four control inputs, which are total thrust, and moments on each control axis.

$$\vec{\mathbf{u}} = [u_1, u_2, u_3, u_4]^T = [T_\Sigma, M_1, M_2, M_3]^T,$$

2.5. Aerodynamic Considerations

- **Steady-state Thrust Estimation Model** When the rotor is on a steady state (near hovering position), we can express the thrust from a i^{th}rotor as follows[3]:

  $$T_i = C_T \rho A V_{tip}^2 = C_T \rho A R^2 \Omega_i^2 (> 0).$$

  Here, $k_T := C_T \rho A R^2$ can be considered as a lumped thrust coefficient at a given flight condition.

  When we are not able to get the thrust coefficient experimentally, one way to estimate the coefficient is applying Blade Element Momentum Theory (BEMT) for a given rotor design.

$$C_T = \frac{1}{2}\sigma C_{l\alpha}\left[\frac{\theta_{b0}}{3} - \frac{1}{2}\sqrt{\frac{C_T}{2}}\right]$$

$$\text{(by guessing iteratively with:) } \theta_{b0} = \frac{6C_T}{\sigma C_{l\alpha}} + \frac{3}{2}\sqrt{\frac{C_T}{2}}$$

$$\text{(by using analytic solution:) } C_T = \frac{1}{192}\left[\left[3 \pm \sqrt{9 + 192\frac{\theta_{b0}}{\sigma C_{l\alpha}}}\right] \cdot (\sigma C_{l\alpha})^2 + 32\sigma C_{l\alpha}\theta_{b0}\right]$$

One can use a EPP1045 propeller, which has a NACA5306 airfoil with the rotor solidity, $\sigma = \frac{2 \cdot 0.020[m]}{\pi \cdot 0.127[m]} = 0.1003$; section lift-curve slope, $C_{l\alpha} = 6.045$; and blade collective pitch, $\theta_{b0} = 18.43^\circ = 0.3217[rad]$ approximately. Putting these values yields theoritical thrust coefficient, $C_T = 0.0181$.

> cf) One can estimate the lift curve from XFLR5, which is an analysis tool for airfoils, wings and planes operating at low Reynolds Numbers based on XFoil, Lifting Line Theory (LLT), Vortex Lattice Method (VLM), and 3D Panel Method.

- **Steady-state Reaction Torque Estimation Model** When the rotor is on a steady state (near hovering position), we can express the torque from a i^{th} rotor as follows[3]:

$$Q_i = C_Q \rho A V_{tip}^2 R = C_Q \rho A R^3 \Omega_i^2 (> 0).$$

Here, $k_Q := C_Q \rho A R^3$ also can be considered as a lumped torque coefficient for a given flight condition.

From the BEMT, power coeffieicnt (numerically identical with torque coefficient) on hovering condition can be estimated as follows:

$$C_P = \frac{C_T^{3/2}}{\sqrt{2}} + \frac{1}{8}\sigma C_{d0}$$

Putting rotor thrust coefficient, $C_T = 0.0181$; rotor solidity, $\sigma = \frac{2 \cdot 0.020[m]}{\pi \cdot 0.127[m]} = 0.1003$; section zero-lift drag coefficient, $C_{d0} = 0.041$ yields $C_P = C_Q = 0.00224$.

2.6. Thrust and Moment Distributions

Total thrust and moments generated by N number of rotors can be expressed as follows:

$$
\begin{aligned}
T_\Sigma &= \sum_{i=1}^{N} T_i = k_T \sum_{i=1}^{N} \Omega_i^2 \\
M_1 &= -\sum_{i=1}^{N} l_i \sin(\xi_i) T_i = -k_T \sum_{i=1}^{N} l_i \sin(\xi_i) \Omega_i^2 \\
M_2 &= \sum_{i=1}^{N} l_i \cos(\xi_i) T_i = k_T \sum_{i=1}^{N} l_i \cos(\xi_i) \Omega_i^2 \\
M_3 &= \sum_{i=1}^{N} O_i Q_i = k_Q \sum_{i=1}^{N} O_i \Omega_i^2
\end{aligned}
$$

