Model Predictive Control (5LMB0)

Graded Project

Assignment 1

February 21, 2022

Dr. M. Lazar

MSc. V. D. Reyes Dreke

Table of Contents

Organization

- The **assignment 1 of the homework project** work must be performed in groups of at most two students, who will submit a common report. If you are alone, try to find a partner during lectures. If you cannot find a partner, you can work together in a group of 3, as an exception, but you must submit 2 reports, one for 2 students and 1 for the additional student. The reports should be written independently, i.e., a copy of the same report for different groups of 1 or 2 students is not accepted. All students should be able to answer questions about all the parts of the assignments and reports.

- A report of **maximum 2 pages** showing the results of the project must be submitted in **PDF format** (see the report template). Appendices can be added to include additional plots and software functions.

- **The final report should be submitted accompanied by the source code in a .7z archive by e-mail to v.d.reyes.dreke@tue.nl and m.lazar@tue.nl before March 10, 23:59. The e-mail subject must begin with "Assignment 1 MPC 2022". Both source code files and e-mail subject should contain the surname (without initials) of the students that worked on the report, like MPC_Project_2022_Smith_Ploeg.7z.**

- The assessment will be made based on the quality (accuracy and critical nature of observations) of the report, correctness of the MPC design methods and of the MATLAB code, and creativity in terms of solving the assignments of the project.

- Questions regarding the assignment must be posted on the CANVAS forum or asked during the Seminars. Additional contact hours after the last lecture can be arranged upon request.

Inverted Pendulum System

Figure 1 depicts a diagram of the electrically driven inverted pendulum. This project aims to control the states of the inverted pendulum to an equilibrium point of interest. To control the states, a dc voltage (u) is used as the input of the motor that moves the mass (m). For modelling this system, the angle (q), the angular velocity ($\dot{q}$), and the current motor (i) are considered first, second and third state variables, respectively.

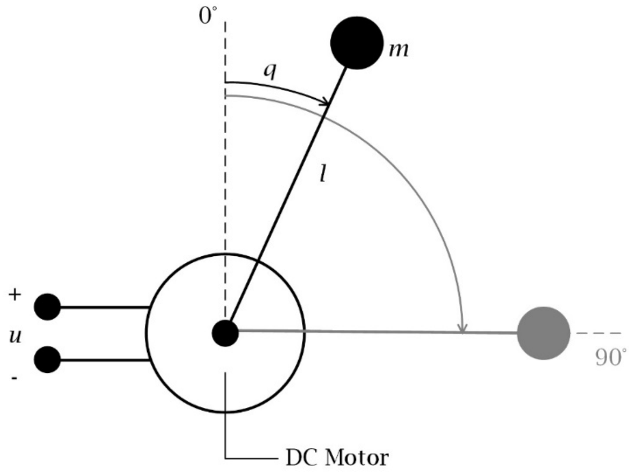

Figure 1 The electrically driven inverted pendulum.

Equations (1) and (2) describe the behavior of the system. In Eq (1), using Newton's second law, the behavior of the forces acting upon the mass is described, as follows

$$J\frac{d\dot{q}(t)}{dt} = mgl \cdot \sin\big(q(t)\big) - b \cdot \dot{q}(t) + k \cdot i(t). \quad (1)$$

In Eq (2), using Kirchoff's law, the voltage of the circuit elements is calculated, as follows

$$L\frac{di(t)}{dt} = -k \cdot \dot{q}(t) - R \cdot i(t) + u(t). \quad (2)$$

The inverted pendulum is subject to the following constraints:

$$\mathbb{X} = \{x \in \mathbb{R}^3 \,|\, |x_1| \le 2\pi, |x_2| \le 12, |x_3| \le 8\}, \quad (3)$$

$$\mathbb{U} = \{u \in \mathbb{R} \,|\, |u| \le 10\}. \quad (4)$$

Table 1 presents the system parameters for this project.

Table 1 System parameters.

Description	Symbol	Value	Unit
Electrical resistance	R	1.0	Ω
Electrical inductance	L	$1.0\ 10^{-3}$	H
Motor constant	k	$6.0\ 10^{-2}$	NA^{-1}
Friction coefficient	b	$1.0\ 10^{-3}$	Nsm^{-1}
Pendulum mass	m	$7.0\ 10^{-2}$	kg
Pendulum length	l	$1.0\ 10^{-1}$	m
Pendulum inertia	$J = ml^2$	$7.0\ 10^{-4}$	kgm^2
Standard gravity	g	9.81	ms^{-2}

Simulink Model Description

The following Simulink model presented in Figure. 2, called “Pendulum_Nonlinear_System.slx”, will be used in some of the assignments and it will be provided to the students. In addition to this Simulink file, we provide a MATLAB’s script, called “MPC_Project_Pendulum_2022.m”, which contains all the parameters needed to run the assignments. The first, second and third initial state are represented with the variables in the MATLAB’s script “**x1_0**”, “**x2_0**” and “**x3_0**”, respectively. Furthermore, these variables are used as the initial conditions in the integrator blocks called “**Angle**”, “**Velocity**” and “**Current**”, respectively. The simulation time of this model is defined by the formula $T\_sim * T\_sample$, where “**T_sim**” is the number of time-steps of the simulation and “**T_sample**” is the sample time.

