MP-282
Dynamic Modeling and Control of Multirotor Aerial Vehicles
Chapter 1: Introduction

Prof. Dr. Davi Antônio dos Santos
Instituto Tecnológico de Aeronáutica
www.professordavisantos.com

São José dos Campos - SP
2020

Contents

Dynamic modeling . . .

Notation

Geometric and Algebraic Vectors:

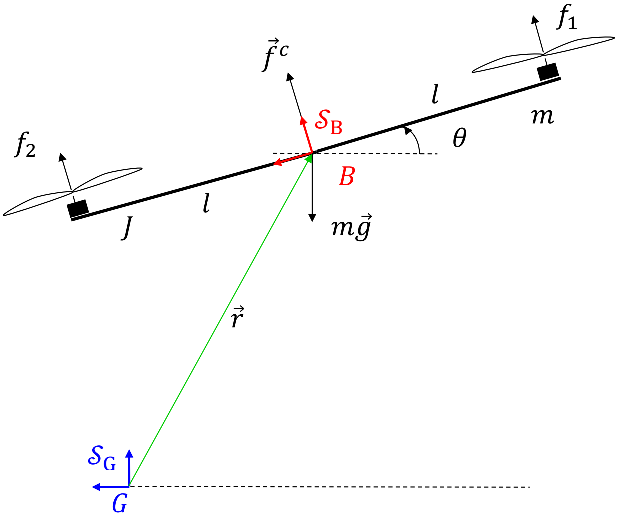
- $\overrightarrow{a}$: geometric vector
- $\hat{a}$: unit geometric vector
- $\mathcal{S}_{\mathrm{B}} \triangleq \{B; \hat{x}_{\mathrm{B}}, \hat{y}_{\mathrm{B}}, \hat{z}_{\mathrm{B}}\}$: body Cartesian coordinate system (CCS)
- $\mathcal{S}_{\mathrm{G}} \triangleq \{G; \hat{x}_{\mathrm{G}}, \hat{y}_{\mathrm{G}}, \hat{z}_{\mathrm{G}}\}$: ground CCS
- $\mathbf{a}_{\mathrm{B}}$: representation of $\overrightarrow{a}$ in $\mathcal{S}_{\mathrm{B}}$ (algebraic vector)
- $\bar{\mathbf{a}}_{\mathrm{B}}$: is a command for $\mathbf{a}_{\mathrm{B}}$
- $\mathbf{D}^{\mathrm{B/G}}$: attitude matrix of $\mathcal{S}_{\mathrm{B}}$ w.r.t. $\mathcal{S}_{\mathrm{G}}$
- $\mathbf{e}_1 \triangleq [1\ 0]^{\mathrm{T}}$ and $\mathbf{e}_2 \triangleq [0\ 1]^{\mathrm{T}}$

Some properties of $\mathbf{D}^{\mathrm{B/G}} \in \mathrm{SO}(3)$ (see more in Chapter 2):

$$\mathbf{a}_{\mathrm{B}} = \mathbf{D}^{\mathrm{B/G}} \mathbf{a}_{\mathrm{G}}$$

$$\left(\mathbf{D}^{\mathrm{B/G}}\right)^{\mathrm{T}} = \left(\mathbf{D}^{\mathrm{B/G}}\right)^{-1} = \mathbf{D}^{\mathrm{G/B}}$$

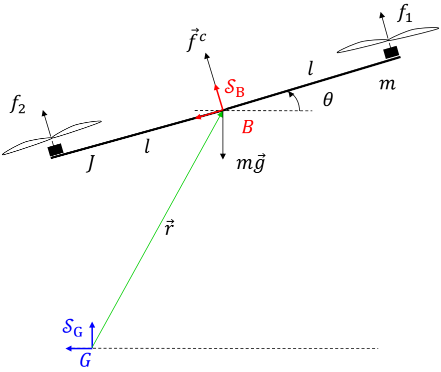
A Fictitious MAV in 2D



Translational Motion

Using the Newton's Second Law, the MAV's translational motion can be described in $\mathcal{S}_{\mathrm{G}}$ by

$$\ddot{\mathbf{r}}_{\mathrm{G}} = -g\mathbf{e}_2 + \frac{1}{m}\mathbf{D}^{\mathrm{G/B}}(f_1 + f_2)\mathbf{e}_2 \qquad (1)$$



