Quadcopter Mobile Robot
Modeling and Control

November 2017

Figure 1: Parrot AR 2.0 Drone

Consider a QuadCopter Drone (Mobile Robot) as in Figure **??**. The goal is to model and control the drone, and test the results in simulations.

1. Dynamic model

Consider the pose of the drone in Earth frame (Figure **??**):

$$P^E = [\xi^E, \eta^E]^T = [x, y, z, \phi, \theta, \psi]^T$$

where the position is given by x,y and z, while the orientation by the Euler angles ϕ (roll),θ(pitch),ψ(yaw).

The dynamic model of the drone can be developed using the Euler-Lagrange formalism (see [?] for details):

$$\begin{cases} \ddot{x} = [c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)] \frac{U_{coll}}{m} \\ \ddot{y} = [c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)] \frac{U_{coll}}{m} \\ \ddot{z} = -g + c(\phi)c(\theta)\frac{U_{coll}}{m} \\ \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} = J^{-1}(\eta) \left(\begin{bmatrix} U_\phi \\ U_\theta \\ U_\psi \end{bmatrix} - C(\eta, \dot{\eta})\dot{\eta} \right) \end{cases}$$

with the Jacobian

$$J(\eta) = \begin{bmatrix} I_x & 0 & -I_x s(\theta) \\ 0 & I_y c^2(\phi) + I_z s^2(\phi) & (I_y - I_z)c(\phi)s(\phi)c(\theta) \\ -I_x s(\theta) & (I_y - I_z)c(\phi)s(\phi)c(\theta) & I_x s^2(\theta) + I_y s^2(\phi)c^2(\theta) + I_z c^2(\phi)c^2(\theta) \end{bmatrix}$$

and the Coriolis matrix

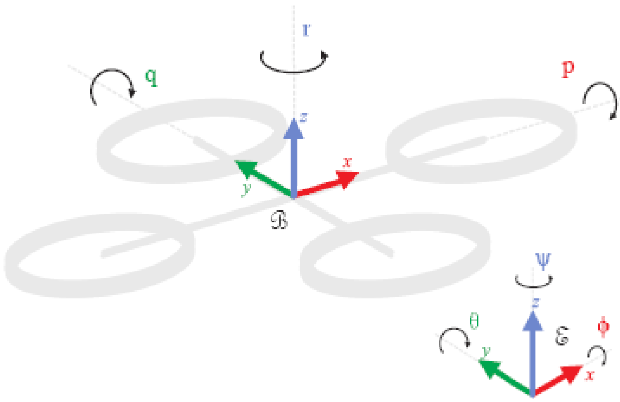

Figure 2: Quadcopter schematics (x- forward, y - lateral/sideway , z - vertical)

$$C(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & 13 \\ c_{21} & c_{22} & 23 \\ c_{31} & c_{32} & 33 \end{bmatrix}$$

$c_{11} = 0$
$c_{12} = (I_y - I_z)(\dot{\theta}c(\phi)s(\phi) + \dot{\psi}s^2(\phi)c(\theta)) + (I_z - I_y)\dot{\psi}c^2(\phi)c(\theta) - I_x\dot{\psi}c(\theta)$
$c_{13} = (I_z - I_y)\dot{\psi}c^2(\theta)s(\phi)c(\phi)$
$c_{21} = (I_z - I_y)(\dot{\theta}s(\phi)c(\phi) + \dot{\psi}s^2(\phi)c(\theta)) + (I_y - I_z)\dot{\psi}c^2(\phi)c(\theta) + I_x\dot{\psi}c(\theta)$
$c_{22} = (I_z - I_y)\dot{\phi}c(\phi)s(\phi)$
$c_{23} = -I_x\dot{\psi}s(\theta)c(\theta) + I_y\dot{\psi}s^2(\phi)s(\theta)c(\theta) + I_z\dot{\psi}c^2(\phi)s(\theta)c(\theta)$
$c_{31} = (I_y - I_z)\dot{\psi}c^2(\theta)s(\phi)c(\phi) - I_x\dot{\theta}c(\theta)$
$c_{32} = (I_z - I_y)(\dot{\theta}c(\phi)s(\phi)s(\theta) + \dot{\phi}s^2(\phi)c(\theta)) + (I_y - I_z)\dot{\phi}c^2(\phi)c(\theta)$
$+I_x\dot{\psi}s(\theta)c(\theta) - I_y\dot{\psi}s^2(\phi)s(\theta)c(\theta) - I_z\dot{\psi}c^2(\phi)s(\theta)c(\theta)$
$c_{33} = (I_y - I_z)\dot{\phi}c^2(\theta)s(\phi)c(\phi) - I_y\dot{\theta}s^2(\phi)s(\theta)c(\theta) - I_z\dot{\theta}c^2(\phi)s(\theta)c(\theta) + I_x\dot{\theta}s(\theta)c(\theta)$

