Experiment Procedure
Robot Modelling, Identification, and Control

SRM - Serial Robot Modelling

April 23, 2024

⚠ **IMPORTANT:** It is essential that you carry out the following steps before starting the experiment!

1.) For every Simulink model, select "`Fixed-Step`" as the solver with a variable `Sample Time` $T_s = 0.001$. You can set this under "`Model Configuration Parameters`" in the upper bar.

2.) Avoid hardcoded values, i.e. only use variables within Simulink and define the corresponding values in the Workspace.

3.) Deactivate the check mark at "`Limit data points to...`" in Scopes in order not to lose any data points during longer simulation times.

4.) If you need to compare two systems, the easiest way is to copy the original system and make the changes to the copy. So you always have both versions available.

5.) For "`To Workspace`" blocks, select "`Array`" as the storage format, since they are the easiest to handle.

6.) If the function of a command is not clear, use MATLAB Help.

7.) Use the "`clear`" command in your main script to clean up your workspace before performing a task and avoid errors due to old data.

1 Experiment procedure

Consider the two-link planar manipulator shown in Fig. 1. Here the gravity vector points in the negative y direction

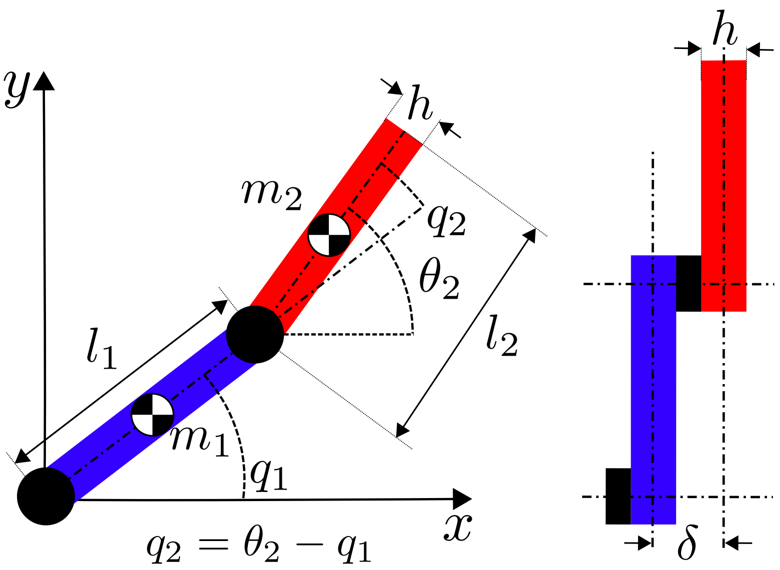

Figure 1: A planar manipulator with two links.

⚠ **IMPORTANT:** In Moodle you are provided with a MATLAB Live Editor file which will guide you through the intermediate steps needed to complete the tasks. Fill in the lines where you see the legend: `<YOUR CODE HERE>`. **Items marked with a ★ must be included in your experiment report.**

T1 (**1 P**) Get the MDH parameters of the two-link arm in Fig. 1.
★ Write them down in your report in a table similar to the one below

Table 1: MDH-Parameters for the double pendulum in in Fig. 1

i	Link 1	Link 2
$a_i[m]$		
$d_i[m]$		
$\alpha_i[rad]$		
$\theta_i[rad]$		

T2 (**2 P**) Find the homogeneous transformation matrices ${}^0\boldsymbol{T}_1$ and ${}^1\boldsymbol{T}_2$ — see eq. (5) in the script — and the Jacobians from the center of mass of each link to the base frame, i.e. ${}^0\boldsymbol{J}(\boldsymbol{q})_{1,CoM_1}$ and ${}^0\boldsymbol{J}(\boldsymbol{q})_{2,CoM_2}$, see eq. (7).
★ Write the corresponding expressions in your report.

Hint: You might want to define all the needed homogeneous transformation matrices including those that identify the position and orientation of the link center of mass relative to the link frame.
Hint: You can right-click on the output of the MATLAB Live Editor to copy the formulas as LATEX expressions.

