Unmanned Aerial Vehicles

M.Sc. in Aerospace Engineering

2017/2018 - Second Semester

Motion Control of the Parrot AR.Drone

Laboratory handout

May 2018

1 Introduction

1.1 Objectives

The main goals of this laboratory are the design, implementation, and analysis of several solutions to the problem of quadrotor motion control. Tests will be conducted using the Parrot AR.Drone DevKit. For that purpose, the following items are addressed:

1. Design of a trajectory tracking controller for a generic quadrotor.
2. Incorporation of a adaptation law in the controller to account for unknow disturbances.
3. Implementation, using the AR.Drone DevKit, of a modified version of the designed controller.
4. Implementation, using the AR.Drone DevKit, of a simple trajectory planner for the definition of desired trajectories taking the form of a straight line, a circle, and a figure-eight.

1.2 Organization and timeline

The laboratory component will take place in two sessions. A single report must be delivered within one week after the second session. Use the cover available in the course's webpage as a frontpage. This report is to be dropped in the mailbox of the *Área Científica de Sistemas, Decisão e Controlo, Piso 5 da Torre Norte.*

1.3 Academic ethics code

All members of the academic community of the University of Lisbon (faculty, researchers, staff members, students, and visitors) are required to uphold high ethical standards. Hence, the report submitted by each group of students must be <u>original</u> and correspond to <u>their actual work</u>.

2 Trajectory Tracking Control

2.1. (T) Using the same notation as in previous laboratory handouts, the linear motion of a quadrotor neglecting drag terms can be described by

$$\begin{cases} \dot{\mathbf{p}} = \mathbf{R}(\boldsymbol{\lambda})\,\mathbf{v} \\ m\dot{\mathbf{v}} = -m\,\boldsymbol{\omega} \times \mathbf{v} + mG\mathbf{R}^T(\boldsymbol{\lambda})\,\mathbf{e}_3 - u_w\mathbf{e}_3 \end{cases},$$

where $\mathbf{e}_3 = [0\ 0\ 1]^T$ and

$$\mathbf{R}(\boldsymbol{\lambda}) = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi).$$

Show that linear acceleration in the inertial frame, given by $\ddot{\mathbf{p}}$, can be written as

$$\ddot{\mathbf{p}} = G\mathbf{e}_3 - \frac{u_w}{m}\mathbf{R}(\lambda)\mathbf{e}_3. \tag{1}$$

2.2. (T) Simplify the dynamics to obtain

$$\ddot{\mathbf{p}} = G\mathbf{e}_3 - \frac{1}{m}\mathbf{R}_z(\psi)\mathbf{u}^*. \tag{2}$$

where $\mathbf{u}^*$ denotes a virtual input. Write $\mathbf{u}^*$ as a function of u_w, θ, and ϕ.

From now on, it is assumed that an inner-loop controller drives $(\phi,\ \theta)$ to $(\phi_r,\ \theta_r)$ and that, for design purposes, the time-scale separation between the outer-loop position dynamics and inner-loop orientation dynamics is sufficiently large to neglect the coupling between the two.

2.3. (T) A **trajectory tracking** controller will be designed in the sequel. First, feedback linearization is performed. Consider the system described by (3) and let $\mathbf{p}_d(t) \in \mathbb{R}^3$ be a desired trajectory, with known continuous first and second time derivatives denoted by $\dot{\mathbf{p}}_d(t) \in \mathbb{R}^3$ and $\ddot{\mathbf{p}}_d(t) \in \mathbb{R}^3$, respectively. Define the error variables $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_d$ and $\tilde{\mathbf{v}} = \dot{\mathbf{p}} - \dot{\mathbf{p}}_d$. Show that the error dynamics can be written as

$$\begin{cases} \dot{\tilde{\mathbf{p}}} = \tilde{\mathbf{v}} \\ \dot{\tilde{\mathbf{v}}} = \mathbf{u} \end{cases},$$

where $\mathbf{u}$ is a new control variable. Write the explicit expression of the input transformation from $\mathbf{u}^*$ to $\mathbf{u}$.

2.4. (T) Define the error system with state

$$\tilde{\mathbf{x}} = \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\mathbf{v}} \end{bmatrix}.$$

Write the error dynamics in the form

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

and propose the control law $\mathbf{u}$ that solves the LQR problem

$$J := \int_0^{+\infty} \left(\tilde{\mathbf{x}}^T(t)\mathbf{Q}\tilde{\mathbf{x}}(t) + \rho\,\|\mathbf{u}(t)\|^2\right) dt,$$

where $\rho > 0$ and $\mathbf{Q}$ is a positive definite matrix.

2.5. (T) With the proposed controller, show that there exists a positive definite matrix $\mathbf{P}$ such that for the Lyapunov function

$$V_1 = \tilde{\mathbf{x}}^T\mathbf{P}\tilde{\mathbf{x}},$$

the time derivative is given by

$$\dot{V}_1 = -\tilde{\mathbf{x}}^T\mathbf{Q}^*\tilde{\mathbf{x}},$$

where

$$\mathbf{Q}^* = \mathbf{Q} + \frac{1}{\rho}\mathbf{P}\mathbf{B}\mathbf{B}^T\mathbf{P}.$$

Conclude about the stability of the equilibrium point $\tilde{\mathbf{x}} = \mathbf{0}$.

