

Fontys Hogescholen

Programming and Visualizing Industrial Robots with a Web-Based GUI

Zirui Song (2652528)

# Global content of the report

In your university report you describe the following topics:

Definition of the initial problem (the reason for your assignment) and the assignment aim. Including the scope and the domain of the problem.

The output of your assignment: what are the deliverables and non deliverables?

HOW THE PROBLEMS ARE TACKLED AND SOLVED.

* + Information gaps you encountered.
  + The research strategies you used for each problem (Field, Library, Workshop, Lab or Showroom).

Methodologies used for your development process, for example:

* + Traditional System Development Life Cycle (SDLC)
  + Agile modeling and prototyping,
  + Object-oriented systems analysis and design
  + Service Oriented design and architecture

Information gathering methods and data collection procedures you used for your research strategy “Field”:

* + Interactive methods (for example: interviewing)
  + Unobtrusive methods (observing; analyzing documents, screens, web sites..)

Techniques for analyzing and structuring systems you used:

* + Techniques for process modeling (for example: DFD’s)
  + Techniques for data modeling (for example: ERD’s)
  + Object-oriented systems analysis techniques

Tools you used, for example

* + Case tools (Visible Analyst, System Architect, Rational Software Architect, Visual Paradigm)
  + Modeling tools
  + Documentation Tools
  + Engineering tools
  + Code generators, Screen generators, Report generators

Integrated Development Environments (IDE) used

* + - Eclipse
    - JDeveloper
    - Oracle designer
    - Visual Studio

Programming languages and frameworks used

The final results of your work (architecture design, database design, screenshots GUI, secure login procedures, user interface test reports, developed new algorithms, etc.)

Conclusions, in which you compare your end result with the initial assignment aim

Recommendations for further research or assignments

The personal reflection and evaluation

* + Reflection/Impact:
    - What did you learn about yourself?
    - In what areas did you experience the most professional growth?
    - What insights have you gained from your assignment
    - Based on your internship experience, what skills would you like to develop in preparation for your career?
    - How has the internship influenced your career goals? Please explain.
  + Evaluation:
    - Did the internship experience meet your personal expectations?
    - How would you assess your performance at the company?
    - Would you recommend this site to future graduate interns?

It should be possible for a layman to read your report. The reader must be able to follow the main line without understanding the technical details. You can easily verify the readability of your report by asking as many ICT people as possible to read it.

INTERNSHIP REPORT

FONTYS UNIVERSITY OF APPLIED SCIENCES

HBO-ICT: English Stream

|  |  |
| --- | --- |
| Data student: | |
| Family name , initials: | Z Song |
| Student number: | 2652528 |
| assignment period: | 30-08-2017 to 05-02-2018 |
| Data company: | |
| Name: | Fontys Hogeschool Engineering |
| Department: | Mechanic & Robotics Laboratory |
| Address: | R1 0.204-0.205 |
| Company tutor: | |
| Family name, initials: | PC Negrete Rubio |
| Position: |  |
| University tutor: | |
| Family name , initials: | E Thaqi |
| Final report: | |
| Title: | Programming and visualizing Industrial robots with a web based GUI |
| Date: | 17-01-2018 |

Approved and signed by the company tutor:

Date: 17-01-2018

Signature:

# Foreword/Preface

The foreword can be written in the ‘I’ shape. The foreword mentions the framework in which the report is written, and in which company (name and type) the assignment is executed. It gives an indication about the nature of the contract. When the assignment is done by two students (a duo-stage), it says who did what and who has written which chapters (exactly, with name and title). This is the only place where the division is discussed; so you should not put the name of the writer above or below each individual chapter.

In the preface you can thank the company tutor and university tutor for their guidance.

The preface chapter has no pagenumber, no sections and no figures.

Contents

[Summary 1](#_Toc503802774)

[Glossary 1](#_Toc503802775)

[1. Introduction 2](#_Toc503802776)

[2. About the company 2](#_Toc503802777)

[3. Project overview 2](#_Toc503802778)

[3.1. Facts 3](#_Toc503802779)

[3.2. Target of Project 6](#_Toc503802780)

[3.3. Analysis to the project 6](#_Toc503802781)

[4. Project Progress 6](#_Toc503802782)

[4.1. Prepositive Prototypes 7](#_Toc503802783)

[4.2. Development of Final Product 8](#_Toc503802784)

[5. Conclusion and Outlook 10](#_Toc503802785)

