Technische Universiteit
Eindhoven
University of Technology

Design Assignment

Policy Learning for an Unbalanced Disc

5SC28 Machine Learning for Systems and Control

Authors:
Maarten Schoukens, Roland Tóth and Gerben Beintema

May 10, 2021

Contents

1 Introduction

The unbalanced disc as depicted in Figure 1 acts as a pendulum. The primary objective of this design assignment is to obtain a controller for the system that can swing up the pendulum and keep it stable at the top position. From a mechanical point of view, the system behaves as an inverted pendulum. Systems that exhibit dynamics similar to an inverted pendulum are quite a common sight in the streets nowadays: segways, hover boards, and (electric) unicycles are all examples of inverted pendulum systems.

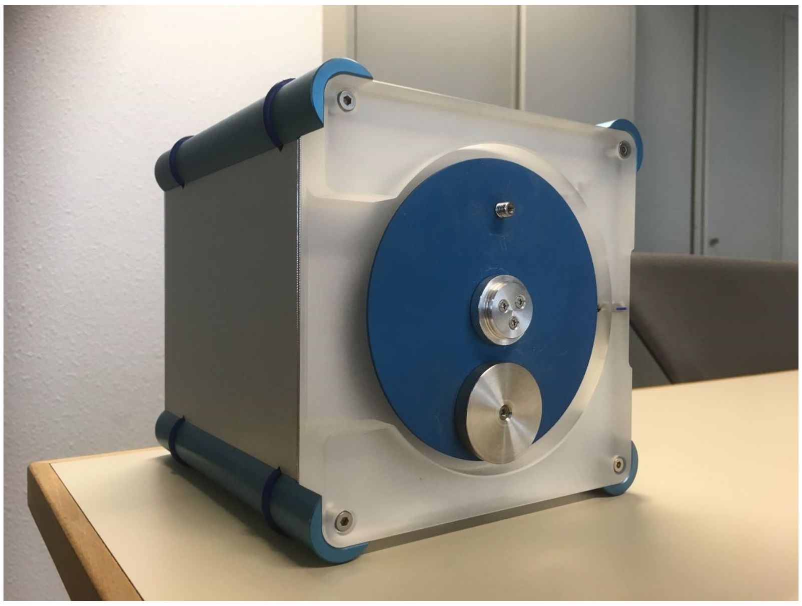

Figure 1: The unbalanced disc setup.

In recent years, several machine learning methods have been introduced based on, for example, Gaussian processes and neural networks. These approaches can be seen as general function approximators which can be trained to capture unknown relationships directly form data. This training process can be seen similar to learning, in which the corresponding AI algorithm gradually builds up a model of the relationship by observing incoming data or results of experiments (interactions with a given system). These methods can also be used to learn a map from response of the system to actuation inputs in order achieve a desired task, like swinging up and stabilizing a pendulum connected to a DC motor. While most control design methods rely on the knowledge of the system dynamics, i.e. they are model based control methods, a learning based approach can obtain a controller of a system in an automated way without any user interaction.

2 First Principles Model

The ideal motion dynamics of the inverted pendulum can be modeled as:

$$\begin{aligned} \dot{\theta}(t) &= \omega(t), \\ \dot{\omega}(t) &= \frac{Mgl}{J}\sin(\theta(t)) - \frac{1}{\tau}\omega(t) + \frac{K_m}{\tau}u(t), \end{aligned} \tag{1}$$

where θ is the angle of the disk in radians with respect to the top position, ω is the angular velocity of the disk in rad/s and $u(t)$ represents the system input in volt. Both θ and ω are measured in the available system setup. Furthermore, in eq. (1), M is the pendulum mass, l is the mass distance from the disk center, J is the inertia, g represents the gravitational acceleration, τ and K_m are the modified time constant and motor gain which are based on the lumped electrical dynamics of the motor and the inertia of the pendulum.

The system input voltage is internally saturated. Trhoughout this design assignment, the input voltage should be limited between -3 and +3V. This results in an underactuated setup which makes the control problem somewhat more challenging.

3 System Setup and Simulation

The students are provided with all the required drivers and Matlab scripts to interact with the unbalanced disc setup. More information on these files can be found in the accompanying documentation and readme files.

Due to the large number of students, only limited access is available for the students to perform real-life experiments on the unbalanced disc system. Therefore, a computer simulation of the unbalanced disk is implemented in Simulink (requires Matlab R2018b), Matlab and Python. These simulation environments will be used for most of the design assignment problems.

The simulink model requires the Simulink 3D Animation toolbox to visualize the unbalanced disk setup (https://nl.mathworks.com/products/3d-animation.html). Note that the simulation files and figures should be added to the matlab path. It could be that the animation does not load well on first startup. Close the animation and open it again in case this happens.

Furthermore, make sure to install the correct drivers to work with the unbalanced disc setup. You can find the installation instructions in the experiment drivers - WindowsDcscUSB folder. The driver is unsigned, read the readme instructions on how to install the drivers.

In case you would like to work with the python unbalanced disc environment, make sure to install the unbalanced disc environment for python and the matlab python engine. See the readme file in the gym-unbalanced-disk folder.

The setups can be accessed on campus in FLUX 0.01, the code to access the room is 4423. Make sure to follow all university COVID-19 regulations when accessing this room. Only 2 students per setup are allowed. You need to reserve a timeslot to access the setups. These timeslots can be reserved through Canvas. Do not book more than 2 timeslots ahead in time to make sure that all groups have the opportunity to access the setups.