The control inputs are the torques on the three rotational axes U_ϕ,U_θ,U_ψ, and the collective force U_{coll}.
In near-hovering operating modes, the model can be greatly simplified by considering that for each angle α we can approximate $sin(\alpha) = \alpha$ and $cos(\alpha) = 1$, and that $U_{coll} = mg + \Delta U_{coll}$ ([?]):

$$\begin{cases} \ddot{x} = \theta g \\ \ddot{y} = -\phi g \\ \ddot{z} = \frac{\Delta U_{coll}}{m} \end{cases} \qquad \begin{cases} \ddot{\phi} = \frac{1}{I_x}U_\phi \\ \ddot{\theta} = \frac{1}{I_y}U_\theta \\ \ddot{\psi} = \frac{1}{I_z}U_\psi \end{cases}$$

This suggest that a linear control could be sufficient when the drone is in near-hovering mode.

2. Linear control

Consider the model rewritten in nonlinear state space form:

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

were the state is $\mathbf{x} = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$ and the input $\mathbf{u} = [U_{coll}, U_\phi, U_\theta, U_\psi]^T$.
Let the equilibrium point in hovering mode be $\mathbf{x}^e = [x^e, y^e, z^e, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$, with inputs $\mathbf{u}^e = [U^e_{coll}, 0, 0, 0]^T$. The linearized model is obtained as:

$$\dot{\mathbf{x}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{(\mathbf{x}^e, \mathbf{u}^e)} \cdot \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{u}}|_{(\mathbf{x}^e, \mathbf{u}^e)} \cdot \mathbf{u}$$

.

For the parameters of the Parrot drone $I_x = 0.002, I_y = 0.0016, I_z = 0.0035, m = 0.4472, g = 9.8$. the linearized model becomes:

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

with

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 9.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -9.8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 2.2 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 \\ 0 & 0 & 625 & 0 \\ 0 & 0 & 0 & 285.7 \end{bmatrix}$$

We will first design a state feedback control $\mathbf{u} = -\mathbf{K} \cdot \mathbf{x}$ to stabilize the system. We impose the following closed loop poles $p = [-20, -20.5, -21, -2.05, -2.1, -2.15, -21.5, -22, -22.5, -2.2, -2.25, -2.3]$, that ensure that the system is stable, without oscillations, and with a relative fast response.

The feedback gain matrix $\mathbf{K}$ can be calculated by using the *place* function in Matlab:

$$K = \begin{bmatrix} 24.7267 & 0.0002 & 22.1294 & -0.0002 & 54.2482 & 0.0000 & 22.3994 & 0.0001 & 10.8415 & -0.0000 & 2.0236 & 0.0000 \\ -0.0000 & -0.3967 & -0.0000 & 1.2228 & -0.0000 & -0.0000 & -0.0000 & -0.4116 & -0.0000 & 0.0915 & -0.0000 & -0.0000 \\ 0.3998 & 0.0000 & 0.0000 & -0.0000 & 1.0992 & 0.0000 & 0.3919 & 0.0000 & 0.0000 & -0.0000 & 0.0776 & 0.0000 \\ -0.0000 & 0.0073 & -0.0000 & -0.0034 & -0.0000 & 0.1435 & -0.0000 & 0.0041 & -0.0000 & -0.0001 & -0.0000 & 0.0772 \end{bmatrix}$$

