Learning for Adaptive and Reactive Robot Control
Instructions for exercises of lecture 9

Professor: Aude Billard
Contact: aude.billard@epfl.ch

Spring Semester

Introduction

The exercises of Lecture 9 are composed of theoretical (pen and paper) exercises and matlab exercises. The exercises revisit fundamental concepts of impedance control with dynamical systems.

They follow *exercises* 10.1, 10.2, 10.7 and *programming exercises* 10.1 *to* 10.3 of the book "*Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach. MIT Press, 2022*".

1 Theoretical exercises [1h]

1.1

Book correspondence: Ex10.1, p272

Consider a second-order, one-dimensional dynamical system (DS) of the form:

$$m\ddot{x} + k(x - x^*) = 0, \qquad x(0) = x_0 \tag{1}$$

We now want to use this DS to hit a target at $x = x^*$, with desired velocity $\dot{x} = \dot{x}^*$ at $t = t^*$

- Can this be achieved?
- If no, what constraint would need to be softened?
- How does is the behavior of the system after reaching the attractor x^*?

Hint: The solution of the second-order differential equation of a mass damper system is a known equation.

1.2

Let us now consider a dynamical system with relative damping to the attractor velocity $\dot{x}^*$:

$$m\ddot{x} + d(\dot{x} - \dot{x}^*) + k(x - x^*) = 0, \qquad x(0) = x_0$$

- What effect does the damping have?
- Can the DS pass through two points in space with two different velocities, i.e.,

$$\begin{aligned} t = t_1^* \quad &\text{with} \quad x(t_1^*) = x_1^* \quad \dot{x}(t_1^*) = \dot{x}_1^* \\ t = t_2^* \quad &\text{with} \quad x(t_2^*) = x_2^* \quad \dot{x}(t_2^*) = \dot{x}_2^* \end{aligned}$$

1.3

Book correspondence: Ex10.2, p273

Consider the DS, designed in Equation 1.2, to hit a nail using a point mass robotic system. If $m = 2$, $k = 25$, and the impact is elastic with an impact time $T = 0.01$.
Assume that you start the motion at height $x(0) = 1.3m$ and that you hit the the nail at the maximum velocity during your cycle.

- What is the force applied to the nail at each cycle?
- What is the cycle time t_c? How does cycle time get affected by the elastic impact?

Hint: The impact force for the robot can be calculated by $F = \frac{2mv}{T}$

1.4

Book correspondence: Ex10.7, p284

Consider the task of reaching and contacting a large fixed object presented in section 8.4, with motion dynamics for the robot as defined by:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + J(q)^T F_c$$

and driven by low level controls (task dependent):

$$\mathrm{F} = \mathrm{V}_x(q,\dot{q})\dot{x} + G_x(q) + M_x(q)\ddot{x} - (\mathbf{D}(\tilde{\mathbf{x}})\dot{\tilde{x}} + \mathbf{K}(\tilde{x})\tilde{x})$$

Design the impedance profile such that the robot is stiff if it is far from the surface and compliant if it is close to the surface.

- What would happen if one designed the stiffness profile and kept the damping matrix constant?
- What role has the inertial matrix?

2 Programming exercises [1h]

2.1 Programming Exercise : Impedance controller

Book correspondence: Programming Ex10.1, p274

The aim of this exercise is to help readers to get a better understanding of the impedance control architecture [equations (10.6), (10.7), and (10.8) in the book] and the effect of the open parameters on the generated motion. Open MATLAB and set the directory to the following folder:

```
lecture9-compliant-control/exercises
```

Open file `lect9_ex1.m`. This script will compute and plot the joint positions, velocities and torques of a 2-DOF robotic arm with three different impedance control approaches. You can modify the parameters of these controllers and the initial configuration of the robot to study their effects when tracking a sinusoidal target.

