Lab 3: Dynamics 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 dynamic model of a five-bar mechanism and then to compare them with the results obtained with ADAMS. We will then compare trajectories crossing a Type 2 singularity. A dynamic controller will also be implemented.

The model used is the same than in the previous lab (same ADAMS file).

The base dynamic parameters of the mockup are:

- zz_{11R} = 0.013704725227 kg.m^2 (gathers inertia of actuated joint 1)
- zz_{21R} = 0.014534710433 kg.m^2 (gathers inertia of actuated joint 2)
- m_{pR}= 0.5071616221 kg (mass of the end-effector)

All other dynamic parameters are neglected.

2 Preparation of the LAB

It is mandatory, **BEFORE** the lab, to compute the mathematical expressions of the dynamic model and the inverse geometric model of the five-bar mechanism. All the necessary elements are given in the lecture note. For the inverse geometric model, you will use a method similar to the computation of the DGM in the previous lab. Do not copy the method from the lecture note as you will have to handle discontinuities when generating trajectories.

3 Preparation of the ADAMS plant

Complete the ADAMS plant from the previous lab

- Create two additional output state variables *Tau1* and *Tau2* the input torque of the actuators.
- Go to the active joint toolbox and impose a motion 0.5*sin(time) on *q11* and -0.5*sin(time) on *q21*.
- Create the plant output (Including the plant outputs designed in the previous lab).
- Export the plant into MATLAB. (You may have to create a fake input variable).
- Insert the ADAMS plant in the SIMULINK used for the kinematic analysis.

4 Dynamic analysis of the five-bar mechanism

Now, we will simulate the dynamic behavior of the five-bar mechanism and compare it with MATLAB. The inverse dynamic model have been explained during the lectures and are detailed in the lecture notes. Complete the SIMULINK scheme with a MATLAB function giving the inverse dynamic model of the five-bar mechanism. You will have to use outputs from the kinematic models designed in the previous lab.

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.

5 Trajectory crossing a Type 2 singularity

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

To input a trajectory in ADAMS the following procedure must be applied

- On MATLAB, generate the vectors of the end-effector trajectory, sampled at 1 ms.

- Compute the joint trajectories through the Inverse Geometric Model. Test your Inverse Geometric Model with your DGM.
- Create two matrices $\mathbf{Q}_1 = [\mathbf{time}, \mathbf{q}_1]$ and $\mathbf{Q}_2 = [\mathbf{time}, \mathbf{q}_2]$
- Create two .txt files with the matlab command "save('q1.txt', 'Q1', '-ASCII','-append');"
  Be cautious: This command will not delete the file if it already exists (you must delete it before).
- In ADAMS **File/Import**. Select "Test Data" - "Create splines" and browse your 'q1.txt' file. Set "Independent Column Index" to 1. Click OK.
- A spline is created in **Elements/Data Elements**. Rename the spline. Do it again for *q2*.
- In the joints motion, use the function builder to define the displacement as a function of the spline. Use **Spline/Akima Fitting Method**.

Questions

Design a trajectory from point (0,0.15) to point (0.05,0.1) in 1s. Simulate the trajectory directly in ADAMS. Plot the torques necessary to follow this trajectory.

Design a trajectory from point (0,0.15) to point (0.05,0.0) in 1s. Plot the torques necessary to follow this trajectory. What happen? Why?

Recall the criterion necessary to cross a Type 2 singularity for the 5-bar mechanism (see lecture note).

Design a new trajectory able to cross the Type 2 singularity in the point (0,0.084492603226555) and simulate it on ADAMS. Plot the torques and comment.

