Learning for Adaptive and Reactive Robot Control
Instructions for exercises of lecture 4

Professor: Aude Billard
Assistants: Harshit Khurana,
Lukas Huber and Yang Liu
Contacts:
aude.billard@epfl.ch, harshit.khurana@epfl.ch,
lukas.huber@epfl.ch, yang.liuu@epfl.ch

Spring Semester 2022

Introduction

INTRO
This part of the course follows *exercises* 3.2*,* 3.3 *and* 3.5 and *programming exercises* 3.1 *to* 3.4 of the book "*Learning for Adaptive and Reactive Robot Control: A Dynamical Systems Approach. MIT Press, 2022*".

1 Theoretical exercises [1h]

1.1

Design a matrix $A \in \mathbb{R}^{2\times 2}$ and Lyapunov function shaping matrix $P \in \mathbb{R}^{2\times 2}$ to ensure that a linear dynamical system (DS),

$$\dot{x} = f(x) = A(x - x^*)$$

with attractor at the origin $x^* = [0\ 0]^T$ to be globally, asymptotically stable (GAS) with respect to the conditions stated using either:

(a) A matrix $Q \in \mathbb{R}^{2\times 2}$ with the following form:

$$Q = q\,\mathbb{I}_2, \quad q \in \mathbb{R}$$

(b) *Optional* A matrix $Q \in \mathbb{R}^{2\times 2}$ with the following form:

$$Q = \begin{bmatrix} q_1 & q_2 \\ q_2 & q_1 \end{bmatrix}, \quad q_1, q_2 \in \mathbb{R}$$

Hint: What conditions should the entries of the Q matrices satisfy?

1.2

Consider a DS of the form

$$\dot{x} = \sum_{k=1}^{K} \gamma_k(x)(A^k x + b^k)$$

with $N = 2$, $K = 2$, and $\gamma_1(x) = 0.7$ and $\gamma_2(x) = 0.3$. Design nondiagonal matrices $A^1 \in \mathbb{R}^{2\times 2}$ and $A^2 \in \mathbb{R}^{2\times 2}$ such that the DS is GAS at the origin - that is, $x^* = [0\ 0]^T$, with a quadratic Lyapunov function (QLF).

Hint: The sum of two negative definite matrices leads to a negative definite matrix.

1.3

Optional: To be done at home.

Consider the nonlinear DS

$$\dot{x} = \mathbb{E}\{p(\dot{x}|x)\} = \sum_{k=1}^{K} \gamma_k(x)(A^k x + b^k)$$

with $N = 1$, $K = 4$. Design the Gaussian mixture regression (GMR) parameters $\Theta_{GMR} = \{\pi_k, \mu^k, \Sigma^k\}_{k=1}^{2}$ that will produce a GAS DS at the target $x^* = 0$ such that:

- If $x = -1$, then $\dot{x} = 0.5$; and when $x = -2$, then $\dot{x} = 2$.
- If $x = +1$, then $\dot{x} = -0.5$; and when $x = +2$, then $\dot{x} = -2$.

2 Programming exercises [1h]

Preliminaries: Instructions for proper installation of MATLAB libraries

Download the folder `lecture4-learning-control-laws` and place it next to the existing folders. Your folder `matlab_exercises` should now have the following structure:

```
🟥 matlab_exercises
 ├─ 🟦 lecture1-introduction
 │   ├─ 🟩 exercises
 │   └─ 🟩 solutions
 ├─ 🟦 lecture2-learning-from-demonstration
 │   ├─ 🟩 exercises
 │   └─ 🟩 solutions
 ├─ 🟦 lecture3-intro-to-dynamical-systems
 │   ├─ 🟩 exercises
 │   └─ 🟩 solutions
 ├─ 🟦 lecture4-learning-control-laws
 │   ├─ 🟩 exercises
 │   └─ setup_code_ch3.m
 └─ 🟦 libraries
```

!!! IMPORTANT !!! Note that there is a file called `setup_code_ch3.m` in the downloaded folder. This script needs to be run before starting the exercises, in order to install required libraries. For this script to be executed successfully, a MEX compiler needs to be setup on your machine. If you are not sure if that's the case, enter `mex -setup` in the MATLAB command line. On Windows, it should then say something similar to

```
MEX configured to use 'Microsoft Visual C++ 2019 (C)' for C language
                          compilation.
```

while on Linux, it should be

```
MEX configured to use 'gcc' for C language compilation.
```

and on MacOS it should be

```
MEX configured to use 'Xcode with Clang' for C language compilation.
```

2.1 Programming Exercise 3.1: Unstable DS regressors

Book correspondence: page 50.
The aim of this instructional exercise is to allow readers to observe the type of instabilities that are generated on the DS when using standard regression algorithms and compare this to using SEDS.