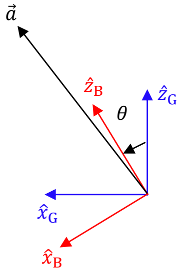
Rotational (Attitude) Motion

Using the Euler's Second Law, the MAV's attitude motion is described in $\mathcal{S}_{\mathrm{B}}$ (or wherever, since there is just one rotational DOF) by

$$\ddot{\theta} = \frac{1}{J}\tau^{c} \quad (2)$$

The relation between θ and the attitude matrix $\mathbf{D}^{\mathrm{B/G}}$ (whose transpose appears in equation (1)) is

$$\mathbf{D}^{\mathrm{B/G}} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (3)$$

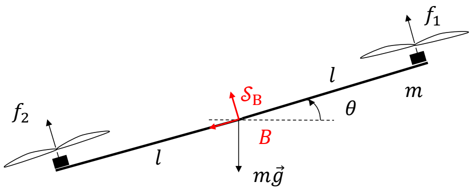


Rotor Configuration

By the geometry in the illustration, we see that the relationship between the individual thrust force magnitudes f_1 and f_2 and the resulting efforts (force and torque) is:

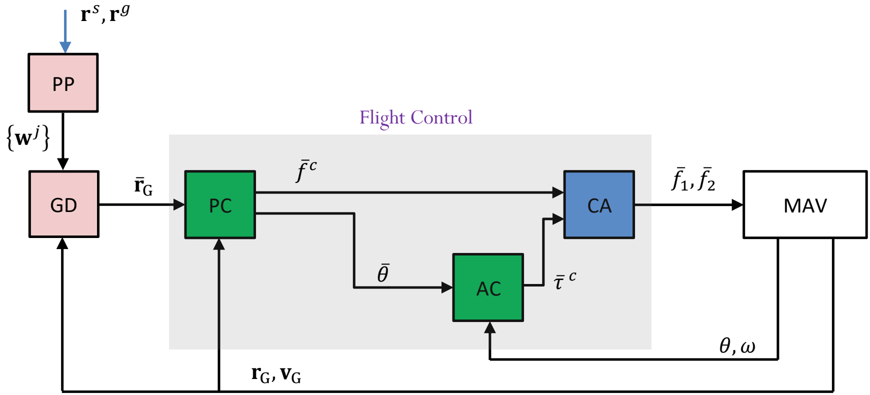
$$\begin{bmatrix} f^c \\ \tau^c \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ l & -l \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \quad (4)$$

where $f^c \triangleq f_1 + f_2$.



Flight Control . . .

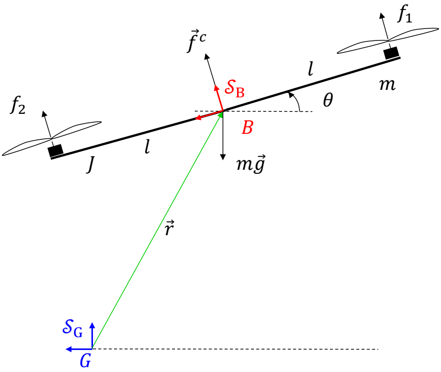
A Hierarchical Control Scheme



Abbreviations: PP (Path Planning)
GD (Guidance)
PC (Position Controller)
AC (Attitude Controller)
CA (Control Allocation)

Time-Scale Separation

- Assume that AC is much faster than PC
- In this case, AC and PC can be designed separately



Attitude Controller

Consider the design model

$$\ddot{\theta} = \frac{1}{J}\bar{\tau}^c \tag{5}$$

Let us adopt a simple saturated proportional-derivative attitude control law:

$$\bar{\tau}^c = \mathrm{sat}_{[-\tau^{\max},\tau^{\max}]}\left(JK_1\left(\bar{\theta} - \theta\right) - JK_2\dot{\theta}\right) \tag{6}$$

From (5)–(6), if the saturation is not active, one can approximately describe the closed-loop attitude dynamics by:

$$\ddot{\theta} + K_2\dot{\theta} + K_1\theta = K_1\bar{\theta} \tag{7}$$

Remark: From equation (7), one can easily design K_1 and K_2, *e.g.*, for given specifications of t_p (peak instant) and M_P (overshoot).

