# Lab 4: Kinematics and Dynamics of a Biglide

Released: October 08, 2018

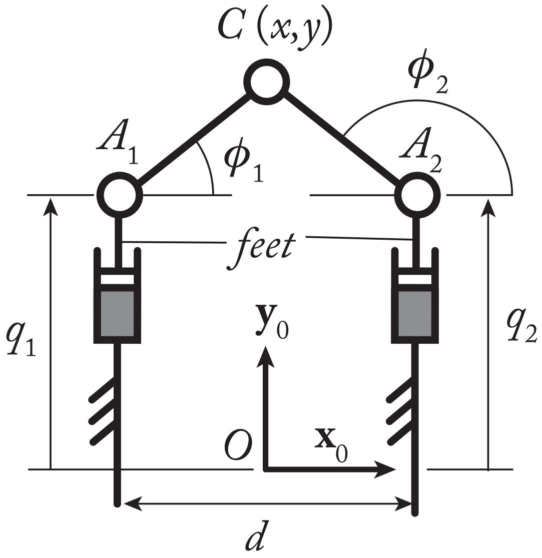


Figure 1: Kinematic model of the Biglide

## 1 Objective

The main objective of the present lab is to model in the MATLAB environment the geometric, kinematic and dynamic models of a Biglide and to compare them with the results obtained with ADAMS. Then, controllers will be designed to track a trajectories in simulation.

The ADAMS model used for this lab is named "Biglide.bin".

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

- $d = 0.4$ m
- $l_{A1C} = 0.3606$ m
- $l_{A2C} = 0.3606$ m

The two prismatic joints are actuated.
The base dynamic parameters are:

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

All other dynamic parameters are neglected.

## 2 Evaluation

This lab is evaluated. The deliverables for this lab are due for **November 8**. A link to a .zip file will be sent to the examiner (guillaume.jeanneau@ec-nantes.fr or abhilash-uday.nayak@ec-nantes.fr). If you have no DropBox or equivalent to send the file, please contact your examiner to get a link to drop your file **BEFORE THE DEADLINE**. The deadline cannot be delayed. A penalty will be applied on the grade of any late report (-2 per day of delay). The .zip file must contain :

- A report containing all the theoretical material, the methodology, and the results obtained with **appropriate comments**.
- The Biglide.bin file with the splines for singularity crossing (correctly named)
- A simulink file where the models are compared
- A simulink file for each controller with a test trajectory
- The plants .m files associated to each simulink files

The files must be organized and named appropriately. An explanation must be given in the report. It is not the job of the examiner to understand how you organized your files. If you are not sure how to present them, ask him during the lab!

## 3 Preparation of the LAB

It is strongly recommended, **BEFORE** the lab, to compute the mathematical expressions of the geometric, kinematic and dynamic models of the Biglide mechanism.

## 4 Creation of the ADAMS plant

First, let us create the ADAMS plant that can be used in co-simulation with MATLAB.

- Create the following 15 state variables for the output.
  - the position of point C (*x, y*) (the end-effector)
  - the velocity of point C (*xd, yd*)

  - the acceleration of point C (*xdd, ydd*)
  - the passive joint variables *phi1, phi2* as defined in Fig. 1
  - the passive joint velocities *phi1d, phi2d*
  - the passive joint accelerations *phi1dd, phi2dd*
  - the forces exerted by the two actuators *f1* and *f2*

- Go to the active joint toolbox and impose a motion 0.1*sin(time) on *q1* and -0.1*sin(time) on *q2*.
- Create the plant output
- Export the plant into MATLAB. (You may have to create a fake input variable).

All these steps have been described in the ADAMS tutorial and are not detailed here. The students will refer to the tutorial if they don't remember how to make an ADAMS mockup.

# 5 Direct Kinematic analysis of the Biglide mechanism

Now, we will simulate the kinematic behavior of the Biglide mechanism and compare it with MATLAB.

## 5.1 Geometric analysis

Write into MATLAB function blocks the geometric models of the Biglide mechanism:

- The direct geometric model.
- The model which gives the passive angle position.

**Questions**

In the SIMULINK template, input $q1$ and $q2$ corresponding to the inputs designed in ADAMS. Compare the results of the outputs ($x$,$y$) obtained from the MATLAB function and the ADAMS simulation. Same question for the passive angles.

## 5.2 Velocity analysis of the Biglide mechanism

Write into the MATLAB function blocks the kinematic models of the Biglide mechanism:

- The direct kinematic model.
- The model that provides the passive angle velocities.

**Question**

In the SIMULINK template, input $q11d$ and $q21d$ corresponding to the inputs designed in ADAMS. Compare the results of the outputs (end effector, passive angle velocities) obtained from the MATLAB function and the ADAMS simulation.

### 5.3 Acceleration analysis of the Biglide mechanism

Complete your SIMULINK scheme with the second-order kinematic models of the five-bar mechanism:

- The direct kinematic model.
- The model that provides the passive angle accelerations.

**Question**

Compare the results of the outputs (end-effector, passive angle accelerations) obtained from the MATLAB function and the ADAMS simulation.

## 6 Dynamic analysis of the five-bar mechanism

Now, we will simulate the dynamic behavior of the Biglide mechanism. Complete the SIMULINK scheme with a MATLAB function giving the inverse dynamic model of the Biglide mechanism.

**Question**

Run the simulation with the inputs on MATLAB similar to ADAMS. Compare the outputs obtained from your MATLAB function and the simulation with ADAMS.

## 7 Control co-simulation

### 7.1 Kinematic control law

- Create a two input variables *q1d* and *q2d* and affect them as control of the joint motion.
- Create a new plant with input *q1d* and *q2d* and output the joint position *q1* and *q2*.
- On SIMULINK create a controller scheme to apply a kinematic control.

**Questions**

Define a trajectory to test your controller.

Run the co-simulation and plot the trajectory tracking.

Change the communication interval time. Does it affect the tracking error? Why?

**Bonus (if you have time):** Design a controller with acceleration inputs *q1dd* and *q2dd*.

### 7.2 Computed Torque Control law

- Create a two input variables *f1* and *f2*. Remove the motions imposed on joints and apply forces driven by *f1* and *f2* instead.
- Create a new plant with input *f1* and *f2* and output the joint position *q1* and *q2*.

**Questions** Compute the dynamic model under the form $\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_a + \mathbf{c}$ for the Biglide mechanism.

On SIMULINK create a controller scheme to apply a CTC control.

Define a trajectory to test your controller.

Run the co-simulation and check the trajectory tracking.

Define a trajectory crossing a Type 2 singularity. What happen, why?

# 8 Trajectory crossing a Type 2 singularity

In this section the trajectories are directly tested on ADAMS. No co-simulation is required.

**Questions**

Compute the criterion necessary to cross a Type 2 singularity for the Biglide mechanism.

Design two trajectories crossing a Type 2 singularity on the Biglide mechanism, one respecting the criterion and the other not respecting it. Input them on ADAMS via splines and plot the torques necessary to follow those trajectories, comment.