

AMORO Lab Part 3: Simulation of a Biglide mechanism

Damien SIX
damien.six@ls2n.fr

Released: August 20, 2021

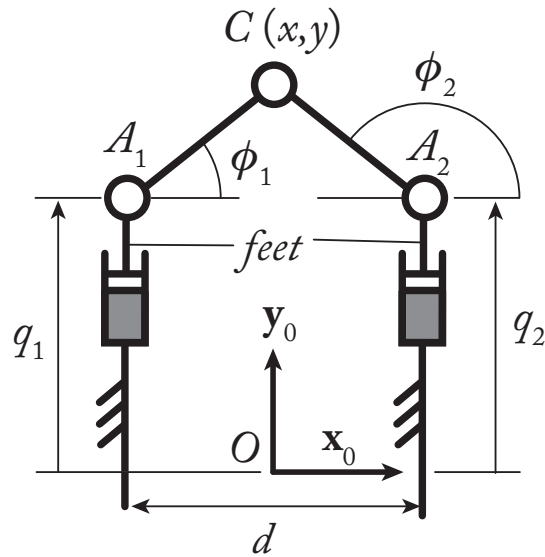


Figure 1: Kinematic model of the Biglide. The prismatic joints are actuated.

1 Objective

The main objective of the present lab is to compute the geometric, kinematic and dynamic models of a Biglide mechanism and to compare them with the results obtained with GAZEBO. Then, a controller will be designed to track a trajectory in simulation.

The kinematic architecture of the Biglide is shown in Fig.1. For the mechanism of the Gazebo mock-up, the geometric parameters are:

- $d = 0.4$ m
- $l_{A_1C} = 0.2828427$ m

- $l_{A2C} = 0.2828427$ m

and the base dynamic parameters are: The base dynamic parameters are:

- $m_p = 3$ kg the mass of the end-effector
- $m_f = 1$ kg the mass of each foot

2 Evaluation

The deliverable for this lab is a zip file containing:

- A detailed computation of the biglide mechanism models (PDF file).
- A detailed report (with comment, figures, etc.) of your work (PDF file). Concerning figures, the scripts output scopes that allow you to check the validity of your models. They are here to help you during the lab. Please do not use them directly in your final report and take the time to make nicer plots with the correct labels, scales, etc. (no lazy print screen). Learning how to make nice figures is also part of your training.
- The *biglide_models.py* file: all models should match with Gazebo simulation. The input-output of the functions in *biglide_models.py* must not be modified from the original file.
- The *control.py* file: when launched, the file must perform the desired trajectory using computed torque control.

This report is the main way to have a clear feedback about your work, so its quality and content will have a great impact on the final grade. Your work must be clearly detailed and commented.

The script files will be tested and their effectiveness will also be part of the grade. To be evaluated your file must be free of any error. You can test if your models pass the evaluation process by running the *model_eval.py* script.

After the end of the four lab sessions, you have **two weeks** to upload the deliverable as a .zip file at the following address <https://www.dropbox.com/request/AgEcJ96a4b1EdvbEftot>.

3 Models of the Biglide

Compute and compare with the Gazebo simulation the following models of the Biglide. All the following models must be validated

- The Direct Geometric Model
- The Direct Geometric Model for passive joints
- The Inverse Geometric Model
- The First Order Kinematic Model
- The First Order Kinematic Model for passive joints
- The Inverse First Order Kinematic Model

- The Second Order Kinematic Model
- The Second Order Kinematic Model for passive joints
- The Inverse Second Order Kinematic Model
- The inverse Dynamic Model

All your model should be filled in the *biglide.models.py* file without changing the input-output of the pre-defined functions.

4 Computed Torque Control simulation

Using the Robot class input and output, code in *control.py* a computed torque control to track a desired trajectory. The trajectory can be of your choice as long as it is tracked properly.

5 Crossing type 2 singularities (bonus points)

- Design a trajectory that crosses a type 2 singularity for this robot.
- What happen when you try to track this trajectory using a computed torque control law?
- Design a trajectory that can cross safely the type 2 singularity. Display the theoretical torque inputs for this trajectory using the inverse Dynamic Model.
- Try to track this trajectory using Gazebo and the computed torque control. What happen (and why)?

The file *control.py* provided with your final report must perform the tracking of a trajectory using computed torque control as asked in section 4. Please use an other script for the trials with the type 2 singularity.