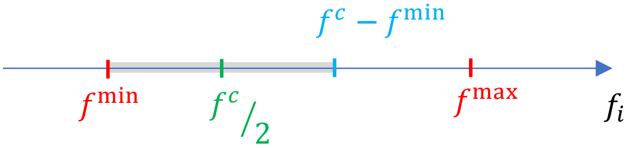
Attitude Controller

Assume that $f^{\min} \leq f_1, f_2 \leq f^{\max}$, $f^{\max} > f^c$, and $f^{\min} > 0$. Therefore, in order to respect the actuation symmetry of f_1 and f_2 with respect to $f^c/2$ (why?), one must assure that

$$f_1, f_2 \in \left[f^{\min}, f^c - f^{\min}\right] \tag{8}$$

Consequently, the maximal torque bound is

$$\tau^{\max} = \left(f^c - 2f^{\min}\right) l \tag{9}$$



Position Controller

Consider the design model

$$\ddot{\mathbf{r}}_{\mathrm{G}} = -g\mathbf{e}_2 + \frac{1}{m}\bar{\mathbf{f}}_{\mathrm{G}}^{c} \tag{10}$$

Let us adopt a simple saturated proportional-derivative position control law:

$$\bar{\mathbf{f}}_{\mathrm{G}}^{c} = \mathrm{sat}_{[\mathbf{f}^{\min},\mathbf{f}^{\max}]}\left(m\mathbf{K}_3\left(\bar{\mathbf{r}}_{\mathrm{G}} - \mathbf{r}_{\mathrm{G}}\right) - m\mathbf{K}_4\dot{\mathbf{r}}_{\mathrm{G}} + mg\mathbf{e}_2\right) \tag{11}$$

From (10)–(11), if the saturation is not active, one can approximately describe the closed-loop position dynamics by:

$$\ddot{\mathbf{r}}_{\mathrm{G}} + \mathbf{K}_4\dot{\mathbf{r}}_{\mathrm{G}} + \mathbf{K}_3\mathbf{r}_{\mathrm{G}} = \mathbf{K}_3\bar{\mathbf{r}}_{\mathrm{G}} \tag{12}$$

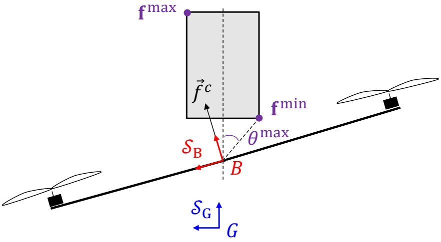
Remark: Just like the attitude controller, one can easily design $\mathbf{K}_3$ and $\mathbf{K}_4$ for given t_p and M_P. Compared to the attitude control design, here t_p must be much larger to respect the time-scale separation assumption!

Position Controller

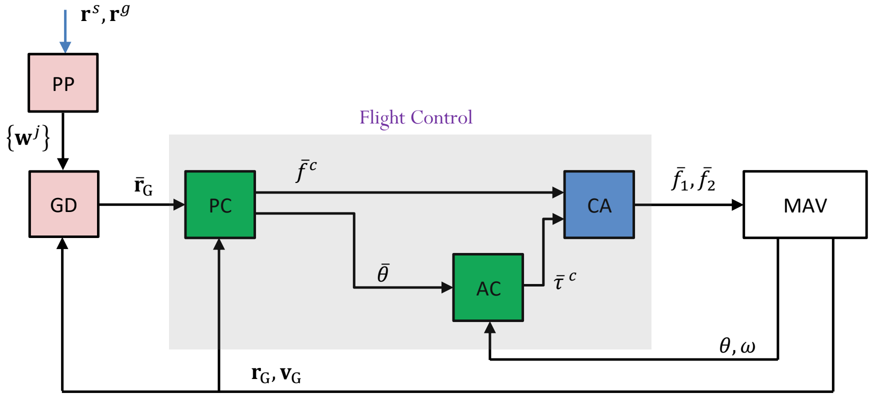
In order to prevent large inclinations, rotor turn-off, and actuator saturation, one can choose the force bounds:

$$\mathbf{f}^{\mathrm{min}} = \begin{bmatrix} -2f^{\mathrm{min}} \tan\theta^{\mathrm{max}} \\ 2f^{\mathrm{min}} \end{bmatrix} \quad (13)$$

$$\mathbf{f}^{\mathrm{max}} = \begin{bmatrix} 2f^{\mathrm{min}} \tan\theta^{\mathrm{max}} \\ 2f^{\mathrm{max}} \end{bmatrix} \quad (14)$$