Therefore, symmetric multi-copter with $N = 4$ (quadcopter) and $l_i = l$ (same arm length), one can get the desired angluar speed of each rotors by,

$$
\begin{bmatrix} T_\Sigma \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ 0 & k_T l & 0 & -k_T l \\ k_T l & 0 & -k_T l & 0 \\ k_Q & -k_Q & k_Q & -k_Q \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}
$$

$$
\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} k_T & k_T & k_T & k_T \\ 0 & k_T l & 0 & -k_T l \\ k_T l & 0 & -k_T l & 0 \\ k_Q & -k_Q & k_Q & -k_Q \end{bmatrix}^{-1} \begin{bmatrix} T_\Sigma \\ M_1 \\ M_2 \\ M_3 \end{bmatrix}
$$

or thrust required as follows:

$$
\begin{bmatrix} T_\Sigma \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ l & 0 & -l & 0 \\ k_Q/k_T & -k_Q/k_T & k_Q/k_T & -k_Q/k_T \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}
$$

$$
\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & l & 0 & -l \\ l & 0 & -l & 0 \\ k_Q/k_T & -k_Q/k_T & k_Q/k_T & -k_Q/k_T \end{bmatrix}^{-1} \begin{bmatrix} T_\Sigma \\ M_1 \\ M_2 \\ M_3 \end{bmatrix}
$$

2.7. Equations of Motion

- **Translational motion** The rate of change for the 6 state variables from the beginning ($\dot{X}, \dot{Y}, \dot{Z}, \ddot{X}, \ddot{Y}, \ddot{Z}$) can be obtained by the following translational motion equations:

$$\dot{\vec{\mathbf{r}}} = \vec{\mathbf{v}}$$
$$m\dot{\vec{\mathbf{v}}} = mg\overrightarrow{\mathbf{a_3}} + \mathbf{R} \cdot T_\Sigma \cdot (-\vec{\mathbf{z}}) \qquad \text{(Newton's 2nd law: } \frac{d}{dt}(m\vec{\mathbf{v}}) = \sum \overrightarrow{\mathbf{F_i}} \text{ , with APPENDIX C.1.)}$$

In detail ($\dot{\vec{\mathbf{r}}}$ terms are trivial);

$$\ddot{X} = -\frac{1}{m}(sin\psi \cdot sin\phi + cos\psi \cdot sin\theta \cdot cos\phi) \cdot T_\Sigma$$
$$\ddot{Y} = -\frac{1}{m}(-cos\psi \cdot sin\phi + sin\psi \cdot sin\theta \cdot cos\phi) \cdot T_\Sigma$$
$$\ddot{Z} = -\frac{1}{m}cos\theta \cdot cos\phi \cdot T_\Sigma + g$$

- **Rotational motion** The rate of change for the other 6 state variables behind ($\dot{\phi}, \dot{\psi}, \dot{\theta}, \dot{p}, \dot{q}, \dot{r}$) can be obtained from the following rotational motion equations:

$$\dot{\mathbf{R}} = \mathbf{R} \cdot sk(\vec{\omega})$$
$$\mathbf{I}\dot{\vec{\omega}} = -\vec{\omega} \times \mathbf{I}\vec{\omega} + \mathbf{M} \qquad \text{(Newton's 2nd law: } \frac{d}{dt}(\vec{\mathbf{H}}) = \sum \overrightarrow{\mathbf{M}}_i \text{ , with APPENDIX C.1.)}$$

Using Euler angle parameterization;

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)sec(\theta) & c(\phi)sec(\theta) \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

$$\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} \cdot qr + \frac{1}{I_{xx}} \cdot M_1$$
$$\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} \cdot rp + \frac{1}{I_{yy}} \cdot M_2$$
$$\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} \cdot pq + \frac{1}{I_{zz}} \cdot M_3$$

2.8. Subsystems

- **Motor Dynamics, Battery, ESC, etc.**

  > To take more subsystems and physics regarding the quadcopter; such as batteries, electornic speed controllers, motors, gyroscopic effects, wind model, etc.; into account, one can apply the dynamics of each system to the simulator.

