Lab 2: Kinematics of a five-bar mechanism

Released: August 26, 2016

Figure 1: The DexTAR robot (Mecademic)

1 Objective

The main objective of the present lab is to model in the MATLAB environment the geometric and kinematic models of a five-bar mechanism and to compare them with the results obtained with ADAMS. Then, a controller will be designed to track a trajectory in simulation.

The ADAMS model used for this lab is named "MODEL_DexTAR.bin". It is based on the DexTAR robot designed by Mecademic (see Fig.1). When parametrazing ADAMS, you can hide parts (as Base_Assembly) find the markers more easily.

The kinematic architecture of the five-bar mechanism is shown in Fig.2. For the mechanism of the ADAMS mock-up, the geometric parameters are:

- Bar length (all bars are equal length): $l = 0.09$ m
- Distance between the two active joints: $d = 0.118$ m

Joints located at A11 and A21 are actuated.

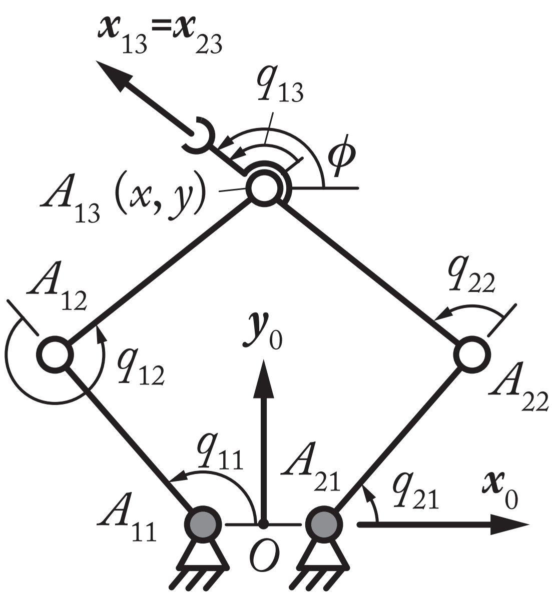

Figure 2: Kinematic architecture of the five-bar mechanism

2 Preparation of the LAB

It is mandatory, **BEFORE** the lab, to compute the mathematical expressions of the geometric and kinematic models of the five-bar mechanism. All the necessary elements are given in the lecture note and in the section 4 of this lab instructions.

3 Creation of the ADAMS plant

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

- Create the following 15 state variables for the output. While creating the outputs, take care that the reference marker to be taken is the MARKER_ref attached to the ground (and not the default ADAMS reference marker).
  - the position of point A13 (*x, y*) (the end-effector)
  - the velocity of point A13 (*xd, yd*)
  - the acceleration of point A13 (*xdd, ydd*)
  - the passive joint variables *q12, q22* and *q13*, as defined in Fig. 2
  - the passive joint velocities *q12d, q22d* and *q13d*
  - the passive joint accelerations *q12dd, q22dd* and *q13dd*.
- Go to the active joint toolbox and impose a motion sin(time) on *q11* and -sin(time) on *q21*.
- Create the plant output

- Export the plant into MATLAB. (You may have to create a fake input variable).
- Insert the ADAMS plant in the SIMULINK template given with the lab material “fivebar.slx”.

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.

4 Direct Kinematic analysis of the five-bar mechanism

Now, we will simulate the kinematic behavior of the five-bar mechanism and compare it with MATLAB. All the present models have been explained during the lectures and are detailed in the lecture notes.

4.1 Geometric analysis

Write into the MATLAB function blocks (in the SIMULINK template) the geometric models of the five-bar mechanism:

- The direct geometric model which gives [x, y] = dgm(q11,q21,gamma). In the lecture note, the direct geometric model gives two solutions, but the two solutions can switch even if a Type 2 singularity is not crossed. To obtain a direct geometric model such that each γ (+1 or -1) corresponds exactly to a working mode, the following procedure must be followed (see Fig. 3)

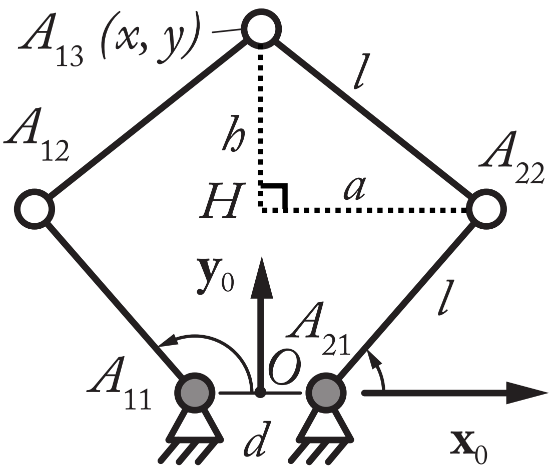

Figure 3: Kinematic architecture of the five-bar mechanism

  - Compute the vector $\overrightarrow{A_{22}H}$ (H is the mid-point of $A_{12}A_{22}$)

- There are two solutions for the point A_{13}, obtained by the 90° rotation of the vector $\overrightarrow{A_{22}H}$, normalization and multiplying by the distance h, giving

$$\overrightarrow{HA_{13}} = \gamma \frac{h}{a} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \overrightarrow{A_{22}H}$$

- Compute the solution $\overrightarrow{OA_{13}} = \overrightarrow{OA_{21}} + \overrightarrow{A_{21}A_{22}} + \overrightarrow{A_{22}H} + \overrightarrow{HA_{13}}$

- The model which gives the passive angle position [q12,q22,q13] = passive_angles(x,y,q11,q21)

Questions

In the SIMULINK template, input $q11$ and $q21$ corresponding to the inputs designed in ADAMS. Compare the results of the outputs (x,y) obtained from the MATLAB function and the ADAMS simulation. Are they equal? If not, why?

