

BACHELOR PROJECT ASSIGNMENT

Student: David W o l l e r

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Bachelor Project: Automatic Planning of Robot Motion

Guidelines:

1. Analyze the