  Here we apply motor dynamics(1st order dynamic system with a time delay) as an example:

$$\dot{\Omega}_i = k_m(\Omega_{i,des} - \Omega_i)$$

3. Attitude control of a quad(+)copter

3.1. Overview

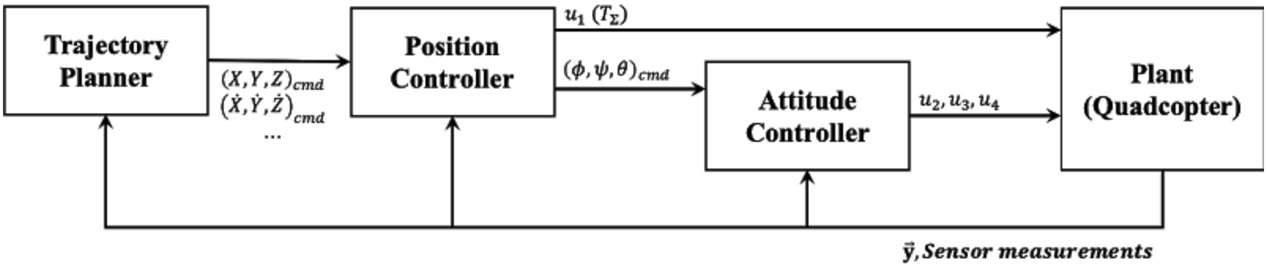

Figure 2. Quadcopter control loop design

3.2. Linearization

> You can skip this part if you are not about to design a linear controller, such as LQR, LQG, H_∞, etc. In this document, we will mainly focus on the PID attitude controller.

Nonlinear quadcopter state-space model can be summarized as follows:

$$\dot{\vec{\mathbf{x}}} = \mathbf{f}(\vec{\mathbf{x}}, \vec{\mathbf{u}})$$

$$\begin{aligned} \text{where, } \vec{\mathbf{x}} &= [X, Y, Z, \dot{X}, \dot{Y}, \dot{Z}, \phi, \theta, \psi, p, q, r]^T, \\ \vec{\mathbf{u}} &= [u_1, u_2, u_3, u_4]^T = [T_\Sigma, M_1, M_2, M_3]^T \end{aligned}$$

Therefore, expressing $\dot{\vec{\mathbf{x}}}$ in terms of $\vec{\mathbf{x}}$, and $\vec{\mathbf{u}}$ can be achieved from the **Section 2.7**:

$$\begin{aligned}
\dot{X} &= \dot{X} \\
\dot{Y} &= \dot{Y} \\
\dot{Z} &= \dot{Z} \\
\ddot{X} &= -\frac{1}{m}(sin\psi \cdot sin\phi + cos\psi \cdot sin\theta \cdot cos\phi) \cdot T_\Sigma \\
\ddot{Y} &= -\frac{1}{m}(-cos\psi \cdot sin\phi + sin\psi \cdot sin\theta \cdot cos\phi) \cdot T_\Sigma \\
\ddot{Z} &= -\frac{1}{m}cos\theta \cdot cos\phi \cdot T_\Sigma + g \\
\dot{\phi} &= p + s(\phi)t(\theta) \cdot q + c(\phi)t(\theta) \cdot r \\
\dot{\theta} &= c(\phi) \cdot q - s(\phi) \cdot r \\
\dot{\psi} &= s(\phi)sec(\theta) \cdot q + c(\phi)sec(\theta) \cdot r \\
\dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}} \cdot qr + \frac{1}{I_{xx}} \cdot M_1 \\
\dot{q} &= \frac{I_{zz} - I_{xx}}{I_{yy}} \cdot rp + \frac{1}{I_{yy}} \cdot M_2 \\
\dot{r} &= \frac{I_{xx} - I_{yy}}{I_{zz}} \cdot pq + \frac{1}{I_{zz}} \cdot M_3
\end{aligned}$$