Instructions: Open MATLAB and set the directory to the folder corresponding to lecture 4 exercises. In this folder, you will find, among others, the following MATLAB scripts:

```
ch3_ex1_gmrDS.m
ch3_ex1_svrDS.m
ch3_ex1_gprDS.m
```

```
ch3_ex1_ffnnDS.m
ch3_ex3_seDS.m
ch3_ex4_LPVDS.m
```

These scripts allow to learn a DS using either of the following five techniques: Gaussian Mixture Regression (GMR), Support Vector Regression (SVR), Gaussian Process Regression (GPR), Feed Forward Neural Networks (ffnn) and Stable Estimator of DS (SEDS) and the Linear Parameter Varying DS (LPV-DS). When running each script, you will be asked to select among a set of predefined trajectories. The goal of this exercise is to assess systematically in both a qualitative and quantitative manner the performance of these five techniques. You will do this by comparing the fit on three trajectories of increasing complexity.

TASK 1 Run each script and select in turn the "C-shape (option number 3)", "J-shape (option number 7)" and "S-shape (option number 22)".
Observe the rendered plots and the mean-square error reported on the prompt. For each shape, fill in the corresponding tables:

C-shape						
Criterion	GMR	SVR	GPR	FFNN	SEDS	LPV-DS
Stability at attractor [Y/N]						
Spurious attractor [Y/N]						
Generalization of dynamic [Y/N]						
RMSE of velocity [m/s]						
Mean velocity deviation						
Computation time [s]						

J-shape						
Criterion	GMR	SVR	GPR	FFNN	SEDS	LPV-DS
Stability at attractor [Y/N]						
Spurious attractor [Y/N]						
Generalization of dynamic [Y/N]						
RMSE of velocity [m/s]						
Mean velocity deviation						
Computation time [s]						

S-shape						
Criterion	GMR	SVR	GPR	FFNN	SEDS	LPV-DS
Stability at attractor [Y/N]						
Spurious attractor [Y/N]						
Generalization of dynamic [Y/N]						
RMSE of velocity [m/s]						
Mean velocity deviation						
Computation time [s]						

TASK 2 Try to find an explanation for why some of the techniques perform well or not on some parts of the trajectories (those with high-low curvature, high-low density of points), knowing the sensitivity of each techniques to these parameters.

TASK 3 Modify the code of SEDS (at Step 3) and compare SEDS [Likelihood] (default) to SEDS [MSE] in terms of reconstruction error (RMSE), model complexity and computation time for the "C-shape (option number 3)", "J-shape (option number 7)" and "S-shape (option number 22)".

TASK 4 Modify the code of SEDS to handdraw trajectories. Draw first a set of trajectories easy for SEDS to learn and second one set difficult to learn. Verify that your assumptions are correct with the results.

TASK 5 Draw self-intersecting trajectories. Is SEDS suitable for these? Where should the self- intersecting trajectories be disconnected/segmented in order to properly use multiple DS learned via SEDS?

TASK 6 Using the LPV-DS code, compare results with different GMM fitting approaches:

1. EM with Model Selection via BIC
2. Collapsed sampler for CRP-GMM
3. Collapsed sampler fo PC-GMM

2.2 Programming Exercise 3.2: Nonlinear DSs as a mixture of linear systems (OPTIONAL)

Book correspondence: page 57.
This exercise will allow the reader to understand the complexity in designing globally asymptotically stable DS as equation (3.21) in the book by designing and visualizing DS as those defined in example 3.1.

Instructions: Open MATLAB and set the directory to the folder corresponding to lecture 4 exercises. By following the instructions in the following MATLAB script:

```
ch3_ex2_designDS.m
```

the reader will be able to reproduce the illustrations from figure 3.5 (in the book). Further, we recommend the following:

- Design values of γ_1, A^1, γ_2, A^2 such that a globally asymptotically stable system is generated. One can use this code as support to solve theoretical exercise 1.2 above (exercise 3.3 in the book).
- Add a third linear DS, $K = 3$, can you find the full set of parameters to ensure GAS? How about simply stability? (See appendix A.)

Readers can also us an instructional GUI, such as the one shown in figure 3.25 (in the book), to complement the previous exercise. With this GUI, only the self-drawn trajectories can be used to learn DS models via SE-LPVDS variants. However, it provides a robot simulation that allows the user to execute the motion of the learned DS and apply perturbations to the robot while it is executing the task.

Instructions: Open MATLAB and set the directory to the folder corresponding to lecture 4 exercises. In this folder, you will find, among others, the following MATLAB scripts:

```
gui_lpvDS.m
```

With this GUI, the user will be able to do the following:

- Draw trajectories within the workspace of a 2-DOF robot arm.
- Once trajectories are drawn, click `store data`
- Manually select K, or use model selection via BIC or an automatic PC-GMM fit.
- Learn a LPV-DS with optimization variant (O1) or (O2).
- Simulate the robot motion following the learned DS.
- Apply perturbations to the robot while it is executing the motion.

Note: An instructional movie on how to use this GUI can be found at the following YouTube video: `https://youtu.be/jDcPaOUMwvI`.