[5.1. Project Summary 10](#_Toc503802786)

[5.2. Further Functionalities 10](#_Toc503802787)

[Evaluation 12](#_Toc503802788)

[References 12](#_Toc503802789)

[Attachment A: Project Plan (Excerpts) 12](#_Toc503802790)

[Attachment B: Links 13](#_Toc503802791)

# Summary

The informative summary is maximum one page long, and summarizes the entire report. You can only write the summary after you have completed the report. (See also below at 1. Introduction, NB 1.) It describes at what company and type of company you've performed the assignment; the initial situation, what problems existed, and therefore which assignment was given. Then you describe and explain which approach you have followed and what the results were, and your conclusions, recommendations and evaluation.

In the summary are no references to other parts of the report; don't go too much on details and make sure the summary is *independent* of the rest of the report. In addition, the summary must be understandable to a layman.

Just like the foreword the summary has no sections, no figures or enumerations. You start numbering the pages here.

Usually the past tense is used for writing this chapter.

# Glossary

The glossary lists and defines your concepts, terms, symbols, and abbreviations, in alphabetical order. The first time you use an abbreviation in the text, please follow these steps: you write everything in full, then the abbreviation in parentheses. For all subsequent calls you use only the abbreviation.

# Introduction

This project is to build up a control-box web application for an industrial robot with JavaScript, which is to be run in any devices and compatible with any industrial robots.

Universal Robots UR5 is equipped in the labortary so it is used for this project.

The introduction should give the reader the first, global information aboutthe assignment; the reader is prepared for reading the rest. There must therefore be at least information about the company, the problems, the assignment and the relevance of the assignment, and already a short indication of the followed strategy. The reader should be encouraged to read the report; so start with a catchy opening sentence, for example a compelling question, an equation, or a spectacular story (anecdote). If you start with a quotation, you must ensure that you know something of the background of that quotation.

Chapter 2 provides information about the company.

Chapter 3 gives all the details about the project and its related things.

Chapter 4 describes how the project is done.

Chapter 5 concludes this project.

# About the company

The Fontys mechatronics laboratory main goal is to recreate and solve industrial challenges while contributing to Fontys students education. The challenges that students face in the lab, are developed following stricted use case scenarios that company partners face in a daily basis, because of that, students get to work in the most innovative topics, giving to our company partners a proof of concept before implementing new technologies.

Engineering students in semester 7 will attend the automation & robotics class and do their practical in the robotics laboratory. The practice content is shifting from the production of dedecated production lines with a fixed layout and processing order for a product or product family, towards intelligent, more or less autonomous, processing stations that can be combined in an ad-hoc way depending on the desired product.



Figure 1: Students working on their projects with UR5 robot

# Project overview

In this chapter you give all the details about your assignment. The reader at the end knows everything about the following areas:

* What are the constraints for your assignment?
* How are you going to phase you assignment?

## Facts

### Universal Robots UR5

Universal Robots is a Danish robotics company founded in 2005. It provides UR3, UR5 and UR10 series of industrial robots to meet worldwide industrial automation requirements in different levels. UR5 is a typical 6-axis industrial robot. The maximum payload of UR5 is 5 kg and its work radius is 850 mm. (Universal Robots A/S, 2009)

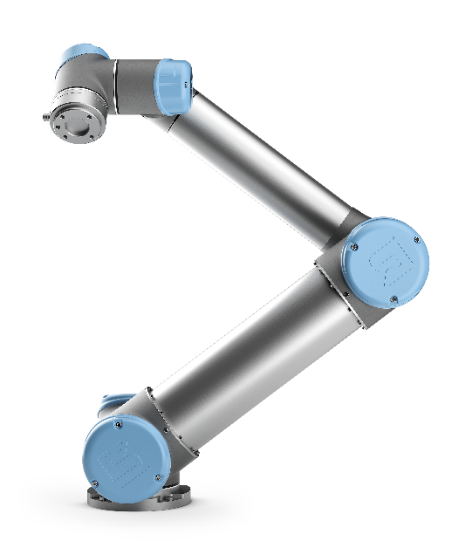


Figure 2: Universal Robots UR5

Fontys Robotics Laboratory equips two UR5 robots, one of which is used for this project.

### Robot Operating System

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms. (About ROS, n.d.)

There are many ways to communicate with ROS:

* Call services,
* Subscribe to or publish on topics, and
* Get or set parameters.