Linearization around the (hovering) equilibrium point, $\vec{\mathbf{x}}_e = [0,0,0,0,0,0,0,0,0,\psi_e,0,0,0]^T$, and $\vec{\mathbf{u}}_e = [mg,0,0,0]^T$ yields linear quadcopter state-space equation as follows:

$$\dot{\vec{\mathbf{x}}}' = A\vec{\mathbf{x}}' + B\vec{\mathbf{u}}'$$

where,

$$\mathbf{A} = \frac{\partial \mathbf{f}(\vec{\mathbf{x}},\vec{\mathbf{u}})}{\partial \vec{\mathbf{x}}}\bigg|_{\substack{\vec{\mathbf{x}}=\vec{\mathbf{x}}_e \\ \vec{\mathbf{u}}=\vec{\mathbf{u}}_e}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -g\sin(\psi_e) & -g\cos(\psi_e) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g\cos(\psi_e) & -g\sin(\psi)_e & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{B} = \frac{\partial \mathbf{f}(\vec{\mathbf{x}},\vec{\mathbf{u}})}{\partial \vec{\mathbf{u}}}\bigg|_{\substack{\vec{\mathbf{x}}=\vec{\mathbf{x}}_e \\ \vec{\mathbf{u}}=\vec{\mathbf{u}}_e}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -1/m & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1/I_{xx} & 0 & 0 \\ 0 & 0 & 1/I_{yy} & 0 \\ 0 & 0 & 0 & 1/I_{zz} \end{bmatrix}$$

and $\vec{\mathbf{x}}' = \vec{\mathbf{x}} - \vec{\mathbf{x}}_e$, $\vec{\mathbf{u}}' = \vec{\mathbf{u}} - \vec{\mathbf{u}}_e$ stand for deviation of states and control inptus from steady state.

> **cf) Linearlization using Tayler expansion**
>
> $$\dot{\vec{\mathbf{x}}} = \mathbf{f}(\vec{\mathbf{x}}_e,\vec{\mathbf{u}}_e) + \frac{\partial \mathbf{f}(\vec{\mathbf{x}},\vec{\mathbf{u}})}{\partial \vec{\mathbf{x}}}\bigg|_{\substack{\vec{\mathbf{x}}=\vec{\mathbf{x}}_e \\ \vec{\mathbf{u}}=\vec{\mathbf{u}}_e}} (\vec{\mathbf{x}} - \vec{\mathbf{x}}_e) + \frac{\partial \mathbf{f}(\vec{\mathbf{x}},\vec{\mathbf{u}})}{\partial \vec{\mathbf{u}}}\bigg|_{\substack{\vec{\mathbf{x}}=\vec{\mathbf{x}}_e \\ \vec{\mathbf{u}}=\vec{\mathbf{u}}_e}} (\vec{\mathbf{u}} - \vec{\mathbf{u}}_e) + \text{H.O.T.}$$
> $$\dot{\vec{\mathbf{x}}} = \dot{\vec{\mathbf{x}}}_e + \frac{\partial \mathbf{f}(\vec{\mathbf{x}},\vec{\mathbf{u}})}{\partial \vec{\mathbf{x}}}\bigg|_{\substack{\vec{\mathbf{x}}=\vec{\mathbf{x}}_e \\ \vec{\mathbf{u}}=\vec{\mathbf{u}}_e}} (\vec{\mathbf{x}} - \vec{\mathbf{x}}_e) + \frac{\partial \mathbf{f}(\vec{\mathbf{x}},\vec{\mathbf{u}})}{\partial \vec{\mathbf{u}}}\bigg|_{\substack{\vec{\mathbf{x}}=\vec{\mathbf{x}}_e \\ \vec{\mathbf{u}}=\vec{\mathbf{u}}_e}} (\vec{\mathbf{u}} - \vec{\mathbf{u}}_e) + \text{H.O.T.}$$
>
> neglecting H.O.T yields;
>
> $$\dot{\vec{\mathbf{x}}}' = A\vec{\mathbf{x}}' + B\vec{\mathbf{u}}'$$

3.3. Controller Design

- **Attitude control**

> This document only presents attitude and Z-velocity PID controller only to verify the correct dynamic simulation.

