Experiment Procedure
Robot Modelling, Identification, and Control

IPI - Robot Inertial Parameter Identification

November 14, 2024

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

1.) Select "`Fixed-Step`" as the solver for your Simulink model with a variable Sample Time $T_s = 0.001$. You will select this later depending on the task. 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 them outside in a central script which is called by the simulation via callback[1].

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.

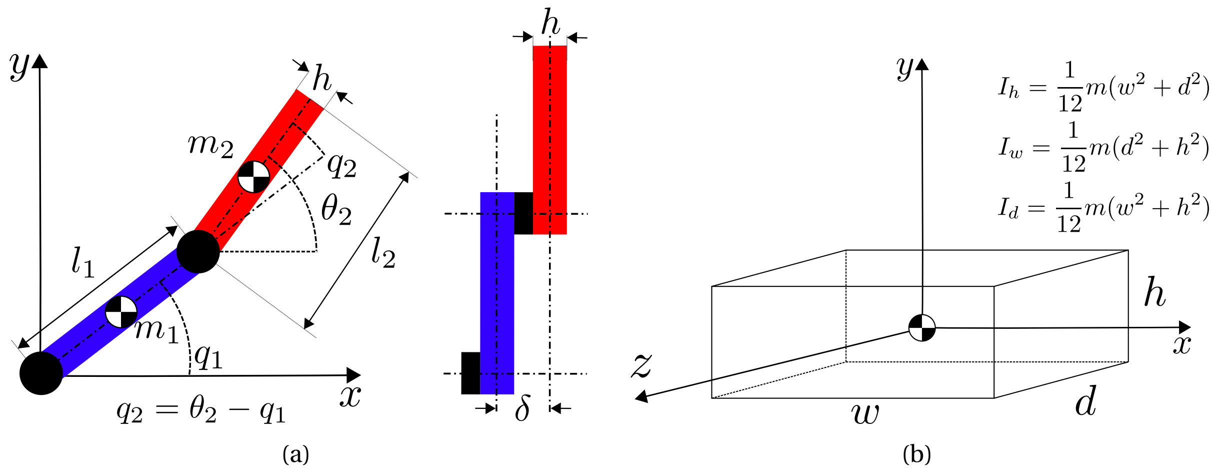
6.) If the function of a command is not clear, use MATLAB Help.

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.

⚠ **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**.

1 Experimental procedure

Consider the double pendulum shown in Fig. 1a. The moments of inertia of the prismatic links are given computed as shown in Fig. 1b.



y
x
h
m_2
q_2
θ_2
l_1
l_2
m_1
q_1
$q_2 = \theta_2 - q_1$
δ
(a)
$I_h = \frac{1}{12}m(w^2 + d^2)$
$I_w = \frac{1}{12}m(d^2 + h^2)$
$I_d = \frac{1}{12}m(w^2 + h^2)$
z
w
d
(b)

Figure 1: 2-DoF planar manipulator: a manipulator schematic, b moments of inertia of a prismatic shape.

T1 (**10 P**) **Regressor form of the dynamics equations**: in this task the goal is to get a numerical comparison between the robot dynamics from the the first lab experiment, the LIP form and minimal-parameter form. You will use MATLAB's Symbolic Math Toolbox to create the required expressions. Use the MATLAB Live Script `ipi_template.mlx`.

a) To construct the LIP form, eq (13), you will first get the expressions for the kinetic and potential energies in terms of the inertial parameters $\boldsymbol{X}$. Use for this eqs. (2) and (3) from the script. Add the definitions in section `TASK 1a`.

b) In section `TASK 1b` form the geometry vectors $\boldsymbol{t}$ and $\boldsymbol{u}$ after eqs. (15) and (16)

c) Use the Lagrangian equations (17) in section `TASK 1c` to construct the regressor matrix $\boldsymbol{C}$
Hint: You can use two loops to achieve this.

d) Now compute the re-grouped parameter vector $\boldsymbol{X}'$ in section `TASK 1d`. For this purpose, the entries of the vector are created by recursively calling eqs. (20), starting from the last robot link.

e) The entries of the newly computed parameter vector $\boldsymbol{X}'$ can be further reduced taking into account eq. (19). In section `TASK 1e`, the index of the elements that meet these conditions will be used to remove the corresponding columns from the regressor matrix $\boldsymbol{C}$ and from the re-grouped parameter vector $\boldsymbol{X}'$.
Hint: Further reduction can be accomplished by checking the rules described in Sec. 2.3.1.
Save the regressor matrix $\boldsymbol{C}$ and inertial parameter vector $\boldsymbol{X}$ as well as the reduced regressor matrix $\boldsymbol{C}_b$ and minimal-parameter vector $\boldsymbol{\beta}_b$, correspondingly.

f) Execute now the code in `TASK 1f`. This will assign numerical values to your matrices and vectors and will create Simulink blocks for your regressor and reduced regressor matrices. Use the file `ipi_model_comparison_template.slx` to test your models. In it you are already provided with the inverse dynamics torque $\boldsymbol{\tau}_{id}$ of the form $\boldsymbol{\tau}_{id} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})$, **you need to compute the regressor form torque** $\boldsymbol{\tau}_{reg}$ with $\boldsymbol{\tau}_{reg} = \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{X}$ **and the minimal-parameter form torque** $\boldsymbol{\tau}_{mp}$ with $\boldsymbol{\tau}_{mp} = \boldsymbol{C}_b(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}})\boldsymbol{\beta}_b$ and check that they all give the same joint torques.

