Norwegian University of Life Sciences
Faculty of Science and Technology

VT2022

Lab report Part II & III / TEL200 – Introduction to Robotics

YuMi Application

Author(s): Mohammed Idris Omar, Liiban Hassan Osman and Ammar Abou Madiara

Abstract

This lab project is divided into two parts, Part II and III. Part II goes through the theory presented in the background section of the introduction. The Background section of Part II contains preliminary theory such as frames, poses, orientation in 3D and presents the kinematic problems in very basic form all of which serves a purpose for both parts as they are necessary to understand the programs developed in the simulation software RobotStudio. Things that are related to or applicable in RobotStudio are italicized to make it easier for distinction.

Part II revolves around creating an application for the YuMi cobot where it is supposed to interact with a modelled button in RobotStudio. The task of the robot is to push when an implemented *digital input signal* value becomes 1 and then reset the button back to its original position when it is 0. The method section shows the modelling process of the button, making it into a *smart component*, creating *Workobjects*, *targets*, *paths*, *procedures*, *emulating* via the *FlexPendant* and finally applying it onto the real robot. The results are recorded in a video, showcasing the simulation of the robot in RobotStudio performing the task and then applied on to the real YuMi cobot. Also, the most efficient way of development process and the usage of the robot arms is discussed in the practical evaluation section.

In Part III it was specified to create an own application for the YuMi cobot as part of the YuMi challenge. YuMi cobot is the legendary collaborative robot from ABB specifically created for the purpose of working together with humans. This allows for the usage of the robot where interacting with humans is necessary. Over the past two years the world has been affected by the Covid-19 pandemic and many test centers were created to test people for possibly being infected by the virus. In the test center humans are the ones doing the testing of other people, which creates the risk of spreading the virus. To prevent the spread of the disease a possible solution would be to make YuMi stationed at the test center and take test of people. This would also help economically as it would get rid of the cost for human labor

Contents

List of figures

1 Introduction

1.1 Background

Before describing the method of development for the YuMi application with the simulation software RobotStudio from ABB, some background information is necessary. The goal of this section is to develop a fundamental understanding of frames in the 3D coordinate system, rotation about the frames axes, homogenous transformation and present the kinematic problems. Understanding these concepts will help a great deal in the simulation, emulation and application process of this lab project.

1.1.1 Frames and pose

Pose is a combination of position and orientation. It describes the distance, location and where an object may be facing. Suppose there is a 3D coordinate frame $\{B\}$ placed in the reference coordinate system $\{A\}$ shown in Fig. 1.1.1a, then the frame $\{B\}$ can be described with respect to frame $\{A\}$ by the pose ${}^{A}\xi_{B}$ as shown in Fig. 1.1.1b.

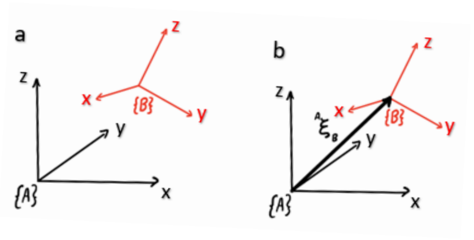


Figure 1.1.1 **a** Frame $\{B\}$ placed in reference coordinate frame $\{A\}$. **b** Pose of $\{B\}$ with respect to $\{A\}$.

This pose ${}^{A}\xi_{B} \sim {}^{A}\boldsymbol{T}_{B}$ where ${}^{A}\boldsymbol{T}_{B}$ is a homogenous transformation matrix and its elements are,

$$
{}^{A}\boldsymbol{T}_{B} = \begin{pmatrix} {}^{A}\boldsymbol{R}_{B} & {}^{A}\boldsymbol{t}_{B} \\ \boldsymbol{0}_{1\times 3} & 1 \end{pmatrix}
$$

where ${}^{A}\boldsymbol{R}_{B}$ is the rotation matrix describing the orientation and ${}^{A}\boldsymbol{t}_{B}$ is the translation matrix from $\{A\}$ to $\{B\}$ respectively. [1]