By assuming $\dot{\phi} \approx p, \dot{\theta} \approx q$, and $\dot{\psi} \approx r$ near the hovering position, we can design a PID controller working in a moderate range of attitude commands. (Please beaware that the positive-Z means downwards in the local NED frame. This explains the negative sign after the "mg" in the T_Σ for the Z-velocity PID control.)

$$T_\Sigma = mg - \left[K_{P,\dot{Z}}(\dot{Z}^{cmd} - \dot{Z}) + K_{I,\dot{Z}} \int (\dot{Z}^{cmd} - \dot{Z}) + K_{D,\dot{Z}}(\ddot{Z}^{cmd} - \ddot{Z})\right]$$

$$M_1 = K_{P,\phi}(\phi^{cmd} - \phi) + K_{I,\phi} \int (\phi^{cmd} - \phi) + K_{D,\phi}(p^{cmd} - p)$$

$$M_2 = K_{P,\theta}(\theta^{cmd} - \theta) + K_{I,\theta} \int (\theta^{cmd} - \theta) + K_{D,\theta}(q^{cmd} - q)$$

$$M_3 = K_{P,\psi}(\psi^{cmd} - \psi) + K_{I,\psi} \int (\psi^{cmd} - \psi) + K_{D,\psi}(r^{cmd} - r)$$

- **Position(Trajectory) control**

> Leaved for your exercise. Here you should design a control stratagy to generate proper attitude commands at a given desired trajectory and/or plant measurements. By adding this loop, one can put a quadcopter at a desired location anywhere in the local navigation (inertial) frame.

- **Trajectory planning**

> Leaved for your exercise. Here you should design a trajectory planner that can optimize the position of a quadcopter with respect to the time. By adding this loop, one can apply the quadcopter to a lot of practical works, such as monitoring, mapping, autonomous landing, obstacle avoidance, etc.

APPENDIX A. LINEAR ALGEBRA

A.1. Skew-symmetric Matrix (also called anti-symmetric)

- $A^T = -A$
- $A + A^T = 0$
- Always singluar: $det(A) = 0$
- Any matrix is the sum of a symmetric and skew symmetric matrix.
- In three-dimensions:

$$A(v) = \begin{pmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{pmatrix} \text{,which can be summarized as, } A(v = [x, y, z]).$$

- Gives alternative way to express the vector cross product: $\mathbf{a} \times \mathbf{b} = [\mathbf{a}_\times]\mathbf{b}$

> Proof) Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$. The vector product $\mathbf{a} \times \mathbf{b}$, in components, reads:

$$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

From the equality above one can see that the following skew-symmetric matrix:

$$[\mathbf{a}_\times] = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$$

APPENDIX B. COORDINATE TRANSFORMATION MATRIX

B.1. Rotation Matrix at a given Attitude

Rotation matrix taking Yaw, Pitch, and Roll sequence can be expressed as follows[2]:

$$
\begin{aligned}
{}^{3}\mathbf{R}_A(\psi) &= \begin{bmatrix} c(\psi) & s(\psi) & 0 \\ -s(\psi) & c(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
{}^{2}\mathbf{R}_3(\theta) &= \begin{bmatrix} c(\theta) & 0 & -s(\theta) \\ 0 & 1 & 0 \\ s(\theta) & 0 & c(\theta) \end{bmatrix} \\
{}^{(B=)1}\mathbf{R}_2(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & c(\phi) & s(\phi) \\ 0 & -s(\phi) & c(\phi) \end{bmatrix}
\end{aligned}
$$

where, $c(\cdot) = \cos(\cdot)$, $s(\cdot) = \sin(\cdot)$.

