Model Predictive Control - EL2700

Assignment 1 : State Feedback Control Design

2020

Automatic control
School of Electrical Engineering and Computer Science
Kungliga Tekniska Högskolan

1 Task: Discrete-time linear design

Throughout these assignments, we will consider the design of a control system for an inverted pendulum on a cart, see Figure 1. The task will be to move the cart from one position to the other while maintaining the pendulum upright. This system has challenging dynamics, similar to the ones that you can find when designing control systems for Segway vehicles, space rockets and many other interesting systems.

In this assignment, we will design a discrete-time linear state-feedback controller for our system. To this end, you will linearize the nonlinear continuous-time dynamics, compute a discrete-time model which describes the state evolution between sampling instances, and use this model to design a linear state feedback controller which achieves the desired control performance.

This task is organized in two parts. In the first part, we will disregard the dynamics of the cart and consider the simpler dynamics of an inverted pendulum. In this way, you will be able to perform calculation by hand and develop insight for how the poles and zeros of the discrete-time system representation change with sampling interval. You should upload a small report to Canvas containing your derivation of the discrete-time state space model and important findings on how sampling interval influences the discrete-time representation.

In the second part, we will revert to numerical computations and design a state feedback controller for the complete system. We will first design a state feedback controller which moves the cart from one position to another while maintaining the pendulum upright. We will then assess the performance of this controller in the presence of disturbance. Next, we will formulate the control law including integral action for dynamic compensation of the disturbance to achieve error-free tracking.

As a part of the assignments, you will be working with Python 3 and the CasADi[1] framework. We encourage you to get familiar with these tools and their examples[2] widely available on the internet.

Nonlinear cart-pendulum model To develop the nonlinear model of the system dynamics, we will use the notation introduced in Figure 1: x is the cart position, θ is the pendulum angle, F is the force applied to the cart, and w is an external disturbance force (force due to wind on the pendulum). By classical mechanics, the system dynamics can be described by the following

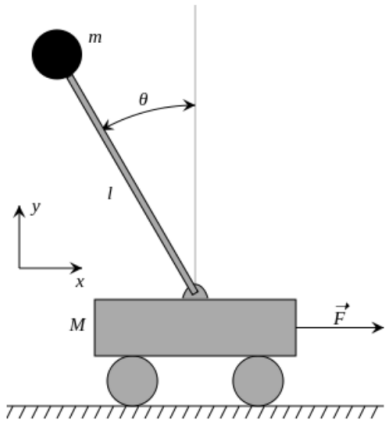

Figure 1: Cart and inverted pendulum schematics

differential equations. Here, M and m denote the mass of the cart and pendulum, respectively, I is the inertia of the pendulum, l is the length of the pendulum, g denotes the gravitational constant,

[1] Website: `https://web.casadi.org/`

[2] CasADi examples: `https://github.com/casadi/casadi/tree/master/docs/examples/python` and `https://github.com/casadi/casadi/wiki/examples`

b_c and b_p represent the coefficient of friction for the cart and the inverted pendulum, respectively.

$$\begin{aligned}(M+m)\ddot{x} - ml\ddot{\theta}\cos\theta + ml\dot{\theta}^2\sin\theta + b_c\dot{x} &= F - w \\ (I+ml^2)\ddot{\theta} + b_p\dot{\theta} - mgl\sin\theta - lw\cos\theta &= ml\ddot{x}\cos\theta\end{aligned} \tag{1}$$

Introducing the state variables $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \theta$, $x_4 = \dot{\theta}$, we can write these equations on state-space form as follows:

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(x_2, x_3, x_4, F, w) \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= f_2(x_2, x_3, x_4, F, w)\end{aligned} \tag{2}$$

where,

$$f_1 = \frac{1}{M+m-\frac{m^2l^2\cos^2 x_3}{I+ml^2}}\left(F + \left(\frac{ml^2\cos^2 x_3}{I+ml^2} - 1\right)w - b_c x_2 - mlx_4^2\sin x_3 + \frac{m^2l^2g\sin x_3\cos x_3}{I+ml^2} - \frac{mlb_p x_4\cos x_3}{I+ml^2}\right)$$

$$f_2 = \frac{1}{I+ml^2-\frac{m^2l^2\cos^2 x_3}{M+m}}\left(\frac{ml\cos x_3}{M+m}F + \frac{Ml\cos x_3}{M+m}w - b_p x_4 + mgl\sin x_3 - \frac{m^2l^2x_4^2\sin x_3\cos x_3}{M+m} - \frac{mlb_c x_2\cos x_3}{M+m}\right)$$

Now, more abstract representation of above equations can be obtained using forward Euler discretization as follows:

$$\begin{aligned}x_{1,k+1} &= x_{1,k} + hx_{2,k} \\ x_{2,k+1} &= x_{2,k} + hf_1(x_{2,k}, x_{3,k}, x_{4,k}, F_k, w_k) \\ x_{3,k+1} &= x_{3,k} + hx_{4,k} \\ x_{4,k+1} &= x_{4,k} + hf_2(x_{2,k}, x_{3,k}, x_{4,k}, F_k, w_k)\end{aligned}$$

with sampling interval h, and where $x_{i,k}$ reads x_i at time-step k, $\forall i = 1, \ldots, 4$.

PART I: Analytical Task

In the first part of this homework, we will perform analytical calculations to linearize and sample a continuous-time system, and to explore the relationship between the locations of the continuous-time system poles and those of its discrete-time counterpart.

Introduction to the analytical task To facilitate analytical calculations, we will disregard the dynamics of the cart and consider the simpler dynamics of the inverted pendulum only. We can derive a model of the pendulum dynamics from the second subsystem in (2) by setting $x_1 = x_2 = 0$ (corresponding to the cart standing still at the initial position), and $w = 0$. In addition, we assume that the mass of the pendulum is much smaller than that of the cart, *i.e.* $M \gg m$. With these assumptions, validate that the pendulum dynamics can be described by a first order ODE system by introducing $x_1 = \theta$, and $x_2 = \dot{\theta}$ as states and considering cart acceleration $u = \frac{F}{M}$ as input

$$\begin{aligned}\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix}x_2 \\ -a_0\sin x_1 - a_1 x_2\end{bmatrix} + \begin{bmatrix}0 \\ b_0 u\cos x_1\end{bmatrix} \\ y &= x_1\end{aligned} \tag{3}$$

with $a_0 = -\frac{mgl}{I+ml^2}$, $a_1 = \frac{b}{I+ml^2}$ and $b_0 = \frac{ml}{I+ml^2}$.

Linear model Verify that the upright pendulum position at rest, $\mathbf{x}_{ref} = [0\ 0]^T$ is an equilibrium point for (3). Linearize (3) around $\mathbf{x}_{ref}$ and write the model on the form

$$\begin{aligned} \frac{d}{dt}\Delta\mathbf{x} &= A_c\Delta\mathbf{x}(t) + B_c\Delta u(t) \\ \Delta y(t) &= C_c\Delta\mathbf{x}(t) \end{aligned} \tag{4}$$

where $\Delta\mathbf{x}(t) = \mathbf{x}(t) - \mathbf{x}_{ref}$, $\Delta u(t) = u(t) - u_{ref}$ and $\Delta y(t) = y(t) - C_c\mathbf{x}_{ref}$. Document your derivations in the report.

Discrete linear model Using zero-order-hold discretization of (4) with sample interval h to determine an equivalent discrete-time linear representation

$$\begin{aligned} \mathbf{x}_{k+1} &= A\mathbf{x}_k + Bu_k \\ y_k &= Cx_k \end{aligned} \tag{5}$$

Here, index k denotes the sampling instance, $\mathbf{x}_k = \Delta\mathbf{x}(kh)$ and $y_k = \Delta y(kh)$. Detail your derivation of A, B and C in the report.

