31383 Robotics
ROBOT – Hands On

(rev 1.0)

1 Objective

The objective of this exercise is to give an introduction to robot programming

When you have finished this exercise you will be able to:

- run the robot with the pendant and python scripts
- Find and use the transformation between task coordinate system and robot coordinate system
- program robot movement in Cartesian coordinates

2 Preparation

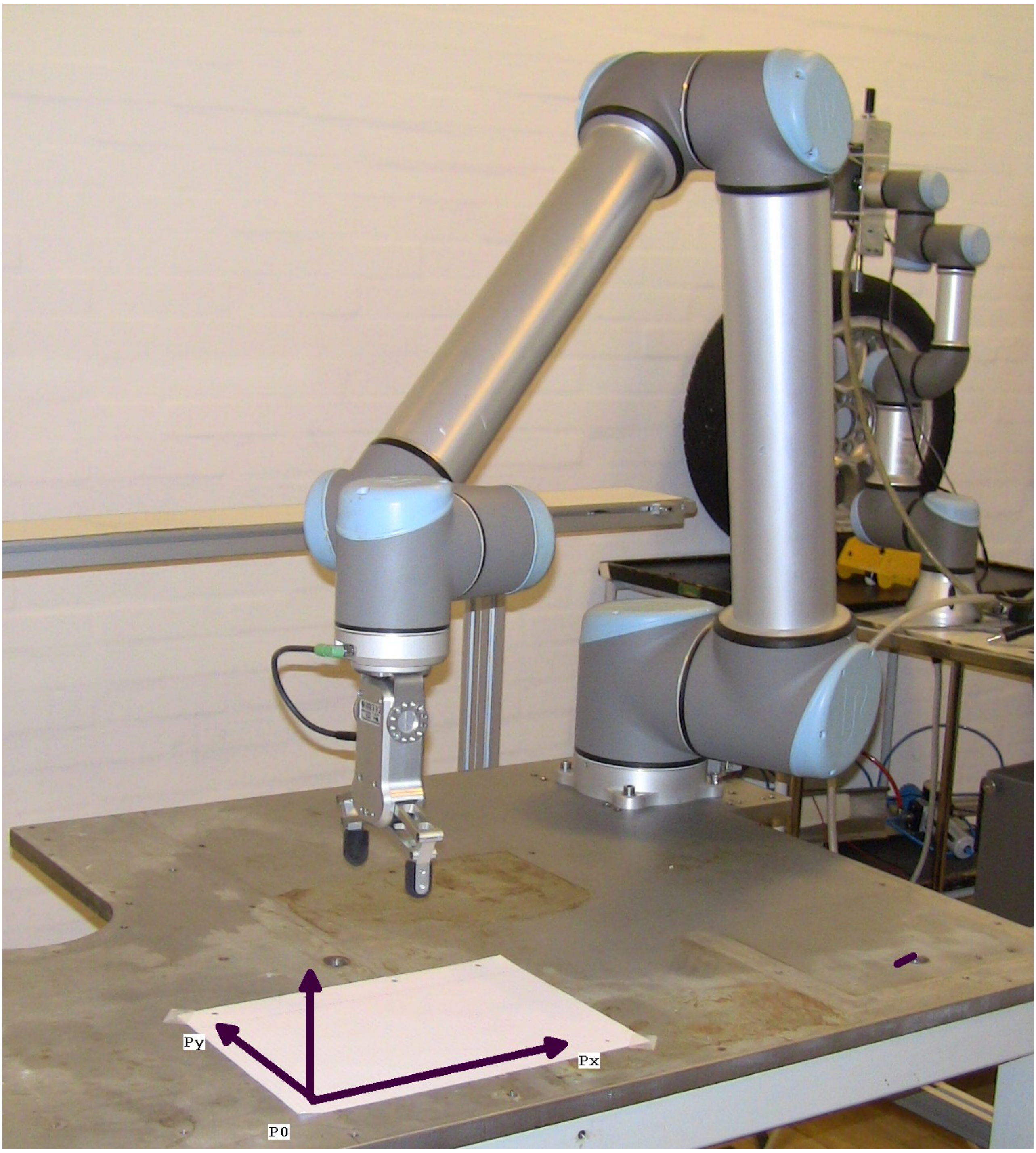
Read this manual and write the matlab script described in 3.1

3 Introduction

The figure below shows the robot with the task coordinate system defined by three points P0, Px and Py on the table. The idea of having a task coordinate system is that it is possible to choose a coordinate system that makes it easy to specify the task and that this specification is independent of the robot used for the implementation. To use the solution on a given robot one only needs to find the transformation between the task coordinate system and the robot base coordinate system.

