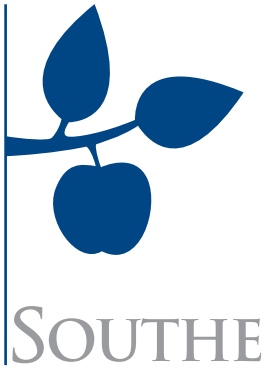
Bachelor



University of Southern Denmark

Extended user interface and simplification of the interactions with RobWork

Anders Ellinge
aelli14@student.sdu.dk

Mathias Elbæk Gregersen
magre14@student.sdu.dk

This page is intentionally left blank

Informations

Institution:	University of Southern Denmark – The Technical Faculty
Course code:	RB-BAP6-U1
Project title:	Extended user interface and simplification of the interactions with RobWork
Supervisors:	Lars-Peter Ellekilde, lpe@mmmi.sdu.dk Thomas Nicky Thulesen, tnt@mmmi.sdu.dk
Project period:	1. of february – Sometime in 2017
Number of pages:	4 pages
Character count:	48,000 (excl. references, appendices etc.)
Signatures:	

Anders Ellinge
aelli14@student.sdu.dk

Mathias Elbæk Gregersen
magre14@student.sdu.dk

Abstract

Your abstract goes here... ...

Contents

Foreword **II**

List of Abbreviations and Symbols **III**

1 Introduction **1**

2 Chapter 1 **2**
2.1 We write shit 2

3 RobWork library and its functionalities **3**
3.1 Frames 3
3.2 WorkCells 3

4 Describe functionality the user needs, and sort in order of importance **4**

Foreword

- The foreword page should describe the aim of the thesis, and its various research stages, and present the partners, funding and circumstances involved in the thesis project. The forewords should also include words of gratitude, addressed to people who have been incremental in your thesis-writing process. The supervisor, the second examiner, and the technical supervisor can be mentioned as well.

List of Abbreviations and Symbols

RW RobWork
RWS RobWorkStudio
Diller Greger

1 Introduction

- Describe the background and motivation of your thesis
- Describe in detail the objectives of your thesis (i.e. what is your aim to achieve), and on what basis the scope of your research area has been chosen
- At the end of the introduction, you might want to give an overview about the structure of your thesis

RobWork is a collection of C++ libraries for simulation and control of robot systems. RobWork is used for research and education as well as for practical robot applications. It has been noted that there exists different tedious processes of editing the environment of the RobWork simulation, involving reconfiguring files, unintuitive user interaction or reloading software. If some of these processes can be shortened or made more intuitive, the user experience of RobWork would greatly benefit from it. An example of an above-mentioned process: Adding a specific object to the environment of a simulated robot requires altering an .XML file and reloading RobWork. To specify goals for this bachelor project, the following features to implement is stated in order of importance:

- A user interface to initialize and load objects into the environment of RobWork with a simple drag and drop-like feature.
- Changing the configurations of the simulated robot in an intuitive matter, involving cursor interaction pulling the robot into the desired position.
- Adding tools or end effector to a simulated robot in a drag and drop-like feature. The tool should then be able to snap unto various parts of the robot.

With skills and experience within software development, object oriented C++ and various programming oriented skills, the authors of this project will try to solve and implement the goals stated. The first step of the project would be to become familiar with RobWork and learn how the source code works. The next step is to design the software for the first feature and disclose a possible solution. The third step is to implement the first feature and test it. The success criteria for the project is a finished, tested and working feature for RobWork of the before mentioned first goal set for this project. If this criterion is met early, the project would continue in similar manner with the second and third feature.

2 Chapter 1

2.1 We write shit

shit shit shit shit

3 RobWork library and its functionalities

This chapter is a general introduction to the RobWork library and the most commonly used data structures and functionalities within.

3.1 Frames

One of the most common data structures from the RobWork library is a frame. A frame is the basic building block in the RobWork library, representing (in the case of RobWork) a local 3D cartesian coordinate system.

In RobWork frames come in 3 different types: fixed frames, moveable frames and joints. Fixed frames are frames that have a constant transform relative to the parent frame.
Movable frames are frames which transform can be freely changed.
Joints are frames that can be assigned values for position, velocity limits and acceleration limits. This type of frame is usually used for devices. Joints can be further divided into 4 subtypes: prismatic joint, revolute joint, dependent joint and virtual joint.

Prismatic joints are joints which motion is linear along a constant direction. Thinking of a pneumatic piston can be an intuitive way of thinking about prismatic joints.

Revolute joints are joints which motion is based on a rotation around a single axis. Thinking of hinges can be an intuitive way of understanding revolute frames.

Dependent joins refer to joints which transform depends on on or multiple other joints. Dependant joins can also be divided into 2 subtypes, dependent prismatic joints and dependent revolute joints, adding the motion specification of the prismatic joint and revolute joint previously mentioned.

Virtual joints BLAH BLAH BLAH.

Frames in a WorkCell are required to have a parent and are given a unique name so that no frames can be confused for another. Only one frame in the WorkCell has no parent. This frame is called WORLD and is created when the WorkCell is constructed. The WORLD frame can be seen as the global 3D cartesian coordinate system for the WorkCell.

3.2 WorkCells

A WorkCell is the basis data structure of RobWork. The WorkCell can be thought of as a box containing all of the other structures and information needed. The WorkCell most commonly contains Frames, Objects, Devices and more which are used to represent the different items in the environment. However just having the

4 Describe functionality the user needs, and sort in order of importance

- Loading objects into the WC, this does not necessarily mean with a drag and drop-like motion, but rather as a proof of concept without regards to the GUI.
- Simple browser for objects with a load button.
- Improvement of the GUI, this means adding a drag and drop-like feature to the GUI, where the user would drag an object from the object browser window into the WC.