T3 (**5 P**) Get the required terms to describe the forward dynamics of the two-link arm (i.e eq. (21) from the script) using the Lagrangian method and the MATLAB Symbolic Toolbox (**use the provided MATLAB Live Script**). Verify that the properties in Section 2.4.5 from the script are satisfied.
★ Report the corresponding expressions for $\boldsymbol{M}(\boldsymbol{q})$, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ and $\boldsymbol{g}(\boldsymbol{q})$

Hint: You can use any of the options given in Section 2.4.4 of the script to get the mass, Coriolis, and gravity terms.

Hint: Use the well known expression for the moments of inertia of a prismatic shape with uniform density. Use Fig. 2 as a reference.

Hint: You can evaluate your symbolic expressions and compare the results with those from the system generated by the MATLAB Robotics System Toolbox. Just run the last cell in the script to generate it. You will see that you have an object named `twoLinkArm` which includes methods such as `inverseDynamics`, `massMatrix`, `velocityProduct` and `gravityTorque` that can get you numerical versions of the corresponding values[1]. **Make sure your modified Denavit-Hartenberg parameters are correct.**

Hint: The functions `getMass`, `getCoriolis`, and `getGravity` that evaluate your symbolic expressions at a given joint angles (velocities) are given in the provided MATLAB Live Editor file. This is an example of how to numerically compare your mass matrix with that computed by the Robotics System Toolbox

```
% Mass matrix M

% Mind the column vector format
q  = rand(2,1);
dq = rand(2,1);

% Your matrix
disp('M = ')
disp(getMass(M,q))
% MATLAB RST matrix
disp('M = ')
disp(twoLinkRobot.massMatrix(q))
```

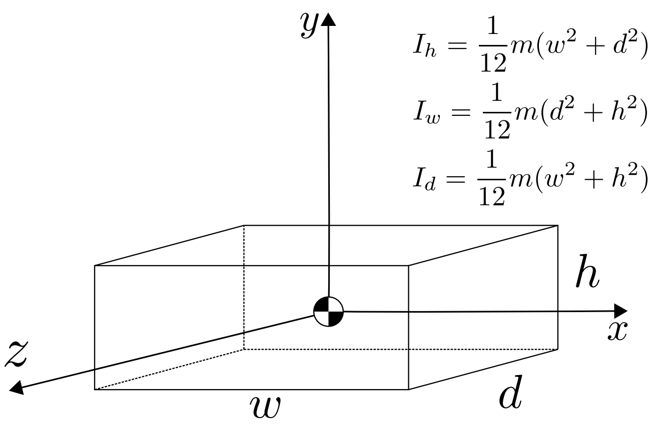

Figure 2: Moments of inertia of a prismatic shape.

IMPORTANT: Bear in mind that, unlike the script, this double pendulum is not made of point masses. You have to use the inertia tensors of the links and the necessary Jacobians in your calculation.

T4 (**3 P**) Use Simulink to simulate the motion of the pendulum using the found dynamics. **Consider the initial conditions to be zero, i.e. $q(0) = 0$ and $\dot{q}(0) = 0$.**
★ Include in your report an image of your Simulink block diagram and a plot of the joint angles vs. time for a simulation time of 50 seconds.
Hint: Use the `matlabFunctionBlock` function (see the provided Live Script) to create a Simulink function block containing the results of your symbolic calculations. You will get blocks whose input are joints angles and/or velocities.

IMPORTANT: Use the provided Simulink template file.

[1] See MATLAB's documentation for further information (`https://de.mathworks.com/help/pdf_doc/robotics/robotics_ug.pdf`).

T5 (**1 P**) Verify that energy is conserved.
★ Include in your report a plot of the sum of potential and kinetic energy during the simulation time.

T6 (**1 P**) Add viscous friction to the joints.
★ Include in your report a plot of the total energy for this case. Comment on your observations. What do you expect will happen to the energy?
💡 **Hint:** Use the following viscous friction model $\tau_{f,i} = \nu_i \dot{q}_i$ where $\nu_i = 0.5 \frac{kgm^2}{s}$ is the friction coefficient.