Given that P0 is the origo of the task system and Px and Py are points on respectively the x-axis and the y-axis of the task system unit vectors defining the task system can be found:

$$v_x = (Px - P0)/|(Px - P0)|$$
$$v_y = (Py - P0)/|(Py - P0)|$$
$$v_z = v_x \times v_Y$$



The coordinates of P0, Px and Py in the robot base coordinate system can be found by moving the tool point to each of the points and read the base coordinates. The transformation from task to base is the given by:

$$P_{base} = (v_x v_y v_z) P_{task} + P0$$

3.1 Matlab script for transformation calculation

Make a matlab script that given the base coordinates for P0, Px and Py in millimeters calculates transformation above in meters. The functions **norm** and **cross** may be helpful.

Check:

```
p0=(577,-722 -21)
px=(579 -477 -23)
py=(425 -720 -23)
```

will give a rotation matrix:

$$R=\begin{pmatrix} 0.0082 & -0.9998 & -0.0130 \\ 0.9999 & 0.0132 & 0.0083 \\ -0.0082 & -0.0132 & 0.9999 \end{pmatrix}$$

4 Exercise program

Get an introduction to the robot system by the teaching assistant.

4.1 Finding the coordinate transformation

The objective of this exercise is to program the robot in the task coordinate system and then use a coordinate transformation to find the robot base coordinates which the can be used in the built in **movel** command of the robot system.

Log on the robot system computer

move the robot to the home position by issuing the command:

python home.py

To simplify the exercise we will keep the same orientation of the robot hand throughout the exercise. The wanted orientation will be obtained with the home.py program. To get a good determination of the coordinate transformation we put a small wooden stick in the robot gripper. Open the hand with the command:

python gripperopen.py

attach the wooden stick to the gripper with a piece of tape so that the stick is in the middle of the gripper and stick 10 mm out. Close the gripper with the command

python gripperclose.py

Now we are ready to find the base coordinates of P0, Px and Py. The robot is moved using the arrows on the upper left corner of the move tab on the teach pendant. It is important not the chance the orientation of the hand. Move the robot so that the wooden stick is exactly over P0. Note the base coordinates. Do the same with Px and Py. It is important that the points are found as accurately as possible to get an accurate transformation. To obtain accuracy it is recommended to lower the speed to 10% on the move tab.

When the coordinates of P0, Px and Py are found use the prepaired matlab script to calculate the base coordinates for the point with task coordinates (0.1,0.15,0.01). Move the robot to the found coordinates with the teach pendant. If the transformation is correct the wooden stick should be at (0.1, 0.15, 0.01) with an accuracy better than one mm.

4.2 Programming the robot

The robot will be programmed in python. The programs work by sending the basic robot commands over a socket as strings to the robot controller. We will only use the move function:

movel(p[x ,y ,x ,Rx, Ry, Rz], acc,vel)

x,y,z	is the tool point position in meters in the base coordinate system
Rx, Ry, Rz	is the axis angle representation of tool orientation
acc	is acceleration in m/s^2
vel	is the velocity m/s

Open the **move.py program** with gedit and enter your P0, Px and Py coordinates in the **robot.transform_init** function. Set the speed back to 100% and run the **home.py** program and then the move.py program and see that it will go to (0.1,0.1 0.005) in the task system.

The goal of the exercise is to make a program that moves three wooden blocks from given positions in the task system to a stack in a given target position. This is done in a number of steps. Base your program on a copy of move.py

Make a python function **goto(x,y,z)** based on **robot.transform(x,y,z)** and **movel**

that makes the robot go to the given task point (x,y,z). Test it with a number of positions.

In python the function definition of goto(x,y,z) is:

```
def goto(x,y,z):
    code
```

The code lines must be indented as indentation in python is used instead of parenthesis to give the structure.

Make a python function **pick_at(x,y,z)** that based on **goto(x,y,z)**

- moves the robot hand to a position 5 cm above (x, y, z)
- opens the hand
- moves to (x,y,z)
- closes the hand
- moves the robot hand to a position 5 cm above (x, y, z)

Test the program with a wooden block

Make a python function **place_at(x,y,z)** that

- moves the robot hand to a position 5 cm above (x, y, z)
- moves to (x,y,z)
- opens the hand
- moves the robot hand to a position 5 cm above (x, y, z)

Combine the function with pick_at and test with a wooden block

combine pick_at and place_at at to make a stack of three blocks.

The start positions of the block should be:

(0.05,0,-0.01), (0.10,0,-0.01), (0.15,0,-0.01)

The position of the stack should be:

(0.20, 0.15,-0.01)

Test the program.

Demonstrate your stack program to the teaching assistant.