SE 380 Introduction to Feedback Control
Gennaro Notomista

PROJECT DESCRIPTION
Due date: Nov 30, 2022

Introduction

A robot manipulator consists of a sequence of rigid bodies, called *links*, interconnected by means of articulations, referred to as *joints*, which can be of either *prismatic* or *revolute* type. Each joint provides the robot with a degree of freedom. The prismatic (revolute) joint allows the two links it connects to translate (rotate) relatively to each other along (about) an axis.

Joints are generally moved using electric motors. The control of the joint motion can happen on multiple layers, from the lowest-level control of the current going through the joint motor to the so-called whole-body control, defined to accomplish high level tasks, such as manipulating objects or walking. In between these two extremes, joints can be controlled by supplying joint velocities (a paradigm known as kinematic control or resolved-rate control of robots) or joint torques (for torque-controlled robots).

Objective

Figure 1: Franka Emika robot manipulator

In this project, we consider the robot manipulator in Fig. 1 produced by Franka Emika (`https://www.franka.de/research`, `https://frankaemika.github.io/docs/control_parameters.html`, `https://www.youtube.com/watch?v=_FbhNsRjqdQ`). The robot has 7 revolute joints and we will adopt the resolved-rate control scheme, i.e., we assume to be able to directly control the joint velocities.

> The objective consists of implementing a controller
> to regulate the 7 joints of the robotic manipulator to desired values.

Instructions

The controller will have to be developed using MATLAB/Simulink and will be tested on the real robot manipulator in the RoboHub (`https://uwaterloo.ca/robohub/`). The files provided to develop the controller in simulation are the following:

project_package
- se380_project_run_sim.m........................the main file to run simulation and animation
- se380_project_sim.slx................the Simulink file where to implement the controller block
- robot_model.mat...................data file containing the dynamic parameters of the real robot
- animate_robot.m..script used to animate the robot

The main file is se380_project_run_sim.m. After installing MATLAB with Simulink and the Robotics System Toolbox, this file can be run. It will simulate a dummy controller (open loop, sinusoidal joint velocity for each of the 7 joints) and show an animation of the robot moving under the effect of this controller.

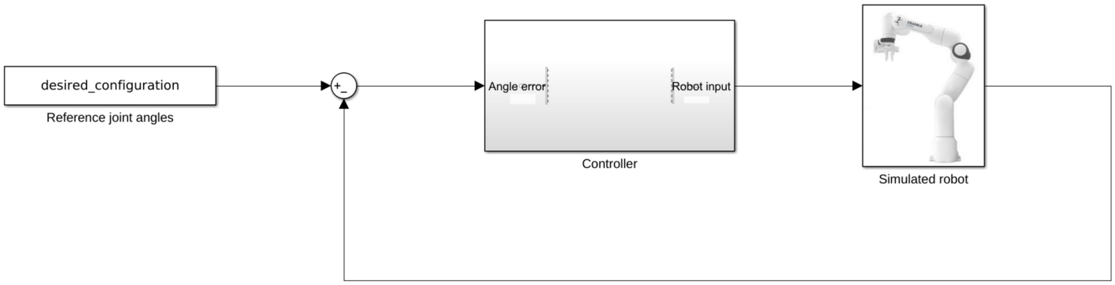

Figure 2: Simulink model of the robot feedback control loop

The simulink file se380_project_sim.slx contains the control loop reported in Fig. 2. The controller will have to be implemented inside the Controller subsystem which has the following input/output interface:

- Input: The input signal is a 7-dimensional vector, whose i-th component is the error between the desired and actual angle of joint i
- Output: The output signal must be a 7-dimensional vector, whose i-th component is the velocity of joint i

The controller must be designed
using either *lead-lag compensators* or *pole placement*.

Performance evaluation

The desired performance of the system are the following:

- Overshoot[1] on each of the joint angles must be 0%
- Settling time[2] must be less than 10s
- Steady-state error must be 0

[1] Maximum amount the system response overshoots its final value, divided by its final value.

[2] Time required for the response curve to remain confined within 2% of the final value.