4 Problems

4.1 Modeling the System Dynamics

The first part of the assignment is to identify the system dynamics using a Gaussian Process (GP) model *and* a (Deep) Artificial Neural Network (ANN) model. Regarding the representation and parametrization of the model structure, a NARX input-output model, a nonlinear state-space or other types of model structures can be considered to capture the system dynamics.

The considered system input is the applied voltage to the system motor. The noisy output is the measured disk angle. Do not use the angular velocity while solving this problem.

The Matlab toolboxes and Python scripts that were used during the practical sessions are recommended be used to complete this objective. Next to the options discussed during the lectures and the practical sessions, the students are free to use any other toolbox or coding environment if desired.

The training/validation data that can be used for this exercise can be obtained through Canvas. A test data set will be provided at a later moment. Students are also allowed to generate their own datasets. However, when doing so, make sure to excite the system sufficiently well such that the dynamics of the system are explored over the full range of interest (e.g. random Gaussian signals, multisine signals, ...). The range of interest for the pendulum identification is a ±120 degree swing around **the stable resting position**. You do not need to consider the case where the pendulum makes a full 360 degree swing (see Figure 2). The final model needs to be tested on the provided test dataset.

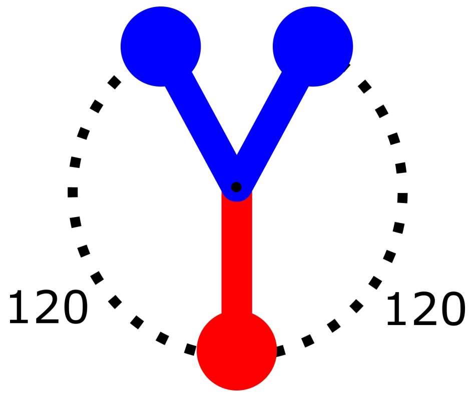

Figure 2: Range of interest for the identified pendulum model.

Study goals:

- Apply the learned concepts of GP and ANN based modeling on an application example under realistic experimental conditions and master the use of available software tools.
- Understand and apply the tuning of hyper-parameters and the resulting accuracy of the model w.r.t. a physical system.
- Understand and apply the system identification cycle (Lecture 1) on data from a practical setup.

4.2 Learning a Swing-Up Policy

This problem should first be solved using the simulation environment, and can be applied on the real-life system in a second step.

The second part of the design project is to obtain a policy (controller) that can swing up the pendulum starting from the stable bottom position and keep the pendulum at the target upright position by rejecting disturbances (see Figure 3). Both the measured angle and angular velocity can be used to solve this problem (i.e. full state feedback).

The students are expected to apply both Q-learning based and actor-critic or model internalization based reinforcement learning methods on the unbalanced disc setup. Explore the advantages and disadvantages of the applied techniques.

Note: the Matlab toolbox PILCO is available through http://mlg.eng.cam.ac.uk/pilco/ .

Study goals:

- Apply Q-learning based reinforcement learning (vanilla Q-Learning, basis function expansion Q-Learning, or DQN) on an application example under realistic experimental conditions and master the use of available software tools.
- Apply the learned concept of reinforcement learning (model internalization or actor-critic methods) on an application example under realistic experimental conditions and master the use of available software tools.
- Understand the tuning of the learning algorithm and the resulting convergence together with the final performance of the control policy w.r.t. a physical system.

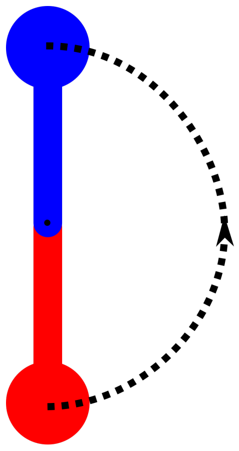

Figure 3: Learning a policy for pendulum swingup.

4.3 Learning a Single Policy for Multiple Targets

This problem should first be solved using the simulation environment.

The final part of the assignment is to extend the swing-up policy to a multiple target policy using actor-critic or model internalization based approaches. Design a policy that can move the pendulum from the bottom position to the upright position, and two other positions $\pm 10°$ to the left and the right of the upright position using a single policy (see Figure 4). Furthermore, improve the obtained policy such that you can also move from one target position to another using one policy (both for swingup and target-to-target movement). Both the measured angle and angular velocity can be used to solve this problem.

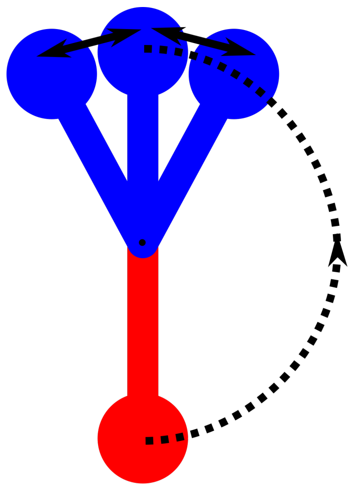

Figure 4: Learning a single policy for multiple targets.

Possible solution: extend the PILCO toolbox to handle multiple targets (see [1]).

Study goals:

- By developing an extension of the simple one-target policy, the student is expected to mastering the theoretical concepts of the advanced RL techniques in more detail.
- By being able to modify the existing software tools, a good understanding of the algorithmic implementation of the applied reinforcement learning tools is obtained.

5 Grading

The grading of the course is explained in the study guide which is available through Canvas.

All group members are expected to have participated in both the modeling and the reinforcement learning part of the assignment.

References

[1] M.P. Deisenroth, P. Englert, J. Peters, and D. Fox. Multi-task policy search for robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3876–3881, Hong Kong, 2014.