### URDF and TF

The Unified Robot Description Format (URDF) is an XML specification to describe a robot. The specification covers kinematic and dynamic description of the robot, visual representation of the robot and collision model of the robot. The description of a robot consists of a set of link elements, and a set of joint elements connecting the links together. (XML Robot Description Format (URDF))

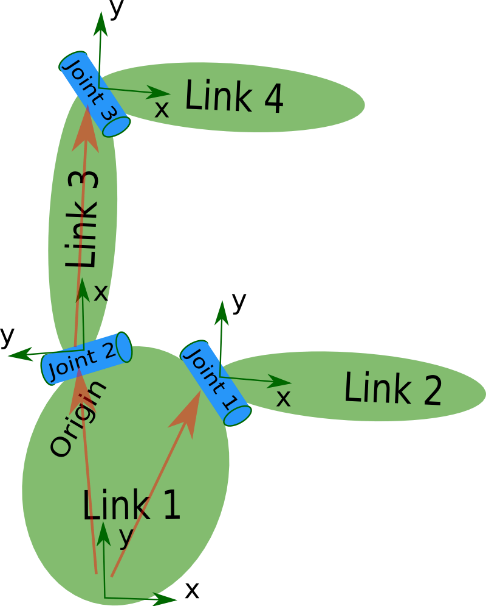


Figure 3: The Unified Robot Description Format (URDF)

So a typical robot description looks something like this:

1. <robot name="pr2">
2. <link> ... </link>
3. <link> ... </link>
4. <link> ... </link>
5. <joint> .... </joint>
6. <joint> .... </joint>
7. <joint> .... </joint>
8. </robot>

The tf library was designed to provide a standard way to keep track of coordinate frames and transform data within an entire system such that individual component users can be confident that the data is in the coordinate frame that they want without requiring knowledge of all the coordinate frames in the system. During early development of the Robot Operating System (ROS), keeping track of coordinate frames was identified as a common pain point for developers. (Foote, 2013)

### MoveIt!

MoveIt! is state of the art software for mobile manipulation, incorporating the latest advances in motion planning, manipulation, 3D perception, kinematics, control and navigation. It provides an easy-to-use platform for developing advanced robotics applications, evaluating new robot designs and building integrated robotics products for industrial, commercial, R&D and other domains. MoveIt! is the most widely used open-source software for manipulation and has been used on over 65 robots.

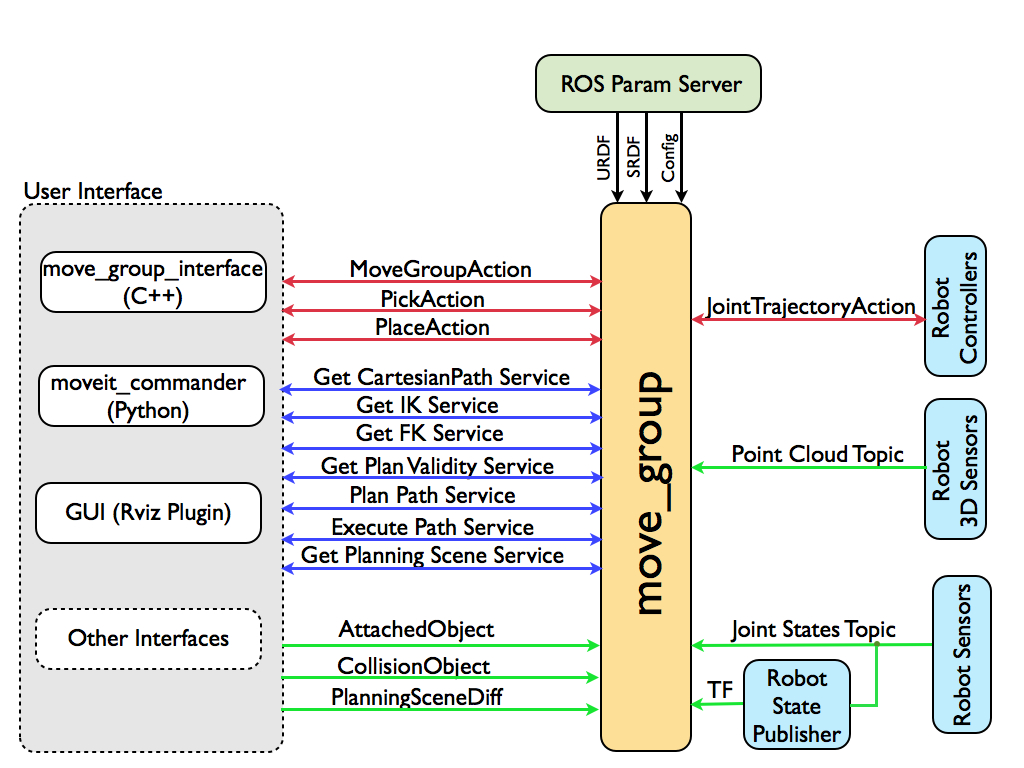


Figure 4: The system architecture of MoveIt!