1.1.2 Orientation in 3D

Rotation of coordinate frames is key to understanding the *Workobjects* in RobotStudio which will be presented in the method section. In the Fig. 1.1.2a, there is a coordinate frame in its original position, then the same frame is rotated $\frac{\pi}{2}$ about its x-axis and another rotation of $\frac{\pi}{2}$ about the y-axis. Reversing the order of rotation as shown in Fig. 1.1.2b, gives a different result and hence the rotation order matters in 3D, and this proves the noncommutativity. [1]

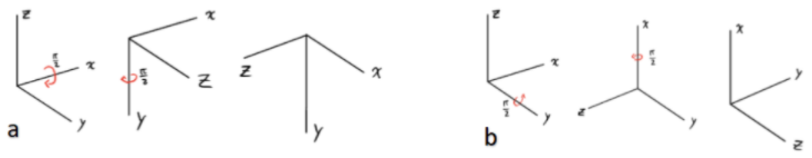


Figure 1.1.1.2 **a** Rotation of coordinate frame. **b** Rotation order of frame reversed.

1.1.3 Kinematic problems

Forward kinematics is when the joints angles of a robot is known, but the position of its end effector is unknown. Inverse kinematics is the opposite of forward kinematics. This is when the end effector pose is known but the required joint angles are unknown. Both problems can be solved by the kinematic function which is represented by κ. The forward kinematic is given by,

$$\xi_E = \kappa(\boldsymbol{q})$$

in this case the kinematic functions κ computes on the known joint configuration angles $\boldsymbol{q}$ in order get the end effector pose ξ_E. The inverse kinematics is then given by,

$$\boldsymbol{q} = \kappa^{-1}(\xi_E)$$

in this case the computation is done on the end effector pose to find the required joint configuration angles. [1]

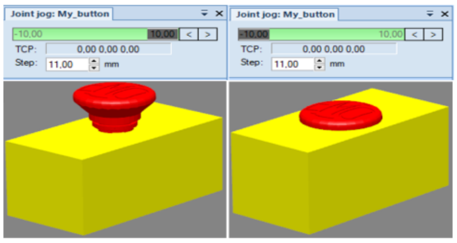
2 Method

2.1 Modeling the E-Stop button

The E-Stop button represents the device that the YuMi cobot will interact with. This section will go through the modelling of the button in the simulation software RobotStudio such that it goes down with a push and up when it is reset.

2.1.1 Create Mechanism, Type, Links and Joints

Navigating to Modeling in the menu tab of RobotStudio and choosing the *Create Mechanism* option, a window on the side will pop up. In this window there is a list called *Mechanism Type,* here it is possible to specify the type of the mechanism. This is a button which does not have any TCP, so the appropriate choice is the *Device* type. [2] For the *Links* of the device the *BaseLink* (i.e., fixed), is the yellow box and the child link is the button itself. In the *Joints* the *Joint Type* is set to *prismatic* with one joint axis as the button moves only up and down giving it one degree of freedom. [1] Its movement is set from -10 mm to +10 mm in the z-direction. After *Compile Mechanism* it is possible to *Joint jog*, the button as shown in Fig. 2.1.1.



TCP: 0,00 0,00 0,00
Step: 11,00 mm

TCP: 0,00 0,00 0,00
Step: 11,00 mm

Figure 2.1.1 To the left, the button is jogged 10 mm and to the right -10 mm in the z-direction, respectively.