1. Run section 1 to observe the first simple position controller, without inertia compensation:

   $$\boldsymbol{\tau} = \mathbf{G}(\mathbf{q}) - (\mathbf{D}\dot{\mathbf{q}} + \mathbf{K}\tilde{\mathbf{q}}) \tag{2}$$

   The joint torque $\boldsymbol{\tau}$ is proportional to the joint position error $\tilde{\mathbf{q}}$ with the stiffness matrix $\mathbf{K}$, the damping matrix $\mathbf{D}$ reduces joint velocities, and the feed-forward term $\mathbf{G}(\mathbf{q})$ compensates for gravity effects.

   (a) Using the provided stiffness and damping, change the initial position, speed and acceleration to see how it affects the trajectory. Does the robot converges to the reference ?

   (b) Change the `Stiffness` variable by one order of magnitude in both directions. How does this affect the tracking performance ? What is the main limitation ?

   (c) The `Damping` variable is tuned to critically dampen the system. What happens when your system is overdamped or underdamped?

2. Run section 2 to observe a controller with inertia and coriolis compensation:

   $$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}) - (\mathbf{D}\dot{\tilde{\mathbf{q}}} + \mathbf{K}\tilde{\mathbf{q}}) \tag{3}$$

   Two feed-forward terms are added to compute the torque needed to reach a desired joint speed $\dot{\mathbf{q}}_d$ and acceleration $\ddot{\mathbf{q}}_d$, which gives better tracking performances when the target is moving.

   (a) Without changing the stiffness and damping, compare the tracking performance of this controller compare to the first one. What do you observe ?

   (b) Try decreasing the stiffness of the controller to get similar performance as with the first controller. What is the effect on the joint torque ?

3. Run section 3 to observe the controller with inertia reshaping (full impedance controller):

   $$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}_d + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_d + \mathbf{G}(\mathbf{q}) - \mathbf{M}(\mathbf{q})\boldsymbol{\Lambda}^{-1}(\mathbf{D}\dot{\tilde{\mathbf{q}}} + \mathbf{K}\tilde{\mathbf{q}}) + (\mathbf{M}(\mathbf{q})\boldsymbol{\Lambda}^{-1} - \mathbf{I}) * \mathbf{J}(\mathbf{q})^T\mathbf{F}_c \tag{4}$$

The added terms allows to change the apparent inertia of the robot, and have the overall 2nd order system:

$$\mathbf{\Lambda}\ddot{\tilde{\mathbf{q}}} + \mathbf{D}\dot{\tilde{\mathbf{q}}} + \mathbf{K}\tilde{\mathbf{q}} = \mathbf{J}(\mathbf{q})^T\mathbf{F}_c \tag{5}$$

which is akin to a virtual mass-spring-damper system with external cartesian force $\mathbf{F}_c$, and virtual mass $\mathbf{\Lambda}$, damping $\mathbf{D}$ and stiffness $\mathbf{K}$.

(a) What is the tracking performance of this controller with the default gains compared to the second one ?

(b) Change the inertia parameter by one order of magnitude in both direction. What is the effect on perturbations rejection and on the controlled joint torques ?

2.2 Programming Exercise : Variable impedance controller

Book correspondence: Programming Ex10.2, p284

The aim of this exercise is to help readers to get a better understanding of the three variable impedance control approaches and the effect of the open parameters on the generated motion.

Open file `lect9_ex2.m`. Similarly to the previous exercise, this script will compute and plot the robot configuration of a 2-DOF robotic arm for three examples of variable impedance controllers. It will also output the variation in impedance for each controller. For easier reading, each approach is separated into sections in the Matlab code.