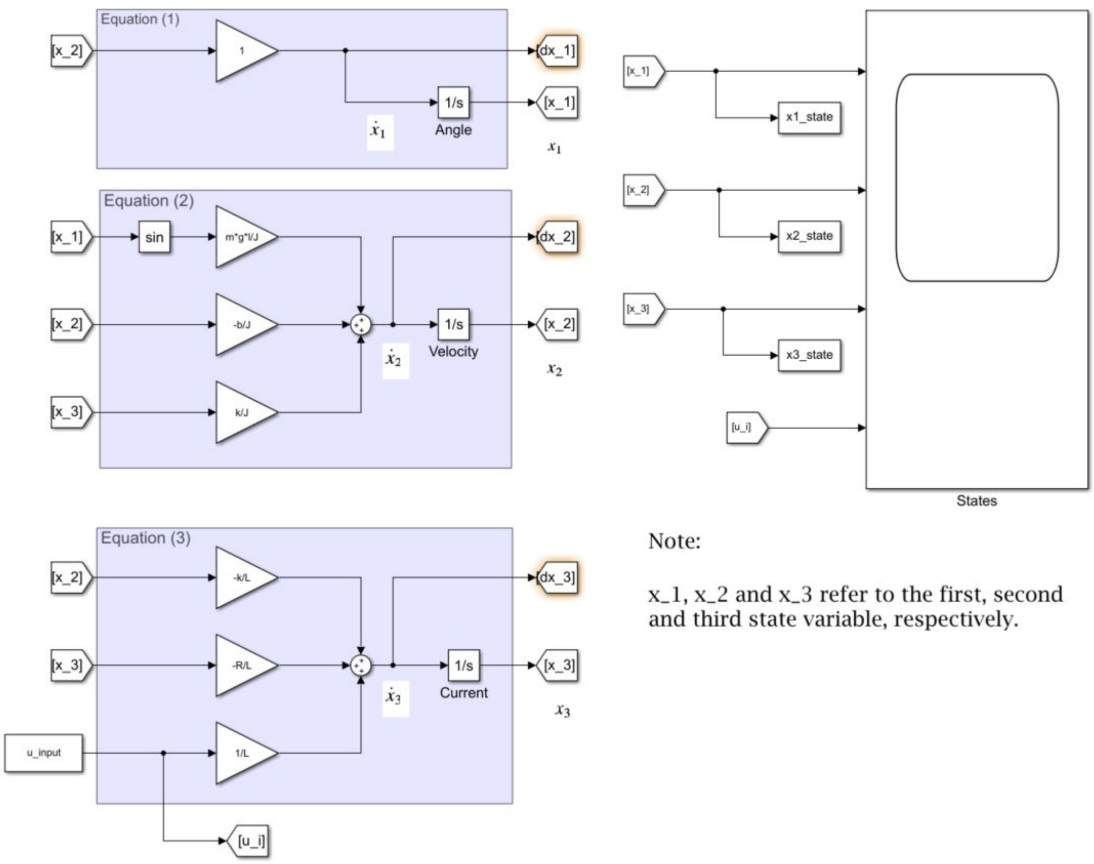

Figure 2 Simulink Model of the Pendulum System

Graded Assignment 1

For grading, there are **10 points for the assignment** in total. Each task of the different parts of the assignment is graded depending on its complexity. With a blue font, it is highlighted the value of each task. The final grade of the homework project is obtained as the average of the individual assignments. Every task is graded based on the correctness of the software and the critical discussion of the results in the report.

The necessary details expected for each assignment in the report are described in Appendix A.

Part 1. State-space model derivation and simulation. (2 points)

Based on (1) and (2), derive a continuous time state space model as follows:

$$\dot{x}(t) = f\big(x(t), u(t)\big). \tag{5}$$

Simulate the provided Simulink implementation of the nonlinear model and confirm that the following combination of states and control inputs describe an equilibrium point:

$$q(0) = \boldsymbol{\pi} \text{ rad}, \quad \dot{q}(0) = 0\ \frac{\text{rad}}{\text{s}}, \quad i(0) = 0 \text{ A}; \quad u(0) = 0 \text{ V}$$

$$q(0) = \frac{\pi}{4} \text{ rad}, \quad \dot{q}(0) = 0\ \frac{\text{rad}}{\text{s}}, \quad i(0) = -0.8093 \text{ A}; \quad u(0) = -0.8093 \text{ V}.$$

Plot the state trajectories as a function of time.

Bonus question: Which of these states is a stable equilibrium point?

