Contents

1 Trajectory Tracking Control - Theory

1.1 Task 2.1

The linear dynamic of the quadrotor, neglecting the drag term, can be described by:

$$\begin{cases} \dot{\boldsymbol{p}} = \boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{v} \\ m\dot{\boldsymbol{v}} = m(\boldsymbol{\omega} \times \boldsymbol{v}) + mG\boldsymbol{R}(\boldsymbol{\lambda})^T\boldsymbol{e_3} - u_w\boldsymbol{e_3} \end{cases} \tag{1}$$

In order to calculate the acceleration we have to derive the first equation such as:

$$\ddot{\boldsymbol{p}} = \dot{\boldsymbol{R}}(\boldsymbol{\lambda})\boldsymbol{v} + \boldsymbol{R}(\boldsymbol{\lambda})\dot{\boldsymbol{v}} \tag{2}$$

We can now substitute in this expression $\dot{\boldsymbol{v}}$ and $\dot{\boldsymbol{R}}(\boldsymbol{\lambda})$ since we know that

- $\dot{\boldsymbol{v}} = \boldsymbol{\omega} \times \boldsymbol{v} + G\boldsymbol{R}(\boldsymbol{\lambda})^T\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{e_3}$
- $\dot{\boldsymbol{R}}(\boldsymbol{\lambda}) = -\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{S}_\times(\boldsymbol{\omega})$

where $\boldsymbol{S}_\times(\boldsymbol{\omega})$ is the skew symmetric matrix generated from the $\boldsymbol{\omega}$ vector such that $\boldsymbol{S}_\times(\boldsymbol{\omega})\boldsymbol{v} = \boldsymbol{\omega} \times \boldsymbol{v}$. In the end we will have:

$$\begin{aligned} \ddot{\boldsymbol{p}} &= \dot{\boldsymbol{R}}(\boldsymbol{\lambda})\boldsymbol{v} + \boldsymbol{R}(\boldsymbol{\lambda})\dot{\boldsymbol{v}} \\ &= -\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{S}_\times(\boldsymbol{\omega})\boldsymbol{v} + \boldsymbol{R}(\boldsymbol{\lambda})\left[\boldsymbol{\omega} \times \boldsymbol{v} + G\boldsymbol{R}(\boldsymbol{\lambda})^T\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{e_3}\right] \\ &= -\cancel{\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{\omega} \times \boldsymbol{v}} + \cancel{\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{\omega} \times \boldsymbol{v}} + G\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{e_3} \\ &= G\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{e_3} \end{aligned} \tag{3}$$

1.2 Task 2.2

The drone dynamics can be modifyed introducing u^*:

$$\begin{aligned} \ddot{\boldsymbol{p}} &= G\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{R}(\lambda)\boldsymbol{e_3} \\ &= G\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{R_z}(\psi)\boldsymbol{R_y}(\theta)\boldsymbol{R_x}(\phi)\boldsymbol{e_3} \\ &= G\boldsymbol{e_3} - \frac{1}{m}\boldsymbol{R_z}(\psi)\underbrace{[\boldsymbol{R_y}(\theta)\boldsymbol{R_x}(\phi)u_w\boldsymbol{e_3}]}_{\boldsymbol{u^*}} \\ &= G\boldsymbol{e_3} - \frac{1}{m}\boldsymbol{R_z}(\psi)\boldsymbol{u^*} \end{aligned} \tag{4}$$

1.3 Task 2.3

Supposing position and velocity error to be defined as following:

$$\begin{cases} \tilde{\boldsymbol{p}} = \boldsymbol{p} - \boldsymbol{p}_d \\ \tilde{\boldsymbol{v}} = \dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_d \end{cases} \tag{5}$$

The error dynamics will be:

$$\begin{cases} \dot{\tilde{\boldsymbol{p}}} = \dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_d = \tilde{\boldsymbol{v}} \\ \dot{\tilde{\boldsymbol{v}}} = \ddot{\boldsymbol{p}} - \ddot{\boldsymbol{p}}_d = \boldsymbol{u} \end{cases} \tag{6}$$

On the other hand we already know that $\ddot{\boldsymbol{p}} = G\boldsymbol{e_3} - \frac{1}{m}\boldsymbol{R_z}(\psi)\boldsymbol{u^*}$, so we can write an expression for $\boldsymbol{u} = \boldsymbol{u}(\boldsymbol{u}^*)$:

$$\boldsymbol{u} = G\boldsymbol{e_3} - \frac{1}{m}\boldsymbol{R_z}(\psi)\boldsymbol{u^*} - \ddot{\boldsymbol{p}}_d \tag{7}$$

1.4 Task 2.4

Defining the error system as:

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \tilde{\boldsymbol{p}} \\ \tilde{\boldsymbol{v}} \end{bmatrix} \tag{8}$$

we will have a new system of the type:

$$\dot{\tilde{\boldsymbol{x}}} = \boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u} \tag{9}$$

where

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0}_{[3\times3]} & \boldsymbol{I}_{[3\times3]} \\ \boldsymbol{0}_{[3\times3]} & \boldsymbol{0}_{[3\times3]} \end{bmatrix} \tag{10}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0}_{[3\times3]} \\ \boldsymbol{I}_{[3\times3]} \end{bmatrix} \tag{11}$$

The proposed control law requires to first calculate a symmetrix positive definite matrix $\boldsymbol{P}$ as the solution of Riccati equation:

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0} \tag{12}$$

where the dimentions of the matrices in the equation are
$\boldsymbol{R}_{[3\times3]} = \rho\boldsymbol{I}_{[3\times3]}$; $\boldsymbol{R}^{-1}_{[3\times3]} = \frac{1}{\rho}\boldsymbol{I}_{[3\times3]}$; $\boldsymbol{P}_{[6\times6]}$; $\boldsymbol{Q}_{[6\times6]}$;

Then to compute the optimal gains:

$$\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} \tag{13}$$

where in this case the gain matrix is:
$\boldsymbol{K}_{[3\times 6]}$;
In the end our new input for the error system will be given by:

$$\boldsymbol{u} = -\boldsymbol{K}\tilde{\boldsymbol{x}} \tag{14}$$

1.5 Task 2.5

Given the scalar function $V_1 = \tilde{\boldsymbol{x}}^T\boldsymbol{P}\tilde{\boldsymbol{x}}$, its time derivative will be:

$$\dot{V}_1 = \dot{\tilde{\boldsymbol{x}}}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\dot{\tilde{\boldsymbol{x}}} \tag{15}$$

We know the dynamics of our system from equation (9) and we also know the form of the input $\boldsymbol{u}$ from the control law already defined in equations (14) (13). So we can apply the substitutions highlighting the fact that in the transposition process we will not change notation if a matrix is symmetric (the case of $\boldsymbol{R}$ and $\boldsymbol{P}$).

For sake of clarity:

$$\boldsymbol{u} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} \tag{16}$$

$$\boldsymbol{u}^T = -\tilde{\boldsymbol{x}}^T\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1} \tag{17}$$

$$\begin{aligned}
\dot{V}_1 &= \left(\boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u}\right)^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\left(\boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{u}\right) \\
&= \left(\tilde{\boldsymbol{x}}^T\boldsymbol{A}^T + \boldsymbol{u}^T\boldsymbol{B}^T\right)\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\left(\boldsymbol{A}\tilde{\boldsymbol{x}} - \boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}}\right) \\
&= \left(\tilde{\boldsymbol{x}}^T\boldsymbol{A}^T - \tilde{\boldsymbol{x}}^T\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\right)\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\left(\boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} \\
&= \tilde{\boldsymbol{x}}^T\left(\boldsymbol{A}^T\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} + \dots \\
&= \tilde{\boldsymbol{x}}^T\left(\underbrace{\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}}_{\text{Riccati Equation}} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} \\
&= \tilde{\boldsymbol{x}}^T\left(-\boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} \\
&= -\tilde{\boldsymbol{x}}^T\left(\boldsymbol{Q} + \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}}
\end{aligned} \tag{18}$$

Now if we insert our specific $\boldsymbol{R}$ matrix we will get:

$$\begin{aligned}\dot{V}_1 &= -\tilde{\boldsymbol{x}}^T \left(\boldsymbol{Q} + \frac{1}{\rho} \boldsymbol{P}\boldsymbol{B}\boldsymbol{I}\boldsymbol{B}^T\boldsymbol{P} \right) \tilde{\boldsymbol{x}} \\ &= -\tilde{\boldsymbol{x}}^T \left(\boldsymbol{Q} + \frac{1}{\rho} \boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P} \right) \tilde{\boldsymbol{x}}\end{aligned} \tag{19}$$

Said $\boldsymbol{Q}^*$ to be:

$$\boldsymbol{Q}^* = \boldsymbol{Q} + \frac{1}{\rho} \boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P} \tag{20}$$

Inserting in this equation the tuned values of $\boldsymbol{Q}$ and the $\boldsymbol{P}$ matrix obtained from the LQR algorithm, we got a positive determinant $\boldsymbol{Q}^*$ matrix. This means that all its eigenvalues are positive and different from zero. According to the Lyapunov stability theory, thanks to the minus sign present in the $\dot{V}_1$ expression, the V_1 scalar function show us the asymptotic stability of the origin equilibrium state. (INSERIRE MATRICI MAGARI BABBETIBOBBETI)

