Departamento de Engenharia Eletrotécnica e de Computadores
Secção de Robótica e Manufatura Integrada

Robotics

2014/2015 – 2nd Semester

Lab Work 1

Due date: 12/04/2015

Forward and Inverse Kinematics

Regente: José António Barata de Oliveira
jab@uninova.pt

Monitor: Ricardo Peres
ra.peres@campus.fct.unl.pt

Introduction

This work aims to provide students with the means to consolidate the fundamental concepts described during the theoretical classes regarding the control of robot joints, more specifically the Rob3 robot. The position of these joints can be set according to their technological parameters, such as sensor resolution and servo-motor control. At this point we can establish the relationship between the number of steps provided in each joint and the corresponding angular positions. For instance, 127 steps in joint 1 correspond to 0 radians. 255 steps translates into about Pi/2 radians. Both the joints and the gripper have a resolution of 8 bits.

The manipulation of the angular position of the robot's joints is not enough when using robots in real world applications. Therefore, we need to map the aggregate of each joint's position to corresponding **(x, y, z)** Cartesian coordinates and orientation **(roll, pitch, yaw)**. The second part of this work consists on implementing the robot's forward and inverse kinematics.

Problem Requirements

As mentioned before, the first part of the problem consists on the direct manipulation of each of Rob3's joints. The necessary functional requirements (FR) are two fold, considering that we need to position each joint using both step inputs and angular coordinates, as specified on table 1.

Table 1- Functional requirements

F.R.	Steps domain	Angular domain (radians)
1	Rotate a joint for a given step position.	Rotate a joint for a given angular position.
2	Rotate all joints for each corresponding given steps.	Same but using angular positions.
3	Rotate a joint for a given step position at a given speed.	Same with angular positions.
4	Rotate all joints for each corresponding given steps with given speeds.	Same with angular positions.
5	Determine the step position of a given joint.	Same with angular positions.
6	Determine the step position of all joints.	Same with angular positions.
7	Read digital input ports.	
8	Write into digital output ports.	

The second part consists on implementing the forward and inverse kinematics of robot Rob3. The functional requirements for this part are mentioned in table 2. Keep in mind that the previous F.R. need to be implemented first.

Table 2- High-level functional requirements

F.R.	Description
9	Movement of each joint (ϴ1, ϴ2,…, ϴ6). Each movement should also provide the position of the robot's gripper in Cartesian robot coordinates (x,y,z) and orientation (roll, pitch, yaw).
10	Movement of the robot's gripper to a position specified by Cartesian coordinates (x,y,z) and orientation (roll, pitch, yaw)
11	Calibrate the robot.
12	Extra: Perform a series of consecutive movements so that the robot mimics a certain behavior (e.g. handshake, dinosaur…). You can add other extras as you seem fit.

Your program should **provide an adequate user interface for the execution of each functional requirement**.

There is a Rob3 simulator to help during the development and testing stages. This simulator allows the detection of most design and implementation flaws, which otherwise would undermine the Rob3's integrity.

Scheduling

1st class: Introduction, short demonstration regarding the proposed work.

2nd class: Development of the routines required to move each joint using steps and radian inputs.

3rd class: Implementation of the F.R. related to Forward Kinematics.

4th class: Implementation of the F.R. related to Inverse Kinematics.

5th class: Finishing touches and extras.

Due date: 12/04/2015

Assessment

The assessment of each work is performed according to the following criteria:

Report (20%):

- Structure: 5%
- Theoretical part: 5%
- Description of development: 5%
- Introduction and conclusion: 5%

Implementation (80%)

- Each functional requirement that is fulfilled will be graded according to its workload and complexity level.
- There are two non-functional requirements, namely, user interface and **robot integrity**.

Annex 2 – Lab-Work Results (fill in and add at the end of your lab. work report)

Lab-Work Results nr. 1	
Course	Robotics
Year	2014/2015
Student Nº____ Student Nº____ Student Nº____	Name: Name: Name:
Delivery Date:	_____/___/_____ (must be near to the email date)

	Requirements' answers (check with "X")				
Functional Requirement	**Success (100%)**	**Almost done**	**Unable to finish with source-code (partial results)**	**Unable to fulfill it**	**Professor's review (leave it blank)**
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

Non-Func. Req.	**Success**	**Almost done**	**Unable to finish with source-code (partial results)**	**Unable to fulfill it**	**Professor's review (leave it blank)**
User interface					
Robot Integrity					