$$
{}^{B}\mathbf{R}_A = {}^{B}\mathbf{R}_2{}^{2}\mathbf{R}_3{}^{3}\mathbf{R}_A = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ -s(\psi)c(\phi)+c(\psi)s(\theta)s(\phi) & c(\psi)c(\phi)+s(\psi)s(\theta)s(\phi) & c(\theta)s(\phi) \\ s(\psi)s(\phi)+c(\psi)s(\theta)c(\phi) & -c(\psi)s(\phi)+s(\psi)s(\theta)c(\phi) & c(\theta)c(\phi) \end{bmatrix}. \qquad (B.1a)
$$

$$
\mathbf{R} = {}^{A}\mathbf{R}_B = ({}^{B}\mathbf{R}_A)^T = \begin{bmatrix} c(\psi)c(\theta) & -s(\psi)c(\phi)+c(\psi)s(\theta)s(\phi) & s(\psi)s(\phi)+c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi)+s(\psi)s(\theta)s(\phi) & -c(\psi)s(\phi)+s(\psi)s(\theta)c(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}. \qquad (B.1b)
$$

B.2. Derivative of a Rotation Matrix

- Derivative of a rotation matrix can be expressed with a skew-symmetrix matrix:

> Proof)
> $\mathbf{R}\mathbf{R}^{\mathbf{T}} = \mathbf{I}$
> $\mathbf{R_x}(\phi)\mathbf{R_x^T}(\phi) = \mathbf{I}$
> $\frac{d}{d\phi}\mathbf{R_x}(\phi)\cdot\mathbf{R_x^T}(\phi) + \mathbf{R_x}(\phi)\cdot\frac{d}{d\phi}\mathbf{R_x^T}(\phi) = \mathbf{0}$
> $\frac{d}{d\phi}\mathbf{R_x}(\phi)\cdot\mathbf{R_x^T}(\phi) + \left(\frac{d}{d\phi}\mathbf{R_x}(\phi)\cdot\mathbf{R_x^T}(\phi)\right)^T = \mathbf{0} \leftarrow \text{(property of a skew-symmetric matrix!)}$
> Now, let $\mathbf{S} = \frac{d}{d\phi}\mathbf{R_x}(\phi)\cdot\mathbf{R_x^T}(\phi)$, where $\mathbf{R_x}(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$. Then,
> $\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\sin\phi & -\cos\phi \\ 0 & \cos\phi & -\sin\phi \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} = \mathbf{S}([1,0,0])$
> Therefore, $\frac{d}{d\phi}\mathbf{R_x}(\phi) = \mathbf{S}([1,0,0])\cdot\mathbf{R_x}(\phi)$
> $(\because \mathbf{R_x^T}(\phi) = \mathbf{R_x^{-1}}(\phi))$

Similarly for other axes,

$$\frac{d}{d\theta}\mathbf{R_y}(\theta) = \mathbf{S}([0,1,0]) \cdot \mathbf{R_y}(\theta)$$
$$\frac{d}{d\psi}\mathbf{R_z}(\psi) = \mathbf{S}([0,0,1]) \cdot \mathbf{R_z}(\psi)$$
$$\frac{d}{d\alpha}\mathbf{R}_l(\alpha) = \mathbf{S}(l) \cdot \mathbf{R}_l(\alpha)$$
$$(l \text{ for arbitrary axis of rotation})$$

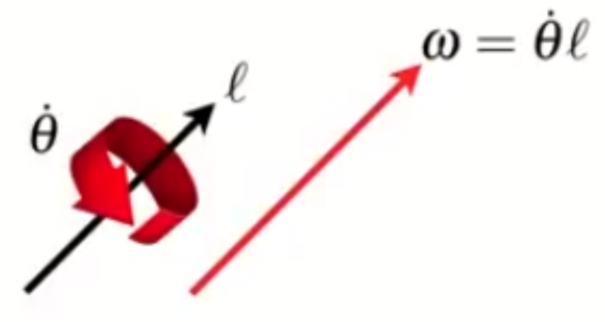

Figure B1. Rotation with an arbitrary axis

Now, putting $d\alpha/dt$ on both sides yields,

$$\frac{d\alpha}{dt}\frac{d}{d\alpha}\mathbf{R}_l(\alpha) = \frac{d\alpha}{dt}\mathbf{S}(l) \cdot \mathbf{R}_l(\alpha)$$
$$\dot{\mathbf{R}}_l(\alpha) = \mathbf{S}(\omega) \cdot \mathbf{R}_l(\alpha)$$
$$(\text{where } l \text{ is a axis of rotation, and } \omega \text{ is a angular velocity vector.})$$