1.6 Task 2.6

Starting from the previously obtained definition of $\boldsymbol{u}^*$ with some simple substitutions we get:

$$\begin{aligned}\boldsymbol{u}^* &= m\boldsymbol{R}_z^T \left(G\boldsymbol{e_3} - \ddot{\boldsymbol{p}}_d - \underbrace{\boldsymbol{u}} \right) \\ &= m\boldsymbol{R}_z^T \left(G\boldsymbol{e_3} - \ddot{\boldsymbol{p}}_d + \underbrace{\boldsymbol{K}}\tilde{\boldsymbol{x}} \right) \\ &= m\boldsymbol{R}_z^T \left(G\boldsymbol{e_3} - \ddot{\boldsymbol{p}}_d + \frac{1}{\rho}\boldsymbol{I}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} \right) \\ &= m\boldsymbol{R}_z^T \left(G\boldsymbol{e_3} - \ddot{\boldsymbol{p}}_d + \frac{1}{\rho}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} \right)\end{aligned} \tag{21}$$

1.7 Task 2.7

Supposing now the presence of an unknown disturbance $\boldsymbol{d}$ in the system and supposing that the controller uses an estimation of that disturbance $\hat{\boldsymbol{d}}$, the dynamics of our system will be modified as following:

$$\begin{cases} \ddot{\boldsymbol{p}} = G\boldsymbol{e_3} - \frac{u_w}{m}\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{e_3} + \boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{d} \\ \boldsymbol{u}^* = m\boldsymbol{R}_z^T \left(G\boldsymbol{e_3} - \ddot{\boldsymbol{p}}_d + \frac{1}{\rho}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \boldsymbol{R}(\boldsymbol{\lambda})\hat{\boldsymbol{d}} \right) \end{cases} \tag{22}$$

We know that we can rewrite our acceleration as

$$\begin{aligned}\ddot{\boldsymbol{p}} &= G\boldsymbol{e_3} - \frac{1}{m}\boldsymbol{R}_z(\psi)\boldsymbol{u}^* + \boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{d} \\ &= \cancel{G\boldsymbol{e_3}} - \cancel{G\boldsymbol{e_3}} + \ddot{\boldsymbol{p}}_d - \frac{1}{\rho}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} - \boldsymbol{R}(\boldsymbol{\lambda})\hat{\boldsymbol{d}} + \boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{d} \\ \ddot{\boldsymbol{p}} - \ddot{\boldsymbol{p}}_d &= -\frac{1}{\rho}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \boldsymbol{R}(\boldsymbol{\lambda})(\boldsymbol{d} - \hat{\boldsymbol{d}}) \\ \ddot{\boldsymbol{p}} - \ddot{\boldsymbol{p}}_d &= -\frac{1}{\rho}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}} = \boldsymbol{u}\end{aligned} \tag{23}$$

Substituting this new expression of $\boldsymbol{u}$ inside the steady state equation we can obtain:

$$\begin{aligned}\dot{\tilde{\boldsymbol{x}}} &= \boldsymbol{A}\tilde{\boldsymbol{x}} + \boldsymbol{B}\left[-\frac{1}{\rho}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}}\right] \\ &= \left(\boldsymbol{A} - \frac{1}{\rho}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}}\end{aligned} \tag{24}$$

1.8 Task 2.8

In this task we will derive the adaptation law for the disturbance. We have the following Lyapunov function candidate:

$$V_2 = V_1 + \frac{1}{K_d}\tilde{\boldsymbol{d}}^2 \tag{25}$$

We can compute the time derivative of V_2 remembering the expression obtained in Task 2.5 for $\dot{V}_1$:

$$\dot{V}_2 = \dot{\tilde{\boldsymbol{x}}}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\dot{\tilde{\boldsymbol{x}}} + \frac{1}{K_d}\dot{\tilde{\boldsymbol{d}}}^T\tilde{\boldsymbol{d}} + \frac{1}{K_d}\tilde{\boldsymbol{d}}^T\dot{\tilde{\boldsymbol{d}}} \tag{26}$$

The next step is to insert the expression for $\dot{\tilde{\boldsymbol{x}}}$ by using equation (24).

$$\begin{aligned}\dot{V}_2 = \left[\left(\boldsymbol{A} - \frac{1}{\rho}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}}\right]^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\left[\left(\boldsymbol{A} - \frac{1}{\rho}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P}\right)\tilde{\boldsymbol{x}} + \boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}}\right] + \\ \frac{1}{K_d}\dot{\tilde{\boldsymbol{d}}}^T\tilde{\boldsymbol{d}} + \frac{1}{K_d}\tilde{\boldsymbol{d}}^T\dot{\tilde{\boldsymbol{d}}}\end{aligned} \tag{27}$$