Figure 4 shows the high-level system architecture for the primary node provided by MoveIt! called move\_group. This node serves as an integrator: pulling all the individual components together to provide a set of ROS actions and services for users to use.

move\_group is a ROS node. It uses the ROS param server to get the URDF, SRDF and MoveIt! configuration. move\_group talks to the robot through ROS topics and actions. It communicates with the robot to get current state information (positions of the joints, etc.), to get point clouds and other sensor data from the robot sensors and to talk to the controllers on the robot. (Sucan & Chitta, n.d.)

### RobotWebTools (including roslibjs and ros3djs)

RobotWebTools is a collection of open-source modules and tools for building web-based robot apps. Their tools allow web applications to interface with a variety of robots running middleware like the popular Robot Operating System (ROS) using the latest in HTML5 and JavaScript. roslibjs and ros3djs are used in this project with the rosbridge on server-side.

rosbridge communicates ROS data messages contained in the JavaScript Object Notation (JSON) for straightforward marshalling and demarshalling. Through its use of WebSockets (a protocol built on top of HTTP), rosbridge can be readily used with modern web browsers without the need for installation. This fact combined with its portability and pervasive use makes the web browser an ideal platform for human-robot interaction. (Toris, et al., 2015)

roslibjs uses WebSockets to connect with rosbridge and provides publishing, subscribing, service calls, actionlib, TF, URDF parsing, and other essential ROS functionality. Functionalities of connection, publishing, subscribing, service calls and parameter updating are used in this project.

ros3djs is the standard JavaScript 3D visualization manager for ROS. It is build ontop of roslibjs and utilizes the power of three.js. Many standard ROS features like interactive markers, URDFs, and maps are included as part of this library.

### Vue.js

Vue (pronounced /vjuː/, like view) is a progressive framework for building user interfaces. Unlike other monolithic frameworks, Vue is designed from the ground up to be incrementally adoptable. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. On the other hand, Vue is also perfectly capable of powering sophisticated Single-Page Applications when used in combination with modern tooling and supporting libraries. (Introduction)

## Target of Project

There are many industrial robots manufactures in our planet. Each brand of industrial robots has its pattern of controlling and programming. This makes trouble for some enterprises that need purchase industrial robots from different manufactures because they must recruit more employees to use and maintain them. So, it is good to make a unified graphical user interface that compatible with all brands of industrial robots. It is also possible to add another robot into the system. And the best thing is, you will be able to control your industrial robots at anywhere (via secured local network of your enterprise) with any device (PCs, laptops, tablets, smartphones, etc.).

The project is aimed to research, design and develop a web based graphical user interface (GUI) that lets the users communicate, program and visualize Industrial robots from different brands that are connected to it. The web based GUI is developed using Javascript, HTML5, CSS and roslibjs for the front-end and it must connect to rosbridge as the server running in ROS (Robot operating system).

Industrial robots have its own GUI to let users program and interact with the robots. Each brand of Industrial robots has its own GUI and they can be very different.

The assignment requires the design and development of a GUI that lets any possible user easily program and interact with any industrial robot connected through rosbridge in ROS to the GUI.

### Goals and Objectives

* Design following Human-computer interaction (HCI) patterns for industrial robots and that  
  communicates to the ROS server using roslibjs.
* Let the user visualize in 3D the movement that the industrial robot will perform.
* Make sure the app can run efficiently in different modern browsers and devices (laptops,  
  tablets and phones).
* Document the product well and make it easily maintainable and extendable.
* Follow the Agile method and meet in a weekly basis with trainee’s company tutor to inform  
  any new challenge and the state of the project.

## Analysis to the project

### Project Architecture

The system where the application is inside can be splitted into 3 parts: the web application itself, ROS and the robot. ROS plays a middleman role in this system. ROS connects to the robot, read the joints and publish to the browser. The browser also sends the commands to ROS and ROS talks to robot about that.

