Master of Electromechanical Engineering

Academic year 2014-2015

ICT & Mechatronics
Group D2 : Robot Arm

Emanuel De Peuter
Quinten Martens
Frederik Ostyn
Wouter Staessens
Thibault Stepman
Dries T'Jampens

Supervisors: Dr. Ir. F. De Belie
Prof. Dr. Ir. R. De Keyser
Dr. Ir. C. Copot
Dr. Ir. B. Van de Wiele

Contents

1 Introduction

2 Task Description

3 Robot Kinematics

In this part, the kinematics of a robot arm will be discussed. First, the Denavit-Hartenberg description will briefly be reviewed. Secondly, the calibration of the robot with the use of a MATLAB script will be presented. Thirdly, the range of the robot will be calculated.

3.1 Mathematical framework

A position and rotation of a link of the robot (or equivalently, the local robot coordinate system) as seen from a fixed global coordinate sytem can be represented by matrix g

$$g = \begin{pmatrix} q & p \\ 0 & 1 \end{pmatrix} \tag{1}$$

where p and q represent respectively the coordinates of the origin of the robot coordinate system in the global system and the rotation matrix from global to robot frame.

If the robot has multiple links (which is the case here), the matrix g_{end} representing the position and the orientation of the end effector can be build by multiplication of matrices

$$g_{end} = h_1 h_2 h_3 h_4 h_5 \tag{2}$$

where h_i describes the position and rotation of link i as seen from the coordinate system of link $i-1$ with $i = 0$ the global (external) coordinate system, $i = 1$ the base, $i = 2$ the shoulder, $i = 3$ the elbow, $i = 4$ the wrist and $i = 5$ the tip of the gripper (or pencil if applicable[1]). The matrices h_i were already given in the Arduino template and will therefore not be repeated here.

The above operation described in Eq. (2) deals with the direct kinematics of the robot arm where the rotation of the servos is known and the matrix g_{end} looked for. The inverse kinematics problem involves finding the correct servo angles if a coordinate and orientation of the end effector in the global system is given. This problem is far more difficult and only an approximate solution based on the Denavit-Hartenberg approach will be presented.

The inverse kinematics problem comes down to solving

$$b = Mx \tag{3}$$

for x, namely $x = M^{-1}b$. Here, x and b are the components of the velocity matrix μ, defined as

$$\mu = \begin{pmatrix} 0 & -\omega_z & \omega_y & v_x \\ \omega_z & 0 & -\omega_x & v_y \\ -\omega_y & \omega_x & 0 & v_z \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{4}$$

[1] In the Arduino program $h_5 = h_{5b} h_{pen}$ with h_5 the matrix describing the gripper tip in the frame of the wrist and h_{pen} the pencil tip in the frame of the gripper tip.

in different bases

$$\mu = \sum_{i=1}^{5} b_i B_i = \sum_{i=1}^{5} x_i X_i \tag{5}$$

where ω_x, ω_y, ω_z, v_x, v_y and v_z are the angular velocity and the velocity of the end effector arm in its own reference coordinate system[2]. This matrix μ was calculated based on the property that $dg_{end}(t)/dt = g(t) \cdot \mu(t)$

In the above mathematical construction, M is the transformation matrix between the bases X and B, respectively associated with coordinates x and b. The columns of M contain the components of the base vectors X_i as expressed in the base B. The base vectors B_i are

$$B_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, B_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$B_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, B_4 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, B_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and the base vectors X_i are

$$X_1 = g_{end}^{-1} h_1 s_1 h_2 h_3 h_4 h_{5b} h_{pen}$$

$$X_2 = g_{end}^{-1} h_1 h_2 s_2 h_3 h_4 h_{5b} h_{pen}$$

$$X_3 = g_{end}^{-1} h_1 s_1 h_2 h_3 s_3 h_4 h_{5b} h_{pen}$$

$$X_4 = g_{end}^{-1} h_1 s_1 h_2 h_3 h_4 s_4 h_{5b} h_{pen}$$

$$X_5 = g_{end}^{-1} h_1 s_1 h_2 h_3 h_4 h_{5b} s_5 h_{pen}$$

with

$$s_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, s_2 = -s_3 = s_4 \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, s_5 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

As both bases are now defined, the coordinate transformation matrix M can be calculated.