Next, we add an output feedback loop after the positions x, y, z (tracking) as in Figure **??**. The new control signal now becomes:

$$\mathbf{u} = -\mathbf{K} \cdot \mathbf{x} + \mathbf{u}_o$$

with

$$\mathbf{u}_o = [u_z, u_y, u_x, 0]^T$$

and

$$u_x = \frac{k_{ix}}{s}(r_x - x), u_y = \frac{k_{iy}}{s}(r_y - y), u_z = \frac{k_{iz}}{s}(r_z - z), ref = [r_x, r_y, r_z]^T$$

The integrator gains can be tuned experimentally in order to obtain good tracking performances.

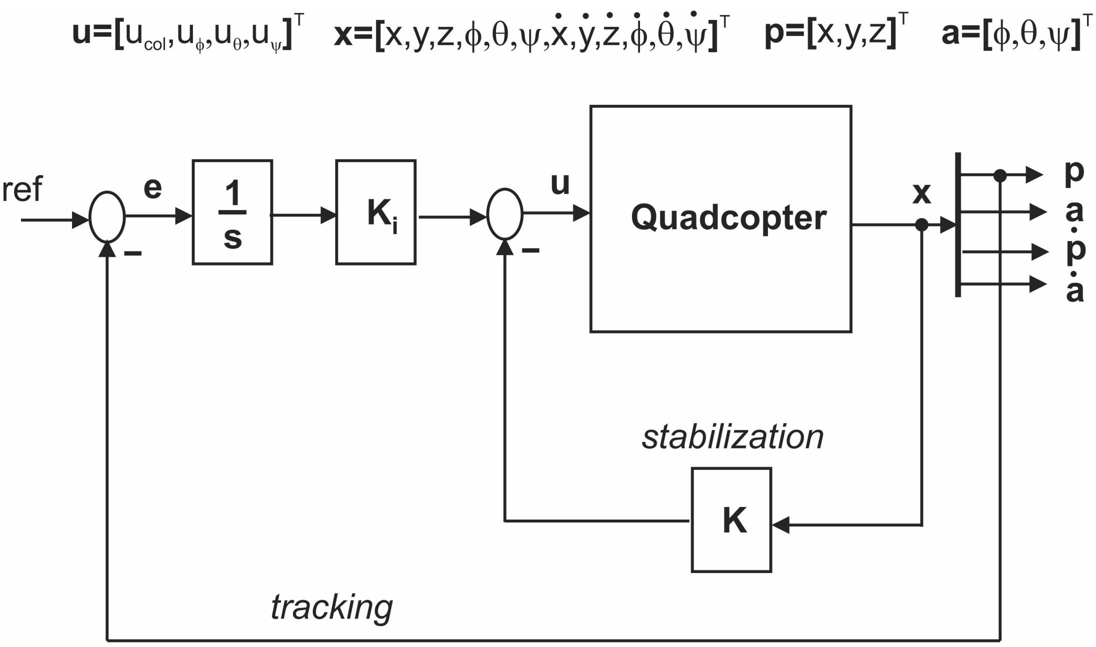

Figure 3: Linear control of QuadCopter

Bibliography

[1] Mark W. Spong, Seth Hutchinson, M. Vidyasagar, *Robot Modeling and Control*, Wiley, 2006.

[2] Mathe Antal Koppany, *Nonlinear Control for Commercial Drones in Autonomous Railway Maintenance*, PhD Thesis, Technical University of Cluj-Napoca, 2016.

[3] Theys B., Dimitriadis G., Andrianne T., Hendrick P, De Schutter J., *Wind Tunnel Testing of a VTOL MAV Propeller in Tilted Operating Mode*, Proceedings of the IEEE International Conference on Unmanned Aircraft Systems (ICUAS), USA, 2014, pp. 1064 – 1072.