Control of Autonomous Systems – Autumn 2025
Jerome Jouffroy, PhD
Exercise Session 4

Figure 1: Longitudinal dynamics of a helicopter

1. **Stabilization of a helicopter**. Consider the longitudinal dynamic model of a helicopter, given by the linear state-space representation

$$\frac{d}{dt}\begin{bmatrix} q \\ \theta \\ u \end{bmatrix} = \begin{bmatrix} -0.4 & 0 & -0.01 \\ 1 & 0 & 0 \\ -1.4 & 9.8 & -0.02 \end{bmatrix}\begin{bmatrix} q \\ \theta \\ u \end{bmatrix} + \begin{bmatrix} 6.3 \\ 0 \\ 9.8 \end{bmatrix}\delta \tag{1}$$

where $u(t)$ is the longitudinal velocity (in m/s) of the helicopter expressed in the earth-fixed frame, $\theta(t)$ is the pitch angle (in radians) and $q(t)$ the pitch angular velocity (in rad/s), while $\delta(t)$ is the control input representing the angle (in rad.) of the rotor thrust with respect to the helicopter (see Figure 1).

1.1. Open a new Simulink file, and create a subsystem where your plant will be. Set the sampling time for the solver of your simulation to $T_s = 0.01s$ so that you can have smooth results on your scopes.

1.2. In a Matlab file/script, define the matrices corresponding to the linear state-space representation (1). Check whether this system is stable by checking the eigenvalues of matrix $\mathbf{A}$.

1.3. In the plant/helicopter subsystem, use a "state-space" block to implement your system (1). Since we would like to have the state $\mathbf{x}(t)$ at the output of this block (i.e. we want $\mathbf{y} = \mathbf{x}$), what should be in the "C" and "D" fields of the block?

1.4. Implement a state-feedback controller that will stabilize the helicopter around the origin. Use the command `place` to tune your controller so that the closed-loop has the desired eigenvalues/poles, ie chosen by you. Check that your controller works for different initial conditions.

1.5. Change your controller into a Linear Quadratic Regulator and tune it using the command `lqr`.

1.6. Add a feedforward gain $\bar{\mathbf{N}}$ so that the helicopter stabilizes around the desired velocity of $\mathbf{r} = 10$ m/s.

2. **A digital controller for the helicopter**. We would like to re-implement the previous controller digitally, ie using a discrete-time framework, while keeping the same continuous-time plant/model for the helicopter as in the previous exercise.

2.1. In a Matlab script, discretize the plant model (1) using the `c2d` command (as well as `ss` and `ssdata`) with sampling period $10T_s$ in order to obtain the matrices $\mathbf{A}_D$ and $\mathbf{B}_D$ of a discrete-time state-space representation from the continuous-time one you obtained in Exercise 1.

2.2. Compute the discrete-time controller gain $\mathbf{K}_D$ using the `dlqr` command.

2.3. Implement your discrete-time controller in Simulink to stabilize your system around the origin (start from a non-zero initial condition). In your implementation, include a 'zero-order hold' block to represent the effect of discretizing/sampling the continuous-time signals.

2.4. We now want the helicopter to stabilize around the desired velocity of $\mathbf{r} = 10$ m/s. Modify the calculations necessary to obtain the feedforward gain so that it will be defined for a discrete-time system with matrices $\mathbf{A}_D$, $\mathbf{B}_D$, $\mathbf{C}_D$ and $\mathbf{D}_D$. Check that it works in your implementation.