2.1.2 Smart Components

Making the button into a smart component makes it possible to joint jog the button up and down with *Input signals* that are connected to *JointMover*. In the editor of the smart component the *JointMover's* can be added under *Compose* and connected to the up and down inputs under *Signals and Connections*, where they are put into the *I/O Connections*. The design diagram of the smart component in Fig. 2.1.2, shows the connection of the inputs to *JointMover's.*

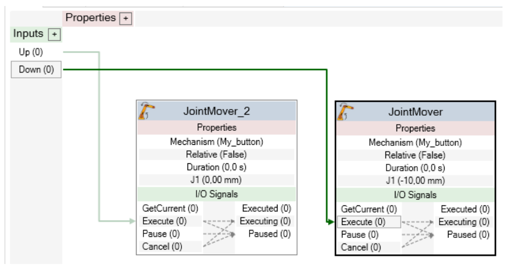


Figure 2.1.2.2 Input signals connected to *JointMover's*.

2.2 Simulation

Program for the real YuMi cobot is developed by simulating the robot performing the task in RobotStudio. This section is dedicated to show the usage of *Workobjects*, defining *Targets*, and creating *Paths* and *Synchronizing* the procedures to the main program aka the *Rapid* code.

2.2.1 Defining Workobjects, Targets and Configuration

Default *Workobject* is the *wobj0* which coincides with the world coordinate system of the robot. The *Target Home* was defined with respect to this frame. *Workobject_1* is the frame defined to describe the position of the button. *AboveButton*, *Push* and *Rotate* are all *Targets* defined with respect to the frame of the button. These *Targets* are defined and stored as coordinates in the *Workobjects.* [2] Not all targets were reachable with default *configuration* therefore, to make the end effector of the robot have the same position at the *Target* an alternative *configuration* for the joint angles had to be chosen. The left and right arm of the YuMi cobot has 7 joints which makes it a 7 degrees of freedom robot which can adopt to any x, y and z-coordinate within its working space. [3]

2.2.2 Paths, Procedures, Synchronizing to Rapid and digital inputs

Paths PushButton and *ResetButton* are procedures which contain the *Targets* that the robot should reach for. These *Paths* are then moved in the *main (entry point)* where the procedures are executed. Clicking on *Synchronize* in RobotStudio transfers the *Paths* and *Targets* to the *Rapid* code. This allows for the modification of the program through the code. *di_EmmergencySituation* was the signal used to perform the action of pushing the button down when signal value on the icon in the *I/O Simulator* showed 1 and reset the button when the value was 0.

2.3 Emulation

To get more realistic results of the YuMi performing the task the *Virtual Flexpendant* was used. Through the *FlexPendant* many errors in the program were detected, this led to a more efficient way of testing before using the *Rapid* code on the real YuMi cobot.

2.4 Application and development process

When reaching the stage of application, a clear form for structure of the development had been implemented. A test-driven development process was intact, where if no error occurred in the simulation, then it would go on to be emulated via the *FlexPendant* and if there were no errors in the emulation then it would be applied on to the real YuMi cobot. In either case if the process failed then the code would be edited.

3 Results

The robot was able to successfully complete the task by pushing the button down with the signal from the *di_EmergencySituation*. In the video delivered in part II of this lab project, the simulation is first shown and thereafter the application of the program on the real YuMi cobot. The *Rapid* code can be found in [4].

4 Discussion

4.1 Practical evaluation

When reaching the application part of this project, it was clear that the development process up until then was flawed. A test-driven development process from the start would have saved a lot of time. Most of the errors that did occur during the application could have been detected with emulating the program through the *FlexPendant*. Creating *Workobjects* and *Targets* that are related to them, proved out to be more efficient then jogging the robot and teaching it the target. Initially it was planned for the robot to use both of its arms to perform the task, but this did not happen due to lack of time and as the robot's right arm gripper was broken.

Part III – YuMi Challenge: A corona test station

5 Introduction

5.1 Background

When the pandemic started the amount of human contact had to be drastically reduced to not spread the Covid-19 virus. Digitalization was a huge part of the solution for reducing the amount of contact between people. Most education system had adapted to being online and the same applied for many jobs where this was possible. [5] Not all jobs can be digitalized, for some jobs direct contact with humans is necessary.