Multiplying $\boldsymbol{P}\tilde{\boldsymbol{x}}$ in to the brackets and rearranging the terms we obtain the following equation:

$$\dot{V}_2 = \tilde{\boldsymbol{x}}^T \Bigg(\underbrace{\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} - \frac{1}{\rho}\boldsymbol{P}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P}}_{-\boldsymbol{Q}} \Bigg) \tilde{\boldsymbol{x}} - \frac{1}{\rho}\tilde{\boldsymbol{x}}^T\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \\ \tilde{\boldsymbol{d}}^T\boldsymbol{R}(\boldsymbol{\lambda})^T\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}} + \frac{1}{K_d}\dot{\tilde{\boldsymbol{d}}}^T\tilde{\boldsymbol{d}} + \frac{1}{K_d}\tilde{\boldsymbol{d}}^T\dot{\tilde{\boldsymbol{d}}} \tag{28}$$

According to equation (20) the first two term of this expression are equal to $\boldsymbol{Q}^*$. Moreover, if we define the disturbance error as $\tilde{\boldsymbol{d}} = \boldsymbol{d} - \hat{\boldsymbol{d}}$ and since $\boldsymbol{d}$ is presumed to be constant, we have the following relation $\dot{\tilde{\boldsymbol{d}}} = -\dot{\hat{\boldsymbol{d}}}$.

$$\dot{V}_2 = -\tilde{\boldsymbol{x}}^T\boldsymbol{Q}^*\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{d}}^T\boldsymbol{R}(\boldsymbol{\lambda})^T\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{x}}^T\boldsymbol{P}\boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda})\tilde{\boldsymbol{d}} - \frac{1}{K_d}\dot{\hat{\boldsymbol{d}}}^T\tilde{\boldsymbol{d}} - \frac{1}{K_d}\tilde{\boldsymbol{d}}^T\dot{\hat{\boldsymbol{d}}} \tag{29}$$

If $\dot{\hat{\boldsymbol{d}}}$ is chosen such that it cancels the terms containing both $\tilde{\boldsymbol{x}}$ and $\tilde{\boldsymbol{d}}$ we get $\dot{V}_2 \leqslant 0$ and the following adaptation law:

$$\dot{V}_2 = -\tilde{\boldsymbol{x}}^T\boldsymbol{Q}^*\tilde{\boldsymbol{x}} \leqslant 0 \tag{30}$$

$$\begin{aligned} \dot{\hat{\boldsymbol{d}}} &= K_d\boldsymbol{R}(\boldsymbol{\lambda})\boldsymbol{B}^T\boldsymbol{P}\tilde{\boldsymbol{x}} \\ &= K_d\boldsymbol{R}(\boldsymbol{\lambda})\rho\boldsymbol{K}\tilde{\boldsymbol{x}} \end{aligned} \tag{31}$$

1.9 Task 2.9

Building up a new system with the disturbance as a part of the state vector we get a new A matrix:

$$\begin{bmatrix} \dot{\tilde{\boldsymbol{x}}} \\ \dot{\tilde{\boldsymbol{d}}} \end{bmatrix} = \begin{bmatrix} \left(\boldsymbol{A} - \frac{1}{\rho}\boldsymbol{B}\boldsymbol{B}^T\boldsymbol{P}\right) & \boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda}) \\ K_d\boldsymbol{R}(\boldsymbol{\lambda})\rho\boldsymbol{K} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{x}} \\ \tilde{\boldsymbol{d}} \end{bmatrix} = \boldsymbol{0} \tag{32}$$

Considering the equilibrium condition in which the state derivative is null we can see from the second line of the matrix that $K_d\boldsymbol{R}(\boldsymbol{\lambda})\rho\boldsymbol{K} \cdot \tilde{\boldsymbol{x}} = 0$. This equation is satisfied only for $\tilde{\boldsymbol{x}} = 0$ since there is no combination of the three euler angles $\boldsymbol{\lambda}$ that can bring simultaneously all the $\boldsymbol{R}(\boldsymbol{\lambda})$ components to zero. The same type of consideration can now be done for the first line where we already know that $\tilde{\boldsymbol{x}} = 0$ so the first equation becomes $\boldsymbol{B}\boldsymbol{R}(\boldsymbol{\lambda}) \cdot \tilde{\boldsymbol{d}} = 0$. For no combination of $\boldsymbol{\lambda}$ is going to bring to zero all the $\boldsymbol{R}(\boldsymbol{\lambda})$ matrix components, we will have that the only solution is for $\tilde{\boldsymbol{d}} = 0$. Thus $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}) = (0, 0)$ is an equilibrium point for our system.
Considering V_2 as a scalar candidate Lyapunov function, we can clearly see that as $||\boldsymbol{x}|| \to \infty \implies V_2 \to \infty$, where $\boldsymbol{x}$ is the new state defined as:

$$\boldsymbol{x} = \begin{bmatrix} \dot{\tilde{\boldsymbol{x}}} \\ \dot{\tilde{\boldsymbol{d}}} \end{bmatrix} \tag{33}$$

