INF3480 - spring 2010

Compulsory exercise 2

Deadline: Monday, April 12th (before midnight)

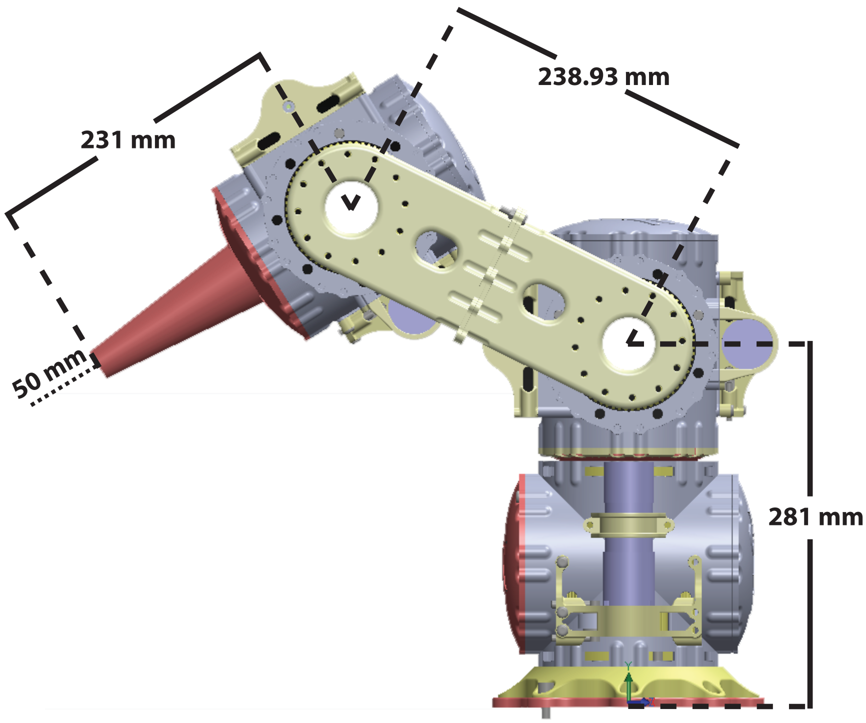

Figure 1: The X2 robot with measurements

Introduction

In this exercise we will keep working on the X2 robot, displayed in Figure 1. As was explained in the prevouis exercise: "The robot consists of three identical parts, the first one is attached to the surface beneath the robot, and does not rotate. The next part is attached on top of the first part, and rotates along the vertical axis. The second and third parts are connected with a link. In each end of this link, there is a revolute joint. A pen will be attached to the tip of the outermost part."

Figure 1 shows the necessary measurements to work with the robot. The dotted line at the end of the robot denotes the pen that will be attached there.

1

Derive the inverse kinematic equations for the X2 robot, and show the different steps. How many solutions exist for the joint angles given an arbitrary position of the tip of the pen?

2

Implement the forward kinematics and the inverse kinematics as matlab functions.
(you derived the forward kinematic equation in compulsory exercise 1)

a) The forward kinematics function takes 3 joint angles as input, and gives the corresponding cartesian coordinates for the tip of the pen as output.

b) The inverse kinematics function takes the position of the tip of the pen as input, and gives the corresponding joint configuration(s) as output.

c) Use the matlab functions to show how you can verify that the inverse and forward kinematics are correctly derived.

Some tips on using matlab is given at the bottom of this document.

3

a) Derive the Jacobian for the X2 robot.

b) Use the Jacobian matrix to find the singular configurations of the robot

c) Give an evaluation of the results from exercise 3b, based on the practical understanding of the anatomy of the robot (in other words: Did your results in 3b match what you expected to be singular configurations for the robot? Why / Why not?).