During the pandemic, many Covid-19 test centers where established, so that people could either drop by or reserve an appointment to get tested. The people performing the tests were someone who worked in the health-sector or were trained to be able to take the test. Regardless these were people performing a task relatively simple enough to be automated by robots. A safe robot to carry out the task of taking the test would be the YuMi cobot from ABB. This robot is meant for collaborative purposes, as it is built with a lot of safety features and it is not very strong, which reduces the risk of it hurting anyone.

5.1.1 Motivation

In October 2020, an uncontrolled outbreak of the virus occurred at Hammerfest hospital among employees. Two of their employees were infected with the Covid-19 virus and all the other employees had to quarantine. Some departments were shutdown whereas some had to reduce their capacity. [6] This shows how easily the virus could affect someone, spread the disease and lead to consequences such as quarantining, departments getting shut down etc.

Developing an application for the YuMi cobot in order to take Covid-19 test of possibly infected patients would reduce such outbreaks, it would also mean no quarantining and there would be no change in the work capacity. It is estimated that the Norwegian government used 650 million NOK on testing centers at the boarders of the country alone. Having a robot stationed at the test center would get rid of the cost for human labor. [7]

6 Method

6.1 Hierarchical structure

The Coordinate systems or frames are related to each other hierarchically. In the YuMi corona test station there are a total of 5 *Workobjects*, each of these compose of two frames namely the *User frame* and the *Object frame*, where the latter is a child to the former. [8] These *Workobjects* are placed in the *World* coordinate frame where their origins can be described relative to the *World* origin as illustrated in Fig. 6.1.

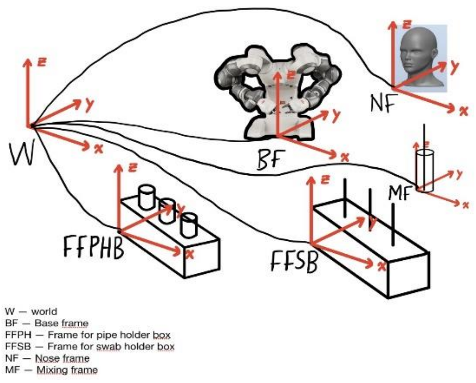


Figure 6.1 *Workobjects* placed in the *World*

Targets can then be conveniently defined with respect to the *Object frame* of *User frame*. This is useful as the objects position in the real world may differ from what is programmed offline, so instead of redefining all targets, it is possible to simply adjust the objects in reference to the objects position in the real world. The *NoseFrame* is something that must be adjusted often if the tests are to be performed on different people as height of people varies.

6.2 Velocity and zonedata

The velocity of the robots end effector which is its servo was set to 200 mm/s, since anything other than this speed in the simulation will look to fast. Based on previous application in Part II, this speed would have been to slow on the real robot, but as this is purely simulation based this fact is neglected.

There are a lot of stop points rather than fly-by points in this program i.e., the robots' external axes must reach the position and stand still before the next instruction of the program is executed. The reason for this is because the robot must be accurate as it deals with *Target points* such as picking up a thin swab and putting it inside the nostril etc. To define these stop points, the *zonedata* is set to *fine*. [9] In Fig 6.2 a tool with a center point travel along a path created by three points, to the left all points are stop points and to the right the point p2 is a fly by point.

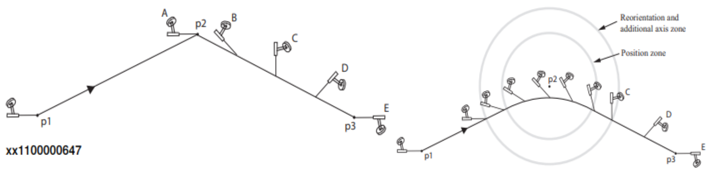


Figure 6.2 Example of stop and fly-by points. (Screenshot taken from [8])

6.3 Robot task and signals

Three *Digital input signals* were created for each test. At signal value 1 the robot is to take a test of the human with a swab and then put it in the pipe holder. An extra signal *di_StopMixing0* was added for the mixing process, to stop it and return the swab and the pipe holder to their original positions. Returning the swab and pipe holder after performing the test is of course not realistic, but this is done for simulation purposes only.