A Hierarchical Control Scheme



Abbreviations: PP (Path Planning)
GD (Guidance)
PC (Position Controller)
AC (Attitude Controller)
CA (Control Allocation)

Control Allocation

In this simple example, by inverting equation (4) and replacing the efforts by the respective commands, we obtain the control allocation for the rotors:

$$\begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/(2l) \\ 1/2 & -1/(2l) \end{bmatrix} \begin{bmatrix} \bar{f}^c \\ \bar{\tau}^c \end{bmatrix} \tag{15}$$

where $\bar{f}^c \triangleq \|\bar{\mathbf{f}}^c_{\mathrm{G}}\|$.

On the other hand, the control allocation for the inner loop can be obtained from:

$$\bar{\theta} = \operatorname{atan} \frac{\mathbf{e}_1^{\mathrm{T}} \bar{\mathbf{f}}^c_{\mathrm{G}}}{\mathbf{e}_2^{\mathrm{T}} \bar{\mathbf{f}}^c_{\mathrm{G}}} \tag{16}$$

Flight Planning . . .

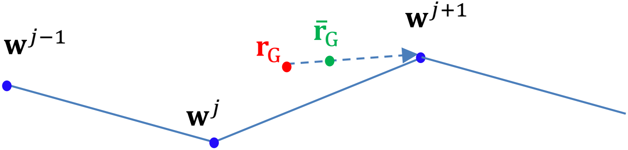
Guidance and Trajectory Planning

Some characteristics:

- Generate a desired trajectory $\bar{\mathbf{r}}_{\mathrm{G}}$ from a sequence of waypoints $\{\mathbf{w}^j\}$.
- It can be as simple as a proportinal control law that guides the MAV towards the next waypoint of the sequence:

$$\bar{\mathbf{r}}_{\mathrm{G}} = \mathbf{r}_{\mathrm{G}} + \mathbf{K}^g \left(\mathbf{w}^j - \mathbf{r}_{\mathrm{G}}\right) \tag{17}$$

where $\mathbf{K}^g$ is a gain (diagonal) matrix.



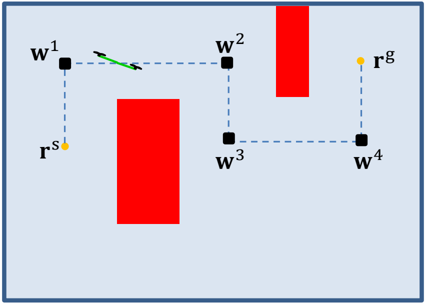
Guidance and Trajectory Planning

Here we have a simplified algorithm for the above guidance law:

Data: $\{\mathbf{w}^j\}$
$j \leftarrow 1$
for $k = 1 : end$ **do**
    $\mathbf{r}_{\mathrm{G}}(k) \leftarrow readInput$
    if $\mathbf{r}_{\mathrm{G}}(k) \in \mathcal{B}(\mathbf{w}^j)$ **then**
        $j \leftarrow j + 1$
    end
    $\bar{\mathbf{r}}_{\mathrm{G}}(k+1) \leftarrow \mathbf{r}_{\mathrm{G}}(k) + \mathbf{K}^g * (\mathbf{w}^j - \mathbf{r}_{\mathrm{G}}(k))$
    $\dots$
end

Path Planning

- It aims at computing an obstacle-free path, generally consisting of a sequence of waypoints $\mathbf{w}^j$, from the starting point $\mathbf{r}^s$ to the goal point $\mathbf{r}^g$.
- There are many methods to make it automatically (*e.g.*, visibility graph, PRM, RRT, BIT, etc.).
- In a very simple scenario, one could manually choose the apparently shortest path which is as far as possible to the obstacles, *e.g.*,



Thanks!