V_2 is continuously differentiable, radially unbounded and positive definite function such that $\dot{V}_2 \leqslant 0$ for all $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}) \in R^n$. Let $S = \{(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{d}}) \in R^n \mid \dot{V}_2 = 0\}$: since our expression for the derivative of our Lyapunov function is $\dot{V}_2 = -\tilde{\boldsymbol{x}}^T \boldsymbol{Q}^* \tilde{\boldsymbol{x}}$ we may have regions, apart from the origin, where $\dot{V}_2 = 0$ (i.e. regions where $\tilde{\boldsymbol{x}} = 0$ but $\tilde{\boldsymbol{d}} \neq 0$). However, since none of these regions contains any equilibrium point (the origin is the only equilibrium point), thanks to La Salle theorem we can state that the origin is globally asymptotically stable.

1.10 Task 2.10

We can model the yaw motion as a first order system with transfer function generally given by:

$$G(s) = \frac{\psi}{u_\psi} = \frac{1}{s + a} \tag{34}$$

If we close the loop with a simple proportional controller acting on the difference between the reference input and the actual yaw angle we will obtain a closed loop transfer function of the type:

$$G(s) = \frac{\psi}{u_\psi} = \frac{K_p}{s + a + K_p} \tag{35}$$

This transfer function, according to the finite value theorem, does not comply with the requirement of zero steady state error. In order to have this characteristic we may insert in our loop an integrator, basically implementing a PI controller. The closed loop transfer function of the system will consquently be:

$$G(s) = \frac{\psi}{u_\psi} = \frac{sK_p + K_i}{s^2 + (a + K_p)s + K_i} \tag{36}$$

Analizing the final value of this transfer function with an arbitrary step input of H amplitude:

$$\lim_{t \to \infty} \psi(t) = \lim_{s \to 0} s \cdot \; G(s) \cdot \frac{H}{s} = \lim_{s \to 0} s \cdot \frac{sK_p + K_i}{s^2 + (a + K_p)s + K_i} \cdot \frac{H}{s} = H \tag{37}$$

Zero steady state error in the end.

2 Trajectory Tracking Control - Simulations

2.1 Controller

2.1.1 Pitch and Roll

In order to compile the outer loop controller in the Simulink model we decide to initially calculate the LQR gains for our system with the specific *lqr* Matlab function. These gains are imported

manually inside the controller function.

$$\boldsymbol{K} = \begin{bmatrix} 2.4495 & 0 & 0 & 2.2269 & 0 & 0 \\ 0 & 3.0000 & 0.0000 & 0 & 2.4617 & 0.0000 \\ 0 & 0.0000 & 3.0000 & 0 & 0.0000 & 2.4617 \end{bmatrix} \tag{38}$$

We define the error state vector as:

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{p} - \boldsymbol{p}_d \\ \boldsymbol{v} - \boldsymbol{v}_d \end{bmatrix} \tag{39}$$

where $\boldsymbol{p}$ and $\boldsymbol{v}$ are the position and the velocity vector acquired from the states of the drone, while $\boldsymbol{p}_d$ and $\boldsymbol{v}_d$ are the position and velocity vector acquired from the trajectory definition function.

Now the input $\boldsymbol{u}$ of our system will be:

$$\boldsymbol{u} = -\boldsymbol{K}\tilde{\boldsymbol{x}} \tag{40}$$

and we can calculate immediately the numerical values for the vector $\boldsymbol{u}^*$ from:

$$\boldsymbol{u}^* = m\boldsymbol{R}_{\boldsymbol{z}}^{-1}(\psi)\left(-\boldsymbol{u} - \ddot{\boldsymbol{p}}_d + G\boldsymbol{e}_3\right) \tag{41}$$

where ψ is known since it comes from the state vector.

On the other hand, recalling the other expression of $\boldsymbol{u}^*$:

$$\boldsymbol{u}^* = \boldsymbol{R}_{\boldsymbol{y}}(\theta)\boldsymbol{R}_{\boldsymbol{x}}(\phi)u_w\boldsymbol{e}_{\boldsymbol{3}} \tag{42}$$

we need to solve this relation in the unknowns θ_r, ϕ_r and u_w as functions of the three components of the $\boldsymbol{u}^*$ vector.

$$\begin{aligned} u_1^* &= u_w \cos\phi \sin\theta \\ u_2^* &= -u_w \sin\phi \\ u_3^* &= u_w \cos\phi \cos\theta \end{aligned} \tag{43}$$

the results will be:

$$\begin{aligned} \theta_r &= \arctan\left(\frac{u_1^*}{u_3^*}\right) \\ \phi_r &= \arctan\left(-\cos\theta\frac{u_2^*}{u_3^*}\right) \\ u_w &= \frac{u_3^*}{\cos\theta\cos\phi} \end{aligned} \tag{44}$$