Each controller is based on the full impedance formulation of equation 4, but has a different way of varying its stiffness and damping matrices. Each approach is detailed below:

1. Run Section 1 to observe the first approach where the gains vary between two configuration $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$:

$$\mathbf{K}(\mathbf{q}) = \frac{\lambda_1}{\lambda_1+\lambda_2}\mathbf{K}_1 + \frac{\lambda_2}{\lambda_1+\lambda_2}\mathbf{K}_2 \qquad \text{and} \quad \mathbf{D}(\mathbf{q}) = \frac{\lambda_1}{\lambda_1+\lambda_2}\mathbf{D}_1 + \frac{\lambda_2}{\lambda_1+\lambda_2}\mathbf{D}_2 \tag{6}$$

$$\lambda_{1,2} = ||(\mathbf{q} - \boldsymbol{\mu}_{1,2})|| \tag{7}$$

The actual stiffness $\mathbf{K}(\mathbf{q})$ and damping $\mathbf{D}(\mathbf{q})$ will vary smoothly between two sets of gain ($\mathbf{K1}$, $\mathbf{D1}$) and ($\mathbf{K2}$, $\mathbf{D2}$), depending on the distance to the reference joint positions $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. This allows different tuning of the controller for different parts of the workspace.

(a) Change the sets of gains ($\mathbf{K}$, $\mathbf{D}$) for approach 1 using the variables `Damping` and `Stiffness`. Observe how the perturbations are corrected when subjected to different sets of gains.

(b) Change the scheduling position parameters $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. What is their effect on the impedance variation?

2. Run Section 2 to observe the second approach where the gains vary between two states, with time-dependence:

$$\mathbf{K}(\mathbf{q},t) = \frac{\lambda_1}{\lambda_1+\lambda_2}\mathbf{K}_1 + \frac{\lambda_2}{\lambda_1+\lambda_2}\mathbf{K}_2 \quad \text{and} \quad \mathbf{D}(\mathbf{q},t) = \frac{\lambda_1}{\lambda_1+\lambda_2}\mathbf{D}_1 + \frac{\lambda_2}{\lambda_1+\lambda_2}\mathbf{D}_2 \tag{8}$$

$$\lambda_1 = ||(\mathbf{q}-\boldsymbol{\mu}_1)|| + ||(\dot{\mathbf{q}}-\dot{\boldsymbol{\mu}}_1)|| * t \quad \text{and} \quad \lambda_2 = ||(\mathbf{q}-\boldsymbol{\mu}_2)|| + ||(\dot{\mathbf{q}}-\dot{\boldsymbol{\mu}}_2)|| \tag{9}$$

Similar to the first approach, but now with a full state reference $\left(\begin{smallmatrix}\boldsymbol{\mu}\\ \dot{\boldsymbol{\mu}}\end{smallmatrix}\right)$ instead of only a position reference. Note the addition of the time t in λ_1 that will make the system converge to the set of gains (**K1**, **D1**).

(a) Change the sets of gains ($\mathbf{K}$, $\mathbf{D}$) for approach 2 using the variables `Damping` and `Stiffness`. Observe how the perturbations are corrected when subjected to different sets of gains.

(b) The perturbations have a noticeable effect on the impedance profile. Why? compare with the first approach.

3. Run Section 3 to observe the third approach with basic time-dependence:

$$\dot{\mathbf{K}} = -c * (\mathbf{K} - \mathbf{K}_d) \quad \text{and} \quad \dot{\mathbf{D}} = -c * (\mathbf{D} - \mathbf{D}_d) \tag{10}$$

Here the gains follow a first order system starting at an initial value ($\mathbf{K}_0$, $\mathbf{D}_0$) and converging to ($\mathbf{K}_d$, $\mathbf{D}_d$) at a rate defined by the constant $c > 0$

(a) Change the initial and desired gains ($\mathbf{K}$, $\mathbf{D}$) for approach 3 using the variables `Damping` and `Stiffness`. Observe how the perturbations are corrected when subjected to different sets of gains.

4. (Optional) You can also modify the dynamic specifications, the initial configuration and the external perturbation force to study their effects on the generated trajectories.

References

[1] Aude Billard, Sina Mirrazavi, and Nadia Figueroa. *Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach.* MIT press, 2022.