

Optimal control based controllers for a LEGO robot

This project aims to design and implement optimal control based controllers for a 2-DoF LEGO robot. We consider the planar robot with two joints as depicted in Figure 1.

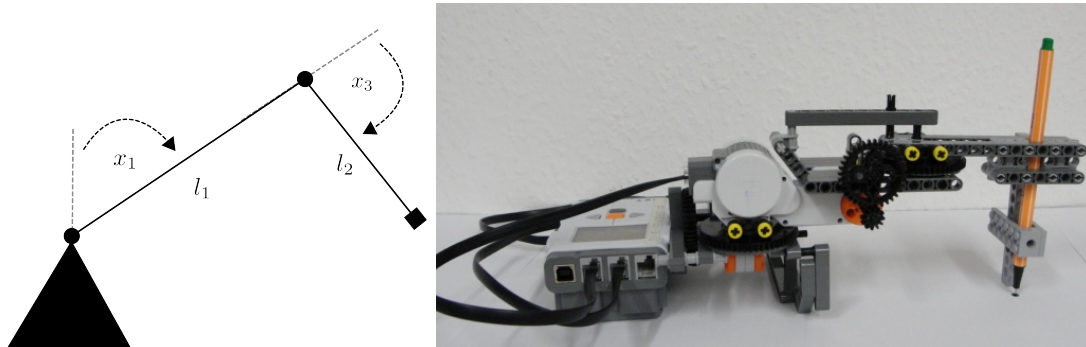


Figure 1: 2 DoF planar robot – sketch and real system.

The *continuous-time* dynamics are given by

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{T_1} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{1}{T_2} \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ \frac{K_1}{T_1} & 0 \\ 0 & 0 \\ 0 & \frac{K_2}{T_2} \end{pmatrix}, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad (1a)$$

where the states $x_{1,2}$ are the angle of the first joint and the corresponding angular velocity and the states $x_{3,4}$ are the angle of the second joint and the corresponding angular velocity. The system parameters are

Table 1: Parameters for 2-DoF Robot.

T_1	0.058
K_1	0.152
T_2	0.062
K_2	0.154

Problem 1: Building of LEGO-Robot via MATLAB

Build the robot shown above. Use the *LEGO MINDSTORMS NXT Support from MATLAB* (or the *RwTH Mindstorms NXT Toolbox*) to assess the quality of the given model of the 2-DoF LEGO robot. In order to do this apply sinusoidal input-signals to the ODE model as well as to the NXT brick.

Problem 2.1: Continuous LQR controller for point-to-point motion

The considered control task is a *rest-to-rest* set-point change from $y_1 = (0, 0)^T$ to $y_2 = (0.2, -0.2)^T$. Design a continuous-time LQR control scheme for the point-to-point motion.

- What is a reasonable choice of the cost function for the considered control task?
- Design an LQR controller and simulate the closed loop behavior.

Problem 2.2: Discrete-time LQR controller design for point-to-point motion

The LEGO brick is a digital controller, so we need to use a discrete-time LQR controller.

- Adapt the control problem to the discrete time case, e.g. by discretizing the dynamics and cost function before the LQR controller design.
- Simulate the behavior of the new scheme and compare it with the continuous-time controller. How does the choice of the sampling time influence the control behavior?

Problem 2.3: Observer design based discrete-time LQR controller

For the simulation one has the full state available, which is not the case for the LEGO robot.

- Extend your control scheme from Problem 2.2. using an observer.
- Simulate the behavior of the new scheme and compare it with the previous schemes.
- Apply your scheme to the LEGO robot. Adjust your cost function/observer, if necessary, to achieve the desired set-point change.

Problem 3: Constrained controller Design for Point-to-point Motion

Now we assume that the states as well as the inputs are subject to constraints:

- inputs are the motor voltages, which are constrained to $u_{1,2} \in [-100, 100]$;
- $x_2, x_4 \in [-4, 4]$.

In order to *guarantee* that the constraints are satisfied, we need to incorporate them into the controller design (model predictive controller (MPC)).

Note that this requires to solve the optimal control problem in MATLAB. There are a few toolboxes available to do this. For this project we will use $\mu AO-MPC$, a free code generation tool. Download $\mu AO-MPC$ and carefully follow the installation instructions:

<http://ifatwww.et.uni-magdeburg.de/syst/muAO-MPC/>.

- Simulate the behavior of your MPC scheme.
- Apply your MPC scheme to the LEGO robot.
- A set of files that might help you finish this task can be found at the following address:
<http://ifatwww.et.uni-magdeburg.de/syst/education/courses/oc/downloads.shtml>
- To control the NXT, you need the latest version of the *RWTH Mindstorms NXT Toolbox*. We have included it in the set of files. If needed, the NXT toolbox can be downloaded from
<http://www.mindstorms.rwth-aachen.de/trac/wiki/Download4.07>.