Impact of sampling to the pole locations of the discrete-time system Evaluate the pole locations of the discrete time system for different values of the sampling interval h. Verify the relation between continuous-time poles s_i and discrete-time poles z_i given by

$$z_i = e^{s_i h}$$

PART II: Design Task

Before proceeding, make sure that you have Python 3[3] and Pip 3 installed in your system. If you have both tools, then you can install the dependencies by running the `install_deps.py` script with Python 3. Suggested Python editors are Visual Studio Code, Spyder and PyCharm, but you are free to use any tool of your choice.

Introduction to the design task In this second part, we will consider the complete cart and pendulum system (2). You are no longer expected to perform analytical computations for generic parameters, but only need to consider the specific setup which we simulate. To this end, carefully look into the Python files that are part of the assignment and the classes they implement. The code entry point is `task1.py`, but you will have to complete parts of the `Pendulum` and `Controller` classes.

Design of the state feedback controller For designing the state feedback controller, we assume that the system to be controlled is described by a linear state space model. Therefore, the first step will be to compute linear continuous-time state space model of the non-linear cart-pendulum system in (2) by linearizing the system equations around the equilibrium point, $\mathbf{x}_{eq} = [0\ 0\ 0\ 0]^T$ ignoring the disturbance d. To this end, verify that the linerized continuous time model has the following form:

$$\dot{\mathbf{x}} = A\mathbf{x} + Bu \tag{6}$$

with

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -b_c v_2 & \frac{m^2 l^2 g v_2}{I+ml^2} & -\frac{mlb_p v_2}{I+ml^2} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{mlb_c v_1}{m+M} & mglv_1 & -b_p v_1 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ v_2 \\ 0 \\ \frac{mlv_1}{m+M} \end{pmatrix}$$

[3]Python 3.6 or later: `https://www.python.org/downloads/`

and

$$v_1 = \frac{m+M}{I(m+M)+mMl^2}, \quad v_2 = \frac{I+ml^2}{I(m+M)+mMl^2}$$

Next, implement the model in the function `pendulum_linear_dynamics` and create a discrete time integrator for the linear model in the fuction `set_integrators`, given a sampling time h, implemented as an internal variable `self.dt`. Propose a reasonable sampling interval h. Now, observe that the function `set_discrete_time_system` is capable of retrieving the matrices A and B given the discrete time integrator, implementing the state-space model

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k \tag{7}$$

Now, based on this discretized linear model in (7), derive a state feedback controller of the form

$$u_k = -L\mathbf{x}_k + l_r r_k \tag{8}$$

where r_k is the constant reference position and L and l_r are state feedback and feed forward gain respectively. Implement this controller in the function `control_law`. Design state feedback gain L to place the closed loop poles at the desired location using the `place` function, and implement it in the function `get_closed_loop_gain`.

Validate that the feed forward gain

$$l_r = \frac{1}{C(I-(A-BK))^{-1}B} \tag{9}$$

achieves error-free tracking in stationarity, and implement it in the function `get_feedforward_gain`. Design the feedback and feed forward gains so that the cart moves 90% of the distance between the initial position and the reference within 10 seconds, keeping the pendulum within $\pm 10^\circ$ around the upright position.

Performance in the presence of disturbance input Controllers based on state feedback (8) tracks the reference signal without error at steady state by careful calibration of the gains L, and l_r as you have already observed in the previous task. However, one of the primary objectives of state feedback regulator is to allow good performance in the presence of disturbances. To verify this, add a small constant disturbance w to our cart-pendulum model to mimic the presence of wind force on the inverted pendulum. With the disturbance input, the linear discretized state space model in (7) becomes

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k + B_w w_k \tag{10}$$

Comment on the performance of the state feedback regulator (8) including the disturbance input w in the Simulink model. Comment on the tracking ability of this state feedback regulator in the presence of disturbance. Verify that the steady state error in the output can be computed as

$$\tilde{y}_{ss} = \lim_{k\to\infty} r_k - y_k = -C[I-(A-BL)]^{-1}B_w w$$

To complete this design project, you should upload a small report for the tasks in part I and filled all classes with successful simulation results to Canvas.

Good Luck!