Control of Autonomous Systems – Autumn 2025
Jerome Jouffroy, Professor
Exercise Session 10

0. Open-loop control of the mass-spring-damper system. Using your simulation of an MSD system with animation (which we have used a couple of times by now) and your lecture notes on open-loop control, design and implement an open-loop controller for the MSD system to bring it from position 0 to position 30 in 10 seconds. For this, you can use the Matlab program `poly3traj` (available on Itslearning), which computes the coefficients of a polynomial trajectory given the time horizon, as well as the initial and final positions and velocities. What happens after 10 seconds? What can you do to make the mass stay at position 30 after 10 seconds?

1. Open-loop control of the altitude of a hot air balloon. Consider the simplified model of a hot air balloon

$$\begin{cases} \dot{\theta} = -\theta + u \\ \dot{h} = v \\ \dot{v} = -\dfrac{v}{2} + \theta \end{cases} , \qquad (1)$$

where $h(t)$ and $v(t)$ are the altitude and the vertical velocity of the balloon, respectively, while $\theta(t)$ is the temperature difference allowing the balloon to change altitude. Control signal $u(t)$ represents the influence of the burner.

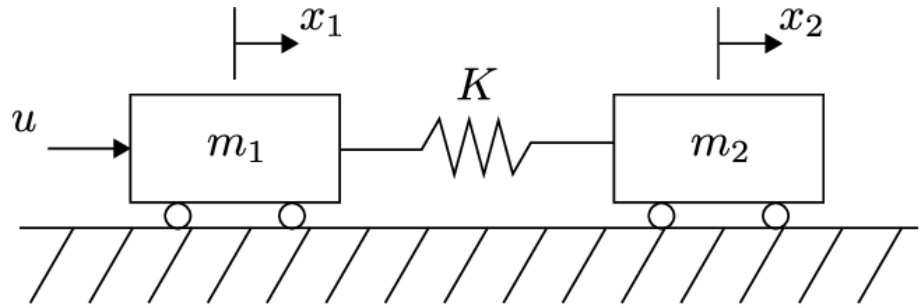

Figure 1: The two-mass-spring system.

1.1. In the Simulink file "balloon_sim_temp.slx", implement (1) in the plant subsystem. Then, design and implement a state-feedback controller. Simulate the whole system so that the balloon reaches an altitude of 100m starting from the ground at altitude 0.

1.2. After obtaining a state-space representation of system (1), compute the transform $\mathbf{T}$ that will change the state-space representation you just obtained into a Controllability Canonical Form. What does the new variable $z_1 = \mathbf{T}_1\mathbf{x}$ represent?

1.3. Use the Matlab script `poly5traj.m` available on Itslearning to compute the coefficients of a desired trajectory that will bring the balloon's altitude from the ground to 100m in 10 minutes (ie 600 seconds).

1.4. Use the desired trajectory of question 1.3. (and its derivatives) to create an open-loop controller that will bring your balloon to 100m in 10 minutes. Compare the result with the feedback controller designed in question 1.1.

2. **Motion planning of a two-mass-spring system**. Consider the underactuated system (ie it has one control input for 2 degrees-of-freedom) of two masses coupled by a spring, as represented in Figure 1, and modelled by

$$\begin{cases} m_1\ddot{x}_1 + k(x_1 - x_2) = u \\ m_2\ddot{x}_2 + k(x_2 - x_1) = 0 \end{cases}, \tag{2}$$

with $u(t)$ being the force applied to the first mass, $x_1(t)$ and $x_2(t)$ are the positions of the masses, and where $m_1 = 10$, $m_2 = 20$ and $k = 5$.

2.1. In the Simulink file "dmass_sim_temp.slx", implement (2) in the plant subsystem. Simulate this system for $u = 0$ and different initial conditions, and see if the results fit with the physics of the system.

2.2. We would like to bring both mass positions[1] from 0 to 40 in 10 seconds, with starting and ending velocities at 0. What should be the minimum order of the polynomial used to generate the desired trajectory?

2.3. Make/use a program to generate the desired trajectory mentioned in question 2.2.

2.4. Give a state-space representation of system (2). Then, compute the transform $\mathbf{T}$ that will change the state-space representation you obtained into a Controllability Canonical Form. What does the new variable $z_1 = \mathbf{T}_1\mathbf{x}$ represent?

2.5 Design and implement an open-loop controller per the requirements of question 2.2. Check the behavior of both positions using a scope. Which variable should match the behavior of the desired trajectory?

[1] Each mass coordinate have their own origin, ie the position 0 is meant from their position at rest, separated by the spring, see Figure 1.