2.1.2 Height

The height controller was simply implemented thanks to the given relation:

$$w_r = -k_w(\boldsymbol{p}_3 - \boldsymbol{p}_{d,3}) \tag{45}$$

2.1.3 Yaw Rate

The yaw rate controller was implemented as a PI controller in order to satisfy the zero steady state error specification. In order to include the integral part we used the fourth element of the ξ vector already present in this function. This vector undergoes an integration every time the function is called.
The yaw angle error is defined as:

$$\psi_e = \psi_d - \psi \tag{46}$$

and the ξ vector:

$$\xi = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \psi_e \end{bmatrix} \tag{47}$$

Our control law will be:

$$\dot{\psi}_r = k_p\psi_e + k_i\xi_4 \tag{48}$$

2.1.4 Disturbance

Thanks to our adaptation law we can have the expression of $\dot{\hat{\boldsymbol{d}}}$. Using the first three slots of the vector ξ we can have this value integrated giving us the estimated value of $\hat{\boldsymbol{d}}$. This value can be used to determine a slightly different value for the $\boldsymbol{u}^*$ components, refining the quality of our controller:

$$\xi = \begin{bmatrix} \dot{\hat{d}}_1 \\ \dot{\hat{d}}_2 \\ \dot{\hat{d}}_3 \\ \psi_e \end{bmatrix} \tag{49}$$

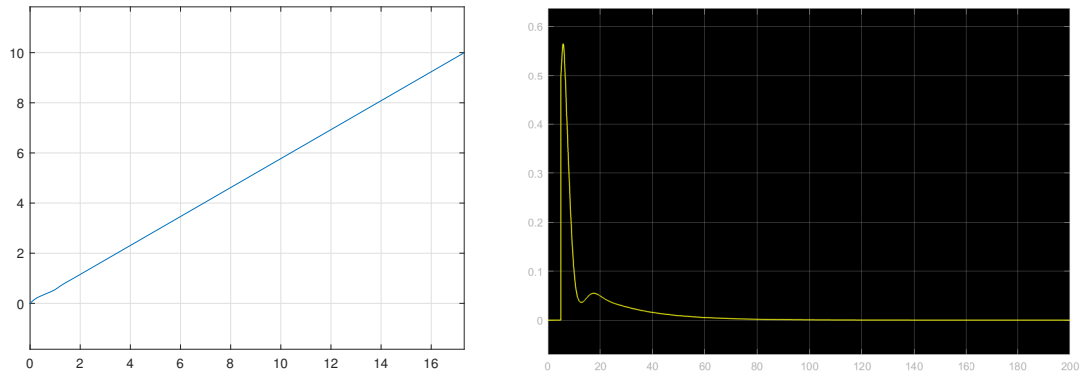
$$\boldsymbol{u}^* = m\boldsymbol{R}_z^{-1}(\psi)\left(-\boldsymbol{u} - \ddot{\boldsymbol{p}}_d + G\boldsymbol{e}_3 + \boldsymbol{R}(\lambda)\hat{\boldsymbol{d}}\right) \tag{50}$$

2.2 Trajectories

2.2.1 Line

$$\begin{aligned}
\dot{\boldsymbol{p}}_d &= \begin{bmatrix} \dot{x} \\ \sqrt{0.1^2 - \dot{x}^2} \\ 0 \end{bmatrix} \\
\boldsymbol{p}_d &= \boldsymbol{p}_0 + \dot{\boldsymbol{p}}_d \cdot dt \\
\ddot{\boldsymbol{p}}_d &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
\psi_d &= 0
\end{aligned} \tag{51}$$

The line trajectory required, as first, the definition of the velocity vector: we decided to keep as a degree of freedom the velocity along the x axis: this value should be chosen between 0 and 0.1. The value of the velocity along y is calculated thanks to the vector relation $||\boldsymbol{v}|| = \sqrt{\dot{x}^2 + \dot{y}^2}$. The signs of the velocities are totally left to user's choice: we just assumed to fly in the positive direction for both x and y but no constraint prevents us to fly in each of the four quadrants.
The position is defined thanks to the variable dt and the value p_0 (defined at each step time as the position acquired from the states vector).
Obviously the acceleration is set to zero as well for the yaw angle.



(a) Line Trajectory

(b) Position Error

Figure 1: Line Trajectory

We can see in Fig. (1) that after a short initial period in which the drone is oscillating around the reference, the line trajectory is completely accomplished. The position error after the first transient part decrease to zero. This is due to the adaption law estimating the disturbance combined with our controller that we have shown has a globally asymptotically stable origin, the error will go to zero.