In this system, rosbridge provides an interface to the browser in ROS. The webapp uses roslibjs to connect to ROS and ros3djs for visualization. ROS connects to the robot with UR Driver and carries out motion planning by MoveIt!. The webapp part with its connection to ROS is to be done in this project.

Figure 5: The system map of the application

### Research Methodology: DOT Framework

In the project, the DOT framework is applied during research phase. The procedure of each research follows Rigor cycle. Just like below:

1. Follow online tutorials and documents (*Library*, *Lab*)
2. Try to build prototype, then the final product (*Workshop*)
3. Show the result on request (*Showroom*)

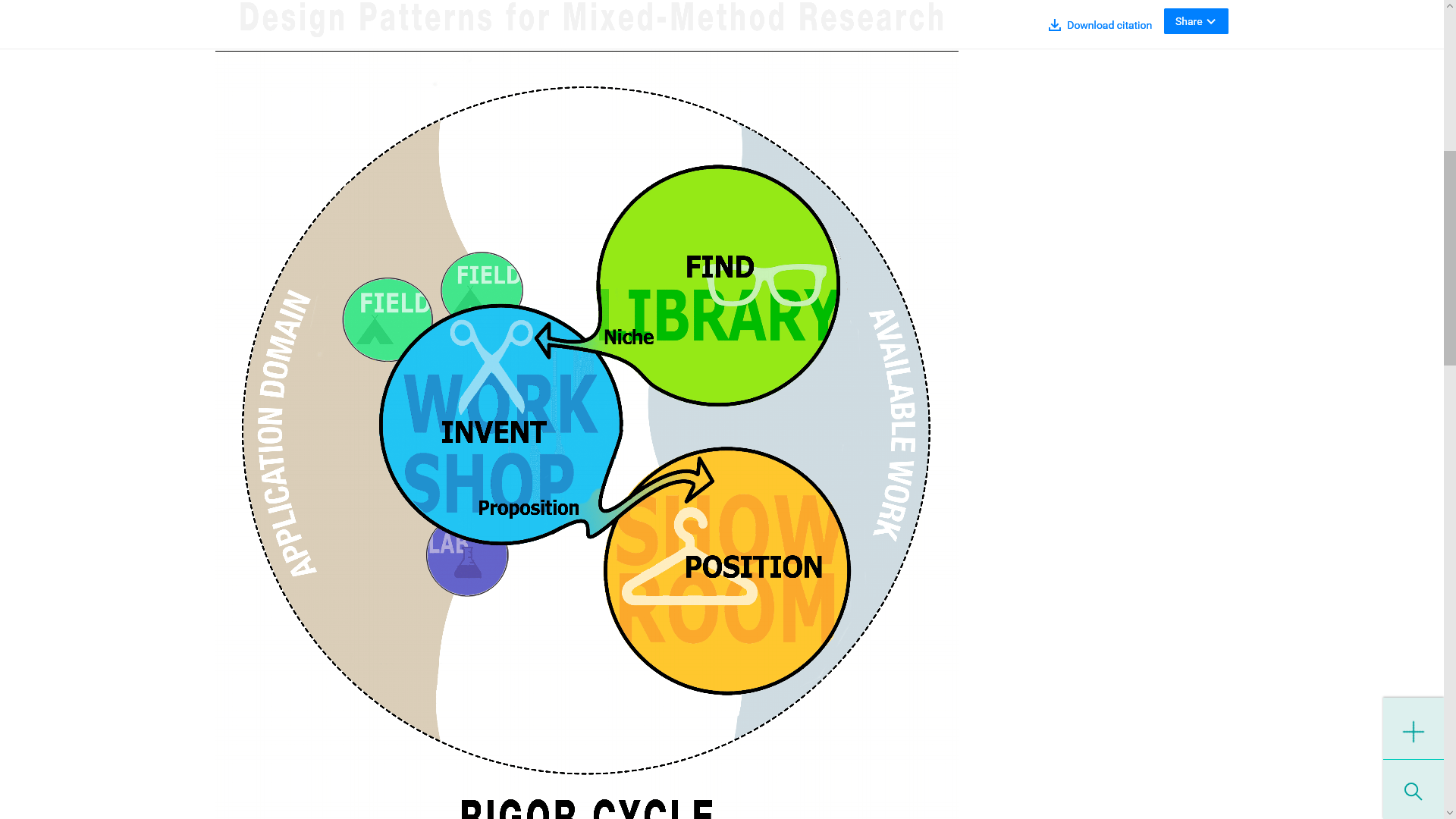


Figure 6: Rigor cycle of DOT framework

# Project Progress

In this chapter you describe **what** you've done and **how**, and especially **why**. You describe what systematic way you have followed from the assignment to the end product(s). Usually this chapter is called Results, but you are free to choose your own title.