The only thing left to do is to find the components b_i of μ in the base B_i. Recalling that

$$dg_{end}(t)/dt = g(t) \cdot \mu(t) \tag{6}$$

and assuming that in a certain step dt the end effector should go from g_{end} to g_{des} with g_{des} the desired state, the general solution in the case of μ being constant during the step can be found as being

$$g_{des} \equiv g(t + dt) = g(t) e^{\mu dt} \equiv g_{end} e^{\mu dt} \tag{7}$$

[2]Remark that $\omega_z \equiv 0$ for this robot as it can turn its wrist up and down and clockwise/anticlockwise, but not in the $z-$direction.

where $e^{\mu dt}$ is a matrix exponential. As we know the base B_i, we know the components b_i of μ if we know μ. Eq. (7) however is not trivial to solve. Therefore $e^{\mu dt}$ is approximated by restricting the number of terms in the Taylor expansion

$$e^{\mu dt} \approx e + \mu dt \tag{8}$$

with e the unit matrix. The solution for μdt is[3] then

$$dsE := \mu dt \approx g_{end}^{-1} g_{des} - e \tag{9}$$

From this result, the $b_i dt$ components can be calculated, which completes the mathematical frame work to find du_i as

$$xdt = M^{-1} bdt \Leftrightarrow du_i = \sum_j M_{ij}^{-1} b_j dt \tag{10}$$

3.2 Calibration of the robot

In order to have a tool to calibrate the robot and to simulate how the presented mathematical framework describes the robot, a MATLAB script containing the direct and inverse algorithms was written. The direct kinematics part calculates and plots g_{end} and the appropriate coordinate systems. The inverse kinematics part asks for a state g_{des} and returns the servo angles that do the job.

3.3 Range of the robot arm

Dat met die cilinders... Wouter?

4 Algorithm to go from point A to B

5 Sensor system design

6 Algorithm in order to avoid obstacles

7 Testing of the combined system

8 Conclusion

[3] μdt is called dsE in the Arduino code.

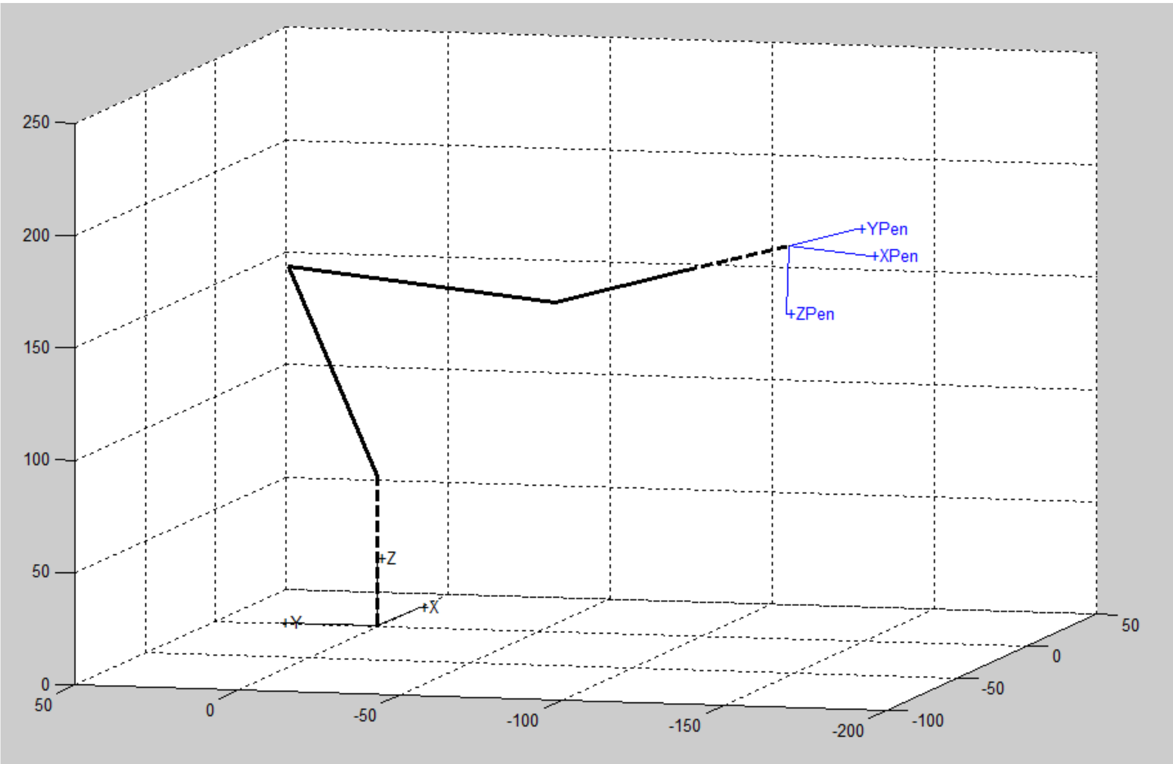

Figure 1: Matlab simulation of the robot arm and the global and end effector coordinate system. This simulation was without the pencil; the end effector is the tip of the gripper.