2.2.2 Circle

$$
\begin{aligned}
\omega &= 0.1 \\
\boldsymbol{p}_d &= \begin{bmatrix} \sin\omega t \\ 1-\cos\omega t \\ 1 \end{bmatrix} \\
\dot{\boldsymbol{p}}_d &= \begin{bmatrix} \omega\cos\omega t \\ \omega\sin\omega t \\ 0 \end{bmatrix} \\
\ddot{\boldsymbol{p}}_d &= \begin{bmatrix} -\omega^2\sin\omega t \\ \omega^2\cos\omega t \\ 0 \end{bmatrix} \\
\psi_d &= 0 \quad \vee \quad \psi_d = \omega \cdot t
\end{aligned}
\tag{52}
$$

The circle trajectory was defined using the trigonometric functions sinus and cosinus to define the position along x and y of the drone. Velocities and acceleration were obtained deriving with respect to the time the expression of the position. Again the sign of ω is the user's choice as well as the possibility to change the point around which to rotate. In our case we chose to rotate around the point (0,1) with positive angular velocity.
The yaw angle was first set to zero for the first experiment while set equal to $\omega \cdot t$ for the second simulation. In this way ψ will follow the motion of the drone pointing always toward the center of the circle.

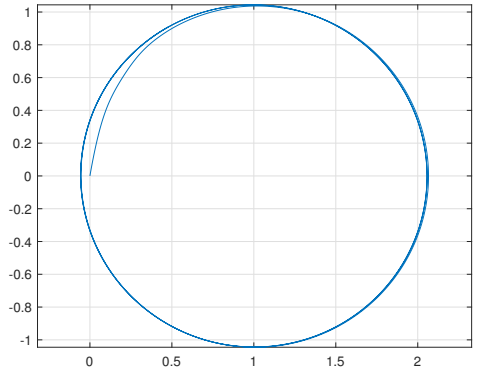
We see in Fig. (2) that both circles are followed during the drone motion. The position error for the first case is around 10 centimeters while in the case of variable yaw angle it oscillates more reaching values very close to 20 centimeters. The yaw angle controller in this way increase the lack of precision of our global trajectory. We see anyway from plot (e) and (f) that the yaw angle motion if fully accomplished.

2.2.3 Eight Trajectory

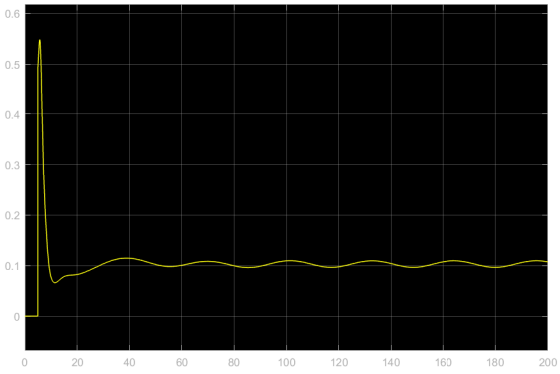
$$
\begin{aligned}
\omega &= 0.15 \\
T &= \frac{2\pi}{\omega} \\
n &= floor(\frac{t}{T}) \\
&\text{if n is even} \\
\boldsymbol{p}_d &= \begin{bmatrix} \sin\omega t \\ 1-\cos\omega t \\ 1 \end{bmatrix} \\
\dot{\boldsymbol{p}}_d &= \begin{bmatrix} +\omega\cos\omega t \\ +\omega\sin\omega t \\ 0 \end{bmatrix} \\
\ddot{\boldsymbol{p}}_d &= \begin{bmatrix} -\omega^2\sin\omega t \\ +\omega^2\cos\omega t \\ 0 \end{bmatrix} \\
\psi_d &= 0 \\
&\text{if n is odd} \\
\boldsymbol{p}_d &= \begin{bmatrix} \sin\omega t \\ \cos\omega t-1 \\ 1 \end{bmatrix} \\
\dot{\boldsymbol{p}}_d &= \begin{bmatrix} +\omega\cos\omega t \\ -\omega\sin\omega t \\ 0 \end{bmatrix} \\
\ddot{\boldsymbol{p}}_d &= \begin{bmatrix} -\omega^2\sin\omega t \\ -\omega^2\cos\omega t \\ 0 \end{bmatrix} \\
\psi_d &= 0
\end{aligned}
\tag{53}
$$

The eight trajectory required a more complicated definition. We need to set a control on the number of the rotation period since the drone will change the point around which to rotate every 360°. For this purpose we define the n parameter as the floor of the t/T ratio, setting different references if n is even or odd. Once set the two position references, velocities and accelerations are defined as in the circle trajectory. Considerations about the signs are in analogy to what already said before. The yaw angle in this case is set to be zero.