2.6. **(T)** Show that, with the LQR controller, the input $\mathbf{u}^*$ in (3) can be written as

$$\mathbf{u}^* = m\mathbf{R}_z^T(\psi)\left(G\mathbf{e}_3 - \ddot{\mathbf{p}}_d + \frac{1}{\rho}\mathbf{B}^T\mathbf{P}\tilde{\mathbf{x}}\right).$$

2.7. **(T)** Suppose now that the quadrotor linear dynamics is described by

$$\ddot{\mathbf{p}} = G\mathbf{e}_3 - \frac{u_w}{m}\mathbf{R}(\lambda)\mathbf{e}_3 + \mathbf{R}(\lambda)\mathbf{d}, \tag{3}$$

where $\mathbf{d}$ is an unknown disturbance and that an estimate $\hat{\mathbf{d}}$ is used in the control law $\mathbf{u}^*$ such that

$$\mathbf{u}^* = m\mathbf{R}_z^T(\psi)\left(G\mathbf{e}_3 - \ddot{\mathbf{p}}_d + \frac{1}{\rho}\mathbf{B}^T\mathbf{P}\tilde{\mathbf{x}} + \mathbf{R}(\lambda)\hat{\mathbf{d}}\right).$$

Let $\tilde{\mathbf{d}} = \mathbf{d} - \hat{\mathbf{d}}$. Show that the new error dynamics is given by

$$\dot{\tilde{\mathbf{x}}} = \left(\mathbf{A} - \frac{1}{\rho}\mathbf{B}\mathbf{B}^T\mathbf{P}\right)\tilde{\mathbf{x}} + \mathbf{B}\mathbf{R}(\lambda)\tilde{\mathbf{d}}.$$

2.8. **(T)** Assuming that $\mathbf{d}$ is constant, an adaptive controller is now designed. For that purpose, consider the Lyapunov-like function

$$V_2 = V_1 + \frac{1}{K_d}\tilde{\mathbf{d}}^2,$$

where $K_d > 0$ is a tuning gain. Propose an adaptation law for $\dot{\hat{\mathbf{d}}}$ such that $\dot{V}_2$ results negative semi-definite.

2.9. **(T)** Assuming that $\ddot{\mathbf{p}}_d$ is constant and different from $G\mathbf{e}_3$, show that the origin of the system with state $\left(\tilde{\mathbf{x}}, \tilde{\mathbf{d}}\right)$ is globally asymptotically stable.

2.10. **(T)** The proposed trajectory tracking controller has been designed independently from the yaw angle dynamics, highlighting the fact that, for quadrotor vehicles, the yaw angle is not constrained by the linear dynamics. Assume that the yaw motion is simply described by a first order system, not necessarily a pure integrator. Design a linear feedback controller that guarantees tracking of a constant reference with zero steady-state error.

3 Implementation and simulation using the AR.Drone DevKit

An updated version of the DevKit is provided for this laboratory. The following changes were introduced:

- The starting script is now called `start_here_CTRL.m`. Under the option `(1) Simulation`, a new suboption called `(4) Trajectory Tracking` is available, which opens the Simulink Model `ARDroneTTSim.slx`.
- The model contains two blocks `desired_trajectory` and `tt_controller`. Notice that the latter is located inside the subsystem `Outer-loop Controller`. The inputs and outputs of these blocks are given but the content is partially empty. To implement and simulate the trajectory tracking controller, both blocks need to be filled in the form of Matlab Functions.

3.1. **(L)** Recalling that $\mathbf{u}^*$ defines a virtual input, write the expressions for the thrust input u_w and the references ϕ_r and θ_r to be used as inputs for the inner-loop controllers. In contrast to the model and inner-loop control structure described in Section 2, the model provided in the AR.Drone DevKit has as inputs the roll and pitch angle references ϕ_r and θ_r, respectively, the vertical velocity reference w_r, and the yaw rate reference $\dot{\psi}_r$.

To implement the controller and adapt it to the available inputs, keep the expressions for the roll angle, pitch angle, and yaw rate references. To simplify the design for the vertical channel, all the trajectories are assumed to belong to the horizontal plane, the roll and pitch angles are very small, and a simple proportional controller is implemented, as given by

$$w_r = -k_w \left(z - z_d\right).$$

3.2. **(L)** Focusing now on the generation of the desired trajectory, fill the block `desired_trajectory` such that $\mathbf{p}_d(t)$ defines a straight line trajectory with a suitable constant height, to be described with a speed of 0.1 m/s and a constant yaw angle. The initial desired position should match the initial position of the quadrotor when the tracking is started. Do not forget that the first and second time derivates of $\mathbf{p}_d(t)$ are also required.

3.3. **(L)** Test the implemented controller and check whether or not tracking is accomplished. Tune the gains if necessary and comment on the achieved performance.

3.4. **(L)** Consider now the tracking of a circular trajectory on the horizontal plane. Make the necessary changes in `desired_trajectory` to generate a circular trajectory with a radius of 1 m, to be described with an angular rate of 0.1 rad/s and a constant yaw angle. Again, the initial desired position should match the initial position of the quadrotor when the tracking is started.

3.5. **(L)** Test the implemented controller with the new desired trajectory and check whether or not tracking is accomplished. Comment on the achieved performance.

3.6. **(L)** Continuing with the circular trajectory, change the desired yaw angle so that the quadrotor always points towards the center of the circunference. Simulate the system with this new trajectory and compare the obtained results with those of the previous questions. Provide an explanation for the observed differences.

3.7. **(L)** Consider now the tracking of a figure-eight trajectory. The desired trajectory should be limited by a rectangle of 1 m by 2 m and the maximum desired linear speed should not exceed 0.15 m/s. Define $\mathbf{p}_d(t)$ and its derivatives accordingly. Test the trajectory tracking controller and comment on the obtained results.