6 Dynamic control simulation

6.1 Introduction to Computed Torque Control

The Computed Torque Control is based on the feedback linearization of the system through the inverse dynamic model. The dynamic model of a parallel robot can be written under the generic form

$$\boldsymbol{\tau} = \mathbf{M}\ddot{\mathbf{q}}_a + \mathbf{c}$$

$\mathbf{M}$ being the inertia matrix, definite positive and $\mathbf{c}$ the vectore of Coriolisis and Centrifugal effects. Considering the input $\boldsymbol{\tau}$ this system in non-linear. The auxiliary control is defined $\mathbf{v}$

$$\mathbf{v} = \mathbf{M}^{-1}(\boldsymbol{\tau} - \mathbf{c})$$

Then this auxiliary input correspond to the robot acceleration.

$$\mathbf{v} = \ddot{\mathbf{q}}_a$$

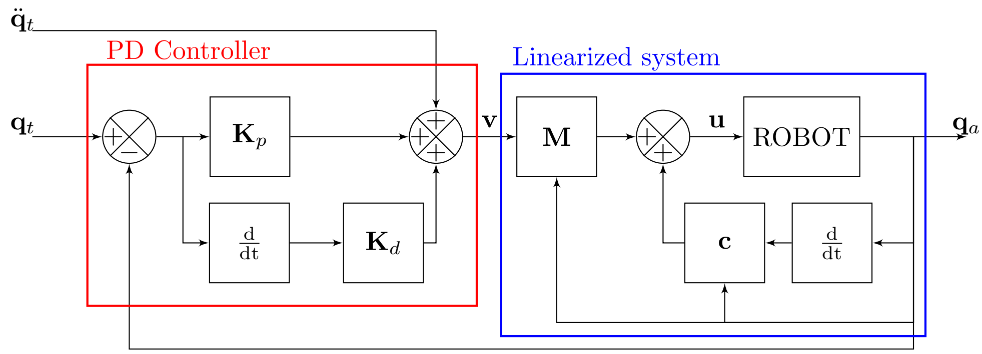

Figure 2: Computed Torque Control Scheme

A PD control law can be applied on this auxiliary input

$$\mathbf{v} = \ddot{\mathbf{q}}_t + \mathbf{K}_d(\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_a) + \mathbf{K}_p(\mathbf{q}_t - \mathbf{q}_a)$$

With $\mathbf{K}_p$ et $\mathbf{K}_d$ definite positive (usually positive diagone matices). The torque input is deduced from this auxiliary control law

$$\begin{aligned}\boldsymbol{\tau} &= \mathbf{M}\mathbf{v} + \mathbf{c} \\ &= \mathbf{M}(\ddot{\mathbf{q}}_t + \mathbf{K}_d(\dot{\mathbf{q}}_t - \dot{\mathbf{q}}_a) + \mathbf{K}_p(\mathbf{q}_t - \mathbf{q}_a)) + \mathbf{c}\end{aligned}$$

Fig.2 shows the controller scheme. With $\tilde{\mathbf{q}} = \mathbf{q}_{ref} - \mathbf{q}_a$, the closed-loop equation of the control system is

$$\ddot{\tilde{\mathbf{q}}} + \mathbf{K}_d\dot{\tilde{\mathbf{q}}} + \mathbf{K}_p\tilde{\mathbf{q}} = \mathbf{0}$$

Ensuring the convergence of the error $\tilde{\mathbf{q}}$ toward $\mathbf{0}$.

6.2 Computed Torque Control design and simulation for the five-bar mechanism

- Create a two input variables *Tau1* and *Tau2*. Remove the motions imposed on joints and apply torques driven by *Tau1* and *Tau2* instead.
- Create a new plant with input *Tau1* and *Tau2* and output the joint position *q11* and *q21*. This is the closest simulation to the real robot.

Questions Compute the matrices $\mathbf{M}$ and $\mathbf{c}$ for the five-bar mechanism.
On SIMULINK create a controller scheme to apply a CTC control.
Define a trajectory between the initial position (0.09,0.06796322) and the position (0,0.1) in 2s.
Run the co-simulation and check the trajectory tracking.
Define a trajectory crossing a Type 2 singularity. What happen, why?