★ **Provide in your report the corresponding expressions for the reduced regressor matrix $\boldsymbol{C}_b$ and the minimal parameter vector $\boldsymbol{\beta}_b$.**

★ **Provide in your report a plots of the joint torques $\boldsymbol{\tau}_{id}$, $\boldsymbol{\tau}_{reg}$, and $\boldsymbol{\tau}_{mp}$.**

T2 (**10 P**) **Trajectory optimization**: in this task, optimal trajectories are created, which will later be used in the virtual identification experiment. The trajectory optimization is started via the script `RMIC_traj_optim_config_start.m`, in which static parameters of the optimization problem are defined. This script contains the function call to `RMIC_call_optim_traj.m`, which calls initial calculations and the optimization algorithm `fmincon()`. In turn, `fmincon()` calls the function `RMIC_optim_problem.m`, which contains the description of the optimization problem. Main components of this function are `RMIC_calc_fourier_traj_poly.m` for the calculation of the parameterizable joint angle trajectories, and `RMIC_arm_inf_matrix.m`, which contains a sample-based evaluation of the robot dynamics in parametric form. At the end of `RMIC_optim_problem.m` the condition number of the information matrix is formed; w.r.t. which the optimization is performed. In addition, nonlinear constraints can be passed to `fmincon()` via `RMIC_nonlConstraints.m`, which can be considered when solving the optimization problem.

a) Open the function `RMIC_calc_fourier_traj_poly.m`. Add the derivatives of (32) in section `% << T2-A >>` at the marked positions with the labels `% <<YOUR CODE HERE>>`.

b) By calling the function `RMIC_calc_fourier_traj_poly.m` within the optimization algorithm the parameters a_{i,m_f} are given by the optimization algorithm. In addition, boundary conditions (34) are to be fulfilled. For this purpose, the Fourier series (32) is superimposed with a fifth degree polynomial given in (33). The parameters are determined using a system of equations $\boldsymbol{Ax} = \boldsymbol{b}$. Fill out the missing indices when calculating (33) in section `% << T2-B >>` (look for the labels `% <<YOUR CODE HERE>>`) which corresponds to the parameter vector $\boldsymbol{b}$. **Which variable in the code corresponds to the matrix $\boldsymbol{A}$?**

c) Set up the outputs of the function `q`, `qD`, and `qDD` using (31). Add the missing entries in section `% << T2-C >>` marked by the labels `% <<YOUR CODE HERE>>`.

d) Open the function `RMIC_arm_inf_matrix.m`, with which the information matrix is to be set up according to equation (25). Create a function named `RMIC_regressor_base_matrix` to get the minimal-parameter regressor matrix you created in the previous task and add it in section `% << T2-D >>`. **Do not forget** to specify any arguments that your function might need.
Hint: You can use the provided template for this function, also remember that all the kinematic parameters are contained in `constPar`.

e) Create the information matrix $\boldsymbol{F}$ see (30), based on `C_tmp` and, if necessary, `C_frct`, do this in the section with the `% << T2-E >>` label.
⚠ **IMPORTANT:** Make sure to specify the correct row indices.

Execute the script `RMIC_traj_optim_config_start.m`. Look at the results of the trajectory optimization. **How do you interpret the results?**
Hint: The exection of `fmincon()` might take a few minutes.
⚠ **IMPORTANT:** At the end of this part, animations of the planar manipulator executing the computed trajectory should be shown.

★ **Provide in your report the plots of the generated excitation trajectories, i.e. q, $\dot{q}$, $\ddot{q}$.**

T3 (**10 P**) **Parameter identification**: in this task, a virtual identification experiment is executed. With the collected measurement data (joint positions, velocities and accelerations as well as joint torques), the minimal parameters of the planar manipulator will be estimated. Please note that this experiment includes the effects of friction and measurement noise.

Open the Simulink model `RMIC_ident_experiment.slx`. You will see the planar manipulator model, a simple controller block and a trajectory block. **Do not modify any of these blocks!**
⚠ **IMPORTANT:** Notice that the recorded variables are corrupted by noise, i.e. the yellow blocks.

a) Open the file `RMIC_ident_experiment_start.m`. This file will allow you to run the Simulink model to collect data. Make sure that the parameters of your optimal trajectory are in the `constPar.a_opt` array. Fill out the section with the `% << T3-A >>` label to run the model and collect data.

Hint: Your measurements are rather noisy, use data averaging to get a filtered version of your measurements before getting the minimal-parameter vector.

b) After collecting the required data compute the information matrix $\boldsymbol{F}$ and the measurement vector $\boldsymbol{b}$ in section `% << T3-B >>`. Use equation (26) to identify the minimal-parameter vector `beta_b_star`. **How do you interpret the results?**

c) Run section `% << T3-C >>` of the script `RMIC_ident_experiment_start.m` to view the plots of the results. A figure in MATLAB will show the comparison between measured and modeled torque based on identification. **How do you evaluate the results?**

d) To test the quality of the identification you are provided with a second trajectory. Set the parameter `constPar.test_trajectory` to one. Now run section `% << T3-D >>` to run the model again using the identified minimal-parameter vector from the previous task and generate a comparison of the generated torques for this new trajectory. Plots of the results will be displayed. **How do you interpret the results?**

★ **Include in your report the values of your identified base parameter vector β_b^*.**

★ **Include in your report plots of the measured and model-based torques from section 3c and 3d.**