Same question for the passive angles.

4.2 Velocity analysis of the five-bar mechanism

Write into the MATLAB function blocks (in the SIMULINK template) the kinematic models of the five-bar mechanism:

- The direct kinematic model which gives [xd, yd] = dkm(q11d,q21d,x,y,q11, q21,q21,q22,q13)

- The model that provides the passive angle velocities [q12d,q22d,q13d] = passive_angleskm(xd,yd,q11d,q21d,x,y,q11,q21,q12,q22,q13)

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.

4.3 Acceleration analysis of the five-bar mechanism

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

- The direct kinematic model which gives he direct kinematic model which gives [xdd, ydd] = dkm2(q11dd,q21dd, xd,yd,q11d,q21d,q12d,q22d,q13d, x,y,q11,q21,q12,q22,q13)

- The model that provides the passive angle velocities [q12dd,q22dd,q13dd] = passive_angleskm2(xdd,ydd,q11dd,q21dd,xd,yd,q11d,q21d,q12d,q22d, q13d,x,y,q11,q21,q12,q22,q13)

Question

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

5 Trajectory generation and tracking

5.1 Recalls

Recall 1: A polynomial trajectory is defined with initials conditions and final conditions (eventually intermediate). The number of initial and final conditions will determine the order of the polynomial used for the trajectory (number of conditions -1).

As an example, displacement in x with initial and final positions and velocities (4 conditions) requires a 3$^{\text{rd}}$ order polynomial function.

$$x = a_1 t^3 + a_2 t^2 + a_3 t + a_4$$

The initials and final conditions are expressed as

$$\begin{aligned}
x(t_i) &= a_1 t_i^3 + a_2 t_i^2 + a_3 t_i + a_4 \\
\dot{x}(t_i) &= 3a_1 t_i^2 + 2a_2 t_i + a_3 \\
x(t_f) &= a_1 t_f^3 + a_2 t_f^2 + a_3 t_f + a_4 \\
\dot{x}(t_f) &= 3a_1 t_f^2 + 2a_2 t_f + a_3
\end{aligned}$$

which can be put in the form of a linear system

$$\mathbf{Pa} = \mathbf{c}$$

with $\mathbf{a} = (a_1, a_2, a_3, a_4)^T$ containing the coefficients of the polynomial function. Solving this linear system gives the polynomial coefficients for the trajectory.

A trajectory between two points $A(x_A, y_A)$ and B(x_B, y_B), with null velocity and acceleration at initial and final position, in the time interval $[0,\ t_f]$ is given by a fifth order polynomial function. This specific case can be expressed as:

$$\begin{aligned}
x(t) &= x_A + s(t)(x_B - x_A) \\
y(t) &= y_A + s(t)(y_B - y_A)
\end{aligned}$$

with

$$s(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5$$

Recall 2: Kinematic control in Cartesian space

The desired objective is to track a Trajectory in Cartesian space. We will establish a kinematic control through ADAMS co-simulation. The desired trajectory is expressed by $\mathbf{x}_t$. The tracking error is given by $\mathbf{e} = \mathbf{x} - \mathbf{x}_t$, where $\mathbf{x}$ is the coordinates of the end-effector. The objective is to make this error converge towards zero using the the active joint velocity $\dot{\mathbf{q}}$ as an input. We want to obtain the closed-loop behavior

$$\dot{\mathbf{e}} = -\lambda \mathbf{e}$$

with $\lambda > 0$. I.e.

$$-\mathbf{A}^{-1}\mathbf{B}\dot{\mathbf{q}} = \dot{\mathbf{x}}_t - \lambda \mathbf{e}$$

with $\mathbf{B}$ and $\mathbf{A}$ issued from the first order kinematic equation of the robot (same notations than in the lecture note). Then the desired control input is

$$\dot{\mathbf{q}} = -\mathbf{B}^{-1}\mathbf{A}(\dot{\mathbf{x}}_t - \lambda \mathbf{e})$$

5.2 Control co-simulation

- Create a two input variables *q11d* and *q21d* and affect them as control of the joint motion. (Example **velo(time)=VARVAL(q11d)**)
- Create a new plant with input *q11d* and *q21d* and output the joint position *q11* and *q21*. The output are chosen to correspond to the real robot (no direct measure of the end-effector position).
- On SIMULINK create a controller scheme to apply a kinematic control.

Questions

Define a trajectory between the initial position (0.09,0.06796322) and the position (0,0.1) in 2s.

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 *q11dd* and *q21dd*.

SAVE ALL YOUR WORK FOR THE NEXT LAB!