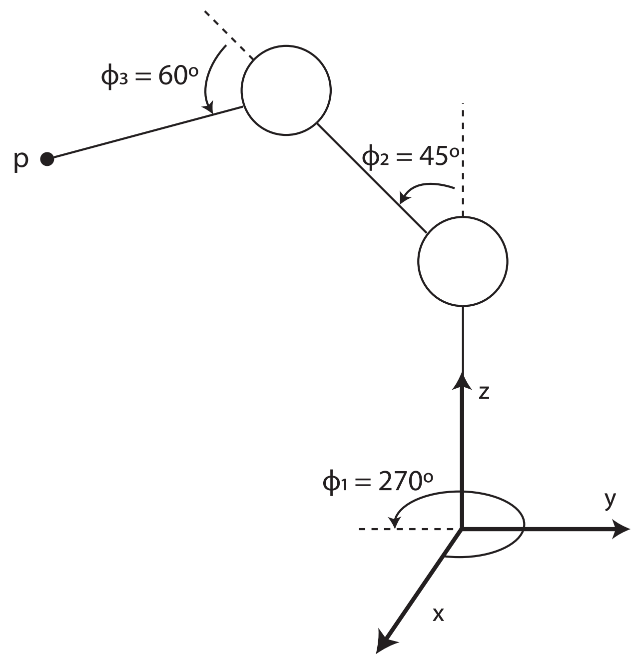

Figure 2: Robot configuration for 4b)

4

a) Implement the Jacobian matrix as a matlab function. This function takes the instant joint angles and joint velocities as input, and gives a 3-dimensional vector of cartesian velocities of the tip of the pen as output.

b) Point p is located at the end-effector of the robot (the tip of the pen). We adjust the robot as displayed in figure 2, where $\phi_1 = 270°, \phi_2 = 45°, \phi_3 = 60°$.
(These angles are not to be used directly, you have to figure out the correct θ-angles corresponding to your placement of the joint coordinate frames yourself.)

Given the configuration in figure 2, and the joint speed vector $\dot{q} = [\dot{\theta_1}\ \dot{\theta_2}\ \dot{\theta_3}]$, where $\dot{\theta_1} = 0.1 rad/s$ and $\dot{\theta_2} = \dot{\theta_3} = 0.05 rad/s$, use the matlab function for the Jacobian matrix to find the speed of the end-effector at point p (tip of the pen).

Requirements:
Each student must hand in their own assignment, and you are required to have read the following requirements to student submissions at the department of informatics: http://www.ifi.uio.no/studinf/skjemaer/declaration.pdf

Your submission should be sent as a zip-file. Send it by e-mail to **vokjelse[at]student.matnat.uio.no**.

Your submission must include:

- The three matlab-functions for forward kinematics, inverse kinematics and velocity kinematics (.m-files).
  Use comments in the code to document the functions!
- A pdf-document with answers to the questions.

Name the file: "inf3480-ex2-*your_username.zip*".

Deadline: Monday, April 12th

Matlab is well documented. Use the help functions and the Internet to find help to solve this assignment. If you use sources from the Internet directly in your code, these must be refered to through comments in your code. A link to the matlab compendium from the course MAT1110 is available on the course website (Norwegian only).

Matlab is available on all the computers at ifi. The ROBIN-lab on the 3rd floor is available to all the students taking this course.

A simple example of matlab use to those of you who have never seen matlab before is given on the next page.

If you prefer working from your own computer, you can use matlab by logging on to windows.ifi.uio.no using remote desktop.

For those of you that have never used matlab before, it is one of the simplest programming environments available. Here is a simple example of how to create and use a function that calculates the sum of all the elements in two vectors:

Code for the matlabfunction myVectorSum.m:

```
function answer = myVectorSum(v1,v2)
%Comment describing my function

    answer = 0;                                  %initial answer value = 0

    for x = 1:length(v1)
        answer = answer + v1(x);                 %Adding the elements of vector v1
    end

    for x = 1:length(v2)
        answer = answer + v2(x);                 %Adding the elements of vector v2
    end

end
```

To use this function, the file myVectorSum.m must be in the working directory or in the matlab searchpath. It is used by calling it from the command line with the necessary arguments (here vectors a and b):

```
>>
>> a = [1,2,3];          (the semicolon prevents output to the screen)
>> b = [4,5,6,7];
>> c = myVectorSum(a,b)

c =

    28

>>
```