Hint: Use the provided MATLAB files for correctness of simulation. Do not change any of the parameters inside the Simulink model, i.e., variable step size, numerical integration method, **except for the initial condition** x1_0.

> Notice: Please check that you are using the new script version called "MPC_Project_Pendulum_2022_update"

Part 2. Linear MPC design based on quadratic programming (QP).

a) Obtaining the prediction model (1 point):

Linearize the nonlinear system at the equilibrium state $\boldsymbol{x_{ss}} = [\boldsymbol{\pi/4} \quad \mathbf{0} \quad \mathbf{-0.8093}\,]^{\boldsymbol{T}}$ and the corresponding equilibrium input $u_{ss} = \mathbf{-0.8093\ V}$. Derive the matrices for the corresponding linear state-space model:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t). \tag{6}$$

Discretize the resulting linear system using the MATLAB function "c2d" and a sample time (**Ts**) of $4\ ms$.

Hint: Use the Jacobian Matrix of $f\big(x(t), u(t)\big)$ to obtain **A** and **B**. More details on linearization can be found in Linearization_Method.pdf.

b) Building the MPC controller (4 points):

For the resulting discrete-time linear model design a linear MPC controller with the following settings:

$$N = 10, \qquad Q = \begin{bmatrix} 120 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \qquad R = 10$$

where N is the prediction horizon, Q is the cost matrix for states, R is the cost for the input.

Given initial states $\boldsymbol{x(0)} = [\mathbf{1.1\pi/4} \quad \mathbf{6} \quad \mathbf{-1}]^{\boldsymbol{T}}$, simulate the linear MPC controller in closed-loop with the linear discrete-time system. Using the same initial sates, simulate the linear MPC in closed-loop with the provided Simulink model of the nonlinear system (5). **The system should stabilize in** $\boldsymbol{x_1^{ss} = \pi/4\, rad,\ x_2^{ss} = 0\, rad/s,\ x_3^{ss} = -0.8093}$ **A and** $\boldsymbol{u_{ss} = -0.8093}$ **V**.

Hint: For simulating the MPC with the provided Simulink model, follow these steps:

- use MATLAB command ***sim()*** to simulate the model,
- use the variable "**u_input**" to send control action to the model,
- use variables "**x1_state**", "**x2_state**", "**x3_state**" to read the current states and
- use variables "**x1_0**", "**x2_0**" and "**x3_0**" to write the next set of initial states into the model.

c) Tuning the MPC (3 points):

Re-tune the linear MPC controller cost matrices Q and R to optimize the convergence of the state trajectories, such as the angle converge faster. Simulate the linear MPC controller in closed-loop with both discrete-time linear model and the nonlinear Simulink model. The initial states are $\boldsymbol{x(0)} = [\mathbf{1.5\pi/4} \quad \mathbf{0} \quad \mathbf{-1.549}]^{\mathrm{T}}$. **The system should stabilize in** $\boldsymbol{x_1^{ss} = \pi/4\, rad,\ x_2^{ss} = 0\, rad/s,\ x_3^{ss} = -0.8093}$ **V and** $\boldsymbol{u_{ss} = -0.8093}$ **V**.

Notice: For avoiding numerical errors during the simulations of the MPC, please formulate the equilibrium point as follows

- $\boldsymbol{x_{ss} = \left[x_{1_{ss}} \quad 0 \quad -\frac{(m*g*l*(sin(x_{1_{ss}})))}{k}\right]^T}$
- $\boldsymbol{u_{ss} = [x_{3_{ss}}]}$

APPENDIX A: Report requirements

The report is limited in size, so please be accurate, succinct and critical about what you write and plot. **The first page is reserved for the title: mention your student ID and name**. 2 additional pages are allowed for the results of the assignments. The necessary details expected for each assignment in the report are described next. Please indicate the course material you have used for the important steps throughout the assignments or external references from literature, if applicable. Regarding the figures, each plot should have properly defined the axis label, title, caption and legend. Also, the plotted signals should be visible.

Part 1.

- Explain and show the formulation of state space model.
- Present a plot of the states as a function of time.
- Based on the simulation results, please present the answer to the following questions:
  - Is $x = [\boldsymbol{\pi}, 0, 0]^T$ a stable equilibrium in open loop (i.e., u(t) =0 for all the simulation time)? Explain your answer.

Part 2.

Part 2a).

- Explain the linearization procedure.
- Present matrices **A** and **B** of the continuous-time representation.

Part 2b).

- Regarding the explanation about the design of the MPC controller explain and present the MATLAB code for:
- Present a figure which contains a plot of the closed-loop trajectories of the states and control inputs using both models.
  - Describe the difference between the closed-loop trajectories with the linear and nonlinear model.

Part 2c).

- Explain how you tuned the new Q and R matrices.
- Plot closed-loop trajectories of states and control inputs from different Q and R matrices.
  - Explain the differences with the results from Part 2b.