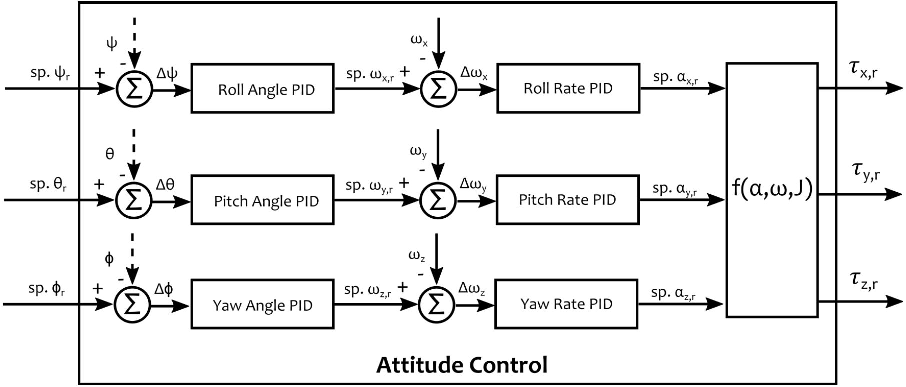
Attitude Control



Attitude Control

- the purpose of Attitude Control is to stabilize the quadrotor at the given orientation set point $(\psi_r, \theta_r, \phi_r)$
- the setpoint angles (ψ_r, θ_r) are generated by the position control block, while the ϕ_r angle is coming from the operator (or from a higher level supervisor control/mission planner)
- the Attitude Control should be fast (suggestion: rate control at over 200 Hz, angle control at 100 Hz)

Angle Control

- The model of roll angle dynamics is $\dot{\psi}_r = \omega_x$. We employ PID control, where the error is $\Delta\psi = \psi_r - \psi$.

$$\dot{\psi}_r = \underbrace{\omega_{r,x}}_{u} = \mathrm{PID}(\Delta\psi) = k_{p,\psi}\Delta\psi + k_{i,\psi}\int_0^t \Delta\psi(\tau)d\tau + k_{d,\psi}\frac{\Delta\psi}{dt} \tag{1a}$$

And similarly for the other two angles:

$$\dot{\theta}_r = \underbrace{\omega_{r,y}}_{u} = \mathrm{PID}(\Delta\theta) = k_{p,\theta}\Delta\theta + k_{i,\theta}\int_0^t \Delta\theta(\tau)d\tau + k_{d,\theta}\frac{\Delta\theta}{dt} \tag{1b}$$

$$\dot{\phi}_r = \underbrace{\omega_{r,z}}_{u} = \mathrm{PID}(\Delta\phi) = k_{p,\phi}\Delta\phi + k_{i,\phi}\int_0^t \Delta\phi(\tau)d\tau + k_{d,\phi}\frac{\Delta\phi}{dt} \tag{1c}$$

Yaw Error

Because the yaw is not restricted we have to take into consideration the entire $[-\pi, \pi]$ range, and handle the discontinuity $\pi, -\pi$ in the angle error expression. This is important to make possible a smooth 2π range of rotation:

$$\Delta\phi = \phi_r - \phi$$

$$\text{if } (\Delta\phi > \pi):$$

$$\quad \Delta\phi = -(2\pi - \Delta\phi)$$

$$\text{elif } (\Delta\phi < -\pi):$$

$$\quad \Delta\phi = 2\pi + \Delta\phi$$

Rate Control

- The purpose of Rate Control is to stabilize the drones angular velocity at the set point $(\omega_{x,r}, \omega_{y,r}, \omega_{z,r})$
- The model of roll angle rate dynamics is $\dot{\omega}_x = \alpha_x$ under small angle assumption.
- We employ PID control, where the error is $\Delta\omega_x = \omega_{x,r} - \omega_x$, using the gyroscope measurements.

$$\dot{\omega}_x = \underbrace{\alpha_{r,x}}_{u} = \text{PID}(\Delta\omega_x) = k_{p,\omega_x}\Delta\omega_x + k_{i,\omega_x}\int_0^t \Delta\omega_x(\tau)d\tau + k_{d,\omega_x}\frac{\Delta\omega_x}{dt} \tag{2a}$$

And similarly for the other two rates:

$$\dot{\omega}_y = \underbrace{\alpha_{r,y}}_{u} = \text{PID}(\Delta\omega_y) = k_{p,\omega_y}\Delta\omega_y + k_{i,\omega_y}\int_0^t \Delta\omega_y(\tau)d\tau + k_{d,\omega_y}\frac{\Delta\omega_y}{dt} \tag{2b}$$

$$\dot{\omega}_z = \underbrace{\alpha_{r,z}}_{u} = \text{PID}(\Delta\omega_z) = k_{p,\omega_z}\Delta\omega_z + k_{i,\omega_z}\int_0^t \Delta\omega_z(\tau)d\tau + k_{d,\omega_z}\frac{\Delta\omega_z}{dt} \tag{2c}$$

Torque Model Inversion

From

$$\boldsymbol{\alpha}^b = \left(\mathbf{J}^b\right)^{-1}\left(\boldsymbol{\tau}^b - \left[\boldsymbol{\omega}^b\right]_\times \mathbf{J}^b \boldsymbol{\omega}^b\right) \tag{3}$$

we obtain that

$$\boxed{\boldsymbol{\tau}^b = \mathbf{J}^b \boldsymbol{\alpha}^b + \left[\boldsymbol{\omega}^b\right]_\times \mathbf{J}^b \boldsymbol{\omega}^b} \tag{4}$$

Rate Control Tuning

- `main_2_sim_tune_ratecontrol.py`
- `controllers.py`
- `pid.py`

Angle Control Tuning

- `main_3_sim_perception_tune_anglecontrol.py`
- `spkf.py`
- `controllers.py`
- `pid.py`