You used a lot of research strategies (e.g. Field, Library) to get to the end result. Explain the techniques you used to execute your strategies. For example describe surveys you conducted, experiments you carried out, comparisons of possible techniques, tools and solutions, used literature found on the Internet or in books.

The progress of this project was splitted into two parts: prototyping and final product developing.

## Prepositive Prototypes

Before the final project was done, three prototypes were made for preparation as following:

* turtlesim on the web
* URDF online viewer
* MoveIt! web-based planner

These prototypes helped to master the necessary technologies that was to be used in the development of final product.

### Prototype 0: turtlesim on the Web

turtlesim is an ROS application that is used in ROS official tutorials. This application helps beginners know the basic knowledge of ROS. This prototype is aim to get a brief overview of ROS and how to communicate between browser and ROS with connection of rosbridge and the driver roslibjs. It simulates the interface of turtlesim and provides some functions by communicate to specific topics and services.

turtlesim has a 500 by 500 background color-changeable field and a turtle inside. The turtle has X, Y, Theta properties for its position in the field, and when it moves, it leaves its trace in the field. The brush width and color can also be changed. To move the turtle, linear and angular velocity values are to be sent for the movement of turtle, which will drive turtle to move for 1 second.

This prototype reproduces the field and the turtle in it, which is built in such structure:

* div#canvas: main field with background color set by turtlesim
  + canvas#drawing: records pen track of turtle
  + div#turtle: sets the position and rotation, contains the image of turtle
    - img#turtleImage: the turtle that can be seen in the webpage

The background of the field is defined by getting the parameters background\_b for blue channel, background\_g for green channel and background\_r for red channel. By subscribing turtle1/pose topic the display refreshes to show the state of turtlesim in real-time.

The control panel provides the functionalities of moving 1 step, moving constantly and stop, changing pen color including turning pen on or off and change background color. These functionailties are achieved in different ways:

* **Move**: Publish linear and angular velocity to the topic /turtleX/cmd\_vel;
* **Change pen color**: Call the service turtleX/set\_pen with pen color, width and on or off;
* **Change background color**: Set parameters background\_b, background\_g and background\_r, then call service clear to make changes effective.

roslibjs is introduced to let the prototype communicate with ROS. It is used to connect the prototype to ROS, subscribe the topics of turtle’s position, publish the topic of move, call the service of clearing field or setting the pen properties, and get or set background color. The prototype also uses setInterval() to achieve letting the turtle move constantly by publishing the same topic per every second and stops it by using clearInterval() when user clicks “Stop” button.

This prototype was done in the first month of the project period. By working on it, we had a general understanding of ROS and how to use roslibjs to communicate between ROS and the browser, which is helpful for working on further things.

### Prototype 1: URDF Online Viewer

This prototype is an application of usage of ros3djs. It presents the status of connected robot by showing its 3D-model in real-time in the browser.

ros3djs provides a 3D visual of robot in collaboration with THREE.js.

send a http get request for iframe

### Prototype 2: MoveIt! Web-Based Planner

MoveIt! is a very popular motion planning tool in ROS, which is also used in this project. This prototype lets the browser contact MoveIt! to make movement to the robot.

During the research, a similar webapp that already exists was found for this project. However, this webapp could not be used at first. This webapp is called rwt\_moveit, which is created by Tokyo Opensource Robotics Kyokai Assosiation, and based on Denso VS060 industrial robot. We contacted them for the solution of the problem and finally found the way to solve it.

This application has a webpage and some nodes written in Python. The nodes

...

## Development of Final Product

The final product was expanded from the prototype 2 with a lot of changes applied on it. The major changes were rebuilding user interfaces, login page, simple programming based on waypoints and response of robot's abnormal stop.

### Rebuild User Interface

The application is designed for running on any device such as PC, laptop, tablet, phone, etc. So the user interface should be fit within all devices. However, the UI for prototype 2 couldn't meet this requirement. So we rebuilt the UI with Bootstrap.