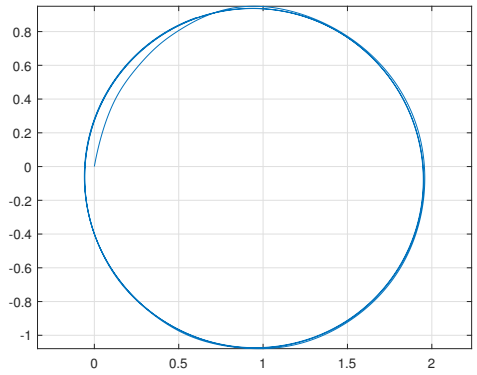
The eight trajectory has an average position error of 15 centimeters. The fact that we can notice directly from the trajectory plot is that the two circles do not meet exactly in (0,0) where the acceleration changes direction. This fact may be related to that even though we have the adaptation law, the limiting factor will be the LQR-controller. The controller will not be able to completely track the desired trajectory during the change of acceleration.



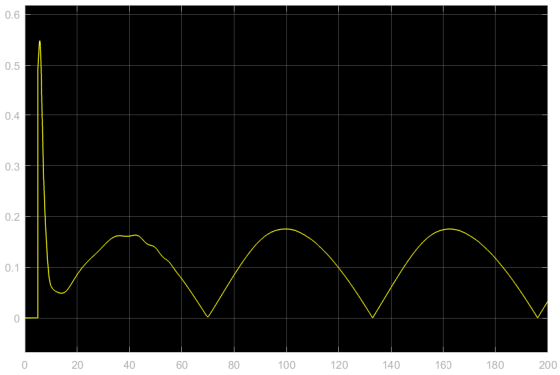
(a) Circle Trajectory



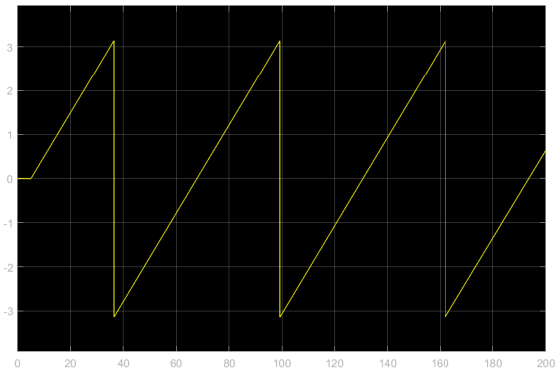
(b) Position Error



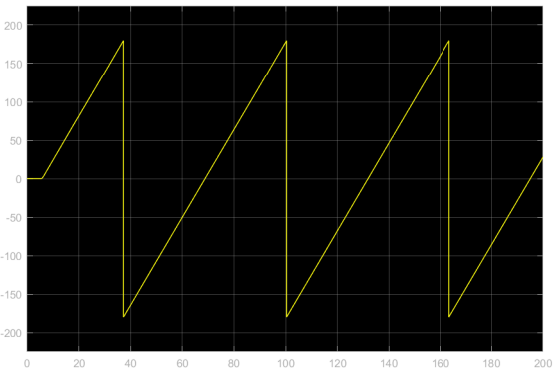
(c) Circle Trajectory with ψ



(d) Position Error with ψ

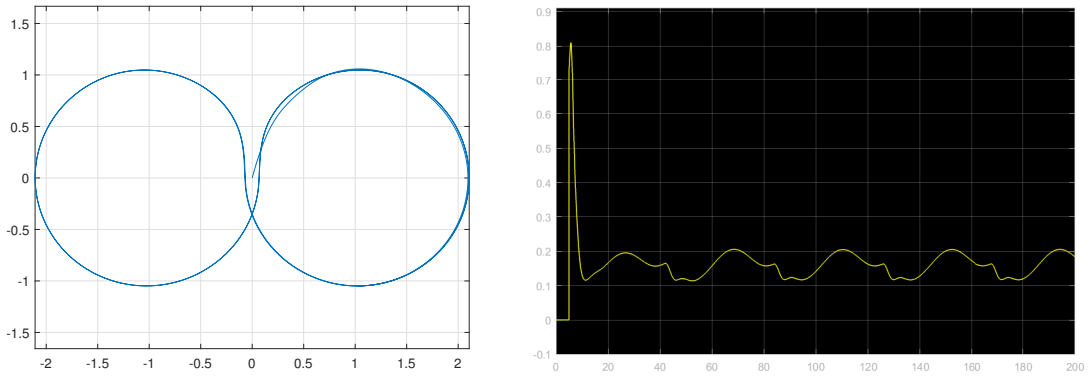


(e) ψ_d (rad)



(f) Heading from States Vector (deg)

Figure 2: Circle Trajectory



(a) Eight Trajectory

(b) Position Error

Figure 3: Eight Trajectory