B.3. Euler Rates and Body Angular Rates

Since $\dot{\phi}, \dot{\theta}$,and $\dot{\psi}$ are small and noting that ${}^3\mathbf{R}_A(\dot{\psi}) = {}^2\mathbf{R}_3(\dot{\theta}) = {}^B\mathbf{R}_2(\dot{\phi}) = \mathbf{I}$, we can relate body angular rates (p, q, r) to Euler rates $(\dot{\phi}, \dot{\theta}, \dot{\psi})$ as follows:

$$
\begin{aligned}
\begin{pmatrix} p \\ q \\ r \end{pmatrix} &= {}^B\mathbf{R}_2(\dot{\phi}) \begin{pmatrix} \dot{\phi} \\ 0 \\ 0 \end{pmatrix} + {}^B\mathbf{R}_2(\phi)^2\mathbf{R}_3(\dot{\theta}) \begin{pmatrix} 0 \\ \dot{\theta} \\ 0 \end{pmatrix} + {}^B\mathbf{R}_2(\phi)^2\mathbf{R}_3(\theta)^3\mathbf{R}_A(\dot{\psi}) \begin{pmatrix} 0 \\ 0 \\ \dot{\psi} \end{pmatrix} \\
&= \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\phi) & c(\theta)s(\phi) \\ 0 & -s(\phi) & c(\theta)c(\phi) \end{bmatrix} \begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} \\
\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} &= \begin{bmatrix} 1 & 0 & -s(\theta) \\ 0 & c(\phi) & c(\theta)s(\phi) \\ 0 & -s(\phi) & c(\theta)c(\phi) \end{bmatrix}^{-1} \begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{bmatrix} 1 & s(\phi)t(\theta) & c(\phi)t(\theta) \\ 0 & c(\phi) & -s(\phi) \\ 0 & s(\phi)sec(\theta) & c(\phi)sec(\theta) \end{bmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}
\end{aligned}
$$

> CAUTIONS!
>
> - NOT ORTHOGONAL.
> - SINGULARITY @ $\theta = \pi/2$.

APPENIDX C. DYNAMICS

C.1. Motion Relative to Rotating Axes (Equation of Coriolis)

The time derivative of any vector quentity $\vec{\mathbf{a}}$ with respect to the (fixed) XY system, can be expressed with the (rotating) xy reference frame. It is a sum of (a) total derivative of $\vec{\mathbf{a}}$ which is measured relative to the xy reference system; and (b) difference between the time derivative of the vector as measured in a fixed reference system and its time derivative as measured in the rotating reference system:

$$\left(\frac{d\vec{\mathbf{a}}}{dt}\right)_{XY} = \left(\frac{d\vec{\mathbf{a}}}{dt}\right)_{xy} + \omega_{xy/XY} \times \vec{\mathbf{a}}$$

(the proof can be found in the most dynamics textbooks)

References

[1] Groves, P. D. (2013). *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house.

[2] 김병수, 김유단, 방효충, 탁민제, 홍성경. (2004). *비행동역학 및 제어*. 경문사.

[3] Leishman, G. J. (2006). *Principles of helicopter aerodynamics*. Cambridge university press.

FOOTNOTES

1. **[Inertial frame: 관성좌표계]** Any coordinate frame that does not accelerate or rotate with respect to the rest of the Universe is an *inertial frame*. An *Earth-centered inertial frame* is norminally centered at the Earth's center of mass and oriented with respect to the Earth's spin axis and the stars. This is not strictly an inertial frame as the Earth eperiences acceleration in its orbit around the Sun, its spin axis slowly moves, and the galaxy rotates. However, these effects are smaller than the measurement noise exhibited by navigation sensors, so an ECI frame may be treated as a true inertial frame for all practical purposes[1]. Simplest expression: A frame of reference in which Newtons's laws apply.↩
2. **Special Orthogonal Group, SO(3): 3차원 특수직교군**↩