Bootstrap provides a responsive gridding system for webpages to make the website fit with any device.

1. <div class="row">
2. <div class="col-xs-12 col-sm-6 col-md-9" id="urdf-container">
3. ..
4. </div>
5. <div class="col-xs-12 col-sm-6 col-md-3">
6. ..
7. </div>
8. </div>

The code above means:

* When the width of window is less than 575 px, the main viewer fills the window in width, setting the controls below the main viewer. On the other hand, the height of main viewer is set to 50% of window height (this is done with JavaScript).
* When When the width of window is more than 576 px but less than 767 px, the main viewer at left takes up 50% of window width, remaining half width for controls at right side. This ensures the right side is wide enough to fit the controls inside when using a device with small screen.
* When the width of window is more than 768 px, the main viewer at left takes up 75% of window width to maximize the main viewer.

Bootstrap also contains different useful controls and a layout system for webpages. These functions made the web page better-looking but less workload.

To make a better viewer, we also changed the main viewer to light theme by setting the color of background and the grid:

1. var viewer = new ROS3D.Viewer({
2. divID : 'urdf',
3. width : width,
4. height : height,
5. antialias : true,
6. background: '#cccccc'
7. });
8. viewer.addObject(new ROS3D.Grid({color: '#666666'}));

### Login (prototype)

A login page is introduced into this application. It prevents unauthorized personnel from improperly using this application and any personal injury or property damage caused by that.

The code of this part is under Vue.js pattern. That means the inputs of username and password are linked with the related variables and the latter updates when the inputs change. When user clicks “Login and connect” button, It checks if the user exists and the password is correct. Currently, as a prototype, the user data is hard-coded into the script file.

### Adding, going to, Saving and Loading of Waypoints

When the goal state of MoveIt! planning executed successfully, the “Add Waypoints” button turns enabled and user can click to save the waypoint into the list (array). The array preserves all waypoints user saved. The waypoints will show as buttons with the name of each waypoint. By clicking it, the robot will move to the waypoint immediately. User can also let the robot moves via all waypoints in order by clicking “Run All Waypoints”.

The information of joints is extracted from the message of /joint\_states with each waypoint’s name, its joints’ names and positions. This is done by set the joint values of sliders as of waypoint, make a MoveIt! planning and execute it in one step. which is done by planning and executing all the waypoints one by one. In this procedure, the completion of robot's each step is detected by subscribing from topic /follow\_joint\_trajectory/result or /arm\_controller/follow\_joint\_trajectory/result for virtual environment, following with actions of planning and executing next waypoint in the list or step out after last waypoint is executed.

The list of waypoints can be saved as a JSON file by clicking “Save Waypoints” button. The array of waypoints is converted to JSON and taken over by FileSaver.js for saving it to external file. The loading of file is splitted into two steps. The first step is trigger click event of a hidden file input to call the open file dialog when user clicks “Load Waypoints” button. The second step happens when a file is selected, which means the file input has changed. FileLoader in JavaScript is used for load the JSON file. After that, the JSON will be converted to array and overwrite the current one.

### Receiving E-Stops and Protective Stops

An emergency button is equipped on UR5 robot. When something emergency happens, the operator can press this button to force the robot stop and lock it. The protective stop is triggered when the robot unexceptedly deviates from its establised track in robot moving to help protective the robot from being damaged. These measures protect the safety of the robot, the operators and the production line.

The UR driver can also detect emergency stops and protective stops. When an emengency stop or a protective stop occurs, ROS will receive one of these following messages:

1. [ERROR][T] Emergency stop pressed!
2. [ERROR][T] Robot is protective stopped!

According to the result of monitoring the topic /rosout\_agg, these messages were found from a node called driver. With this, the topic /rosout\_agg is subscribed; However, only the messages which are typed as “Error” and come from driver will be responsed. The messages will be popped as a message box to tell the user the robot is emergency or protective stopped, stop the running waypoints and prevent user from adding new waypoint here.

# Conclusion and Outlook

## Project Summary

In your conclusion chapter you can summarise your findings and explain or interpret what the results mean. You look back at the initial problem and validate if your solution(s) solved the problem. If your results need a lot of interpretation, it is better to create a chapter “Discussion” before this chapter.

If you have recommendations for further research on the topic you investigated, you can state this here.

The reader should be able to understand this chapter even when he or she only read the introduction chapter! The reader who has read the whole report, should encounter no new information in this last chapter, indeed: he must be able to predict its content!

This last chapter has no figures or enumerations. The maximum length is one page. A combination of tenses can be used to highlight past research and future directions.

## Further Functionalities

Because of time limit of the internship, some problems and functions were not solved in the project. The following aspects can be the goals of further project(s) based on this product.

### Move robot by using arrow buttons (with kinematics.js)

Buttons for move and rotate are placed into the GUI, but functions are not implemented. To make the robot move by arrows possible, the X, Y, Z values of the position and rotation of the robot end effector should be calculated. kinematics.js is a utility that can calculate the posture of end effector with the positions and values of the 6 joints. It is included in the product and can be used for position calculation.

### Work with Node.js for login and database of waypoints

The login function is just a prototype and there is no user management function provided. It is suggested to use Node.js and mongodb to save user data better and safer. The login page should also based on the user data in the Node.js database. In addition, the app can do more such as saving the program made by specific user with the database.

### Enhancement functionality of programming

The simple programming tool only provide the function of adding the waypoint to the list and save/load the list to/from external file. It can be extended to be more functional by adding waypoints removal and reordering function, simple logical and loop statements and variable operations.

### Stability of robot motion

When using the real robot, protective stops often occur. This is because the robot moves too fast and sometimes it misses the target point. To protect the robot, a protective stop is triggered by itself; then the robot is to be unlocked from its HCI. It is necessary to improve the driver module to avoid the robot from moving fast, stopping heavily and triggering protective stops.

### Compatible with other types of industrial robots

This web-based GUI app is compatible to Universal Robots UR5. However, it may also suitable for any type of 6-axis industrial robots or even any robots that using ROS. This product can be splitted into two parts: common part that is same to all robots, and special part that differs to adapt each robot. When adapting a new type of robot, the common part should not change, while the special part can be copied from sample and modified to fit the data and settings of the new robot.

### Make good use of Vue.js

Only part of the code is under Vue.js environment. Putting everything inside the Vue object in app.js file can make good use of Vue.js functionalities and make further coding easier.

# Evaluation

This is not a chapter, and therefore has no number and no paragraphs. Just like the *foreword* the evaluation is a personal part of the report and you can write this component also in the ‘I’-shape. You reflect on the experiences you have had during the assignment. You oversee the whole journey and you discuss what you've learned. You describe what you've found and what you remember as your most "teachable moments" , i.e.: when did the error(s) or problem(s) occur and why; especially how you've solved the problems.

This is not the place to settle outstanding accounts. But suppose there was a profound reorganization at your Department, where many people are transferred or dismissed, then of course this influenced your work, and you need to mention this. But do this carefully, without recriminations.

# References

*About ROS*. (n.d.). Retrieved from ROS.org: http://www.ros.org/about-ros/

Foote, T. (2013). tf: The transform library. *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*, (pp. 1-6).

*Introduction.* (n.d.). Retrieved from Vue.js: https://vuejs.org/v2/guide/

Sucan, I., & Chitta, S. (n.d.). Retrieved from "MoveIt!": http://moveit.ros.org

Toris, R., Kammerl, J., Lu, D., Lee, J., Jenkins, O., Osentoski, S., . . . Chernova, S. (2015). Robot Web Tools: Efficient Messaging for Cloud Robotics. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*

Universal Robots A/S. (2009). *UR5 User Manual.*

*XML Robot Description Format (URDF).* (n.d.). Retrieved from ROS.org: http://wiki.ros.org/urdf/XML/model

# Attachment A: Project Plan (Excerpts)

All attachments are numbered with Roman numerals or with capital letters, and have an informative. The information in an annex supplements the report, but the reader must be able to read the main report (the report without the attachments) completely independently of the attachments.

One of the attachments of your report can be your *Assignment Plan.*

During the assignment you probably write a large number of documents for the company. It is expressly not intended that these documents are, all in their entirety, part of the *attachments*. Include only documents or portions of documents that are relevant for the reader who wishes to read additional information.

# Attachment B: Links of Videos

Videos are made during working period for showing the progress to the company.

Demonstration of Prototype 0:   
<https://youtu.be/MVroEbGX5Gc>

Demonstration of Prototype 1:   
<https://youtu.be/7M6ThsQKKKE>

Demonstration of Prototype 2:  
<https://youtu.be/Pqc1IhRfw_w>

Demonstration of final product:  
<https://youtu.be/iIB0xDJjZNA>