7 Results

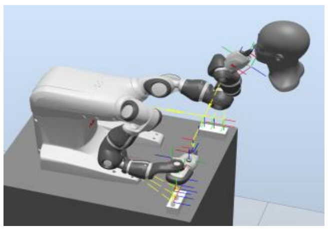


Figure 7.1 Overview screenshot of YuMi taking a covid-19 test in RobotStudio.

In the simulation video [10], YuMi were able to take the test of the human head placed a certain height above in the air. This head was then adjusted to another height to represent the differences in height of people, and the robot were still able find the *path* to the *target* due to the *NoseFrame* being attached to the head. The left hand was used for picking up the swab and then taking the test, while the right hand picked up the pipe holder and held it at center ready for mixing after the test had been performed. *Rapid code* for both arms can be found in [11].

8 Discussion

8.1 Practical evaluation

The YuMi cobot can be trusted to perform the task of taking tests on real people, however there is a lot more work needed to be done before it can be used in a real test station unsupervised. Human behavior can be unpredictable, for example if the robot were to put the swab inside the nose, and the human might react by wanting to sneeze, then this could propose a danger to the human because the robot will not take out the swab at the right time.

A frame has been attached to the nose of the head to make it easier for repositioning if a person that is taller or shorter would be tested, but this is also a tedious task to do as it must be manually adjusted all the time. A more automated approach where the robot could see the nose would be better, however in order to use the integrated vision on the YuMi cobot it must be activated in real life, this was not possible as the robot was no longer in the possession of the university.

8.2 Further work

YuMi is already very safe as it is, the danger of a person sneezing while being tested could be solved by modifying the swab. If the swab were to be created softer and more bendable then it would not propose that much of a threat. To solve the problem of manually adjusting the position of the head, a camera could be mounted on top of the robot to detect the person. With a camera it is also possible to learn the robot recognizing when a person might be sneezing. A library such as OpenCV could be used to develop the algorithm of recognizing when someone is about to sneeze and then program the robot to pull out the swab if that were to be the case.

References

[1] P. Corke, Robotics Vision and Control FUNDEMENTAL ALGORITHMS IN MATLAB®, Springer, 2017.

[2] Operating manual RobotStudio, ABB, 2021.

[3] P. Corke, "https://robotacademy.net.au," © QUT Robot Academy, 10 April 2017. [Online]. Available: https://robotacademy.net.au/masterclass/robotic-arms-and-forward-kinematics/?lesson=262.

[4] M. I. Omar, "github.com," [Online]. Available: https://github.com/Idris802/TEL200/blob/main/Part2/CodeForPushAndReset.mod.

[5] E. H. Horgen, "ssb.no," 24 february 2021. [Online]. Available: https://www.ssb.no/arbeid-og-lonn/artikler-og-publikasjoner/209-000-ansatte-med-avtale-om-hjemmekontor.

[6] A. D. Emma Dalgård, "dagensmedisin.no," 29 June 2021. [Online]. Available: https://www.dagensmedisin.no/artikler/2021/06/29/viktig-lardom-etter-ukontrollert-smittespredning-i-lokalsykehus/.

[7] L. T. J. B. S. H. G. Sophie Lorch-Falch, "nrk.no," 7 October 2020. [Online]. Available: https://www.nrk.no/norge/slik-vil-regjeringen-bruke-koronamilliardene-1.15190833.

[8] Technical refrence manual Rapid overview, ABB.

[9] Technical reference manual RAPID Instructions, Functions and Data types, ABB.

[10] "youtube.com," [Online]. Available: https://www.youtube.com/watch?v=ZIspqZ6Xfl8.

[11] M. I. Omar, "github.com," [Online]. Available: https://github.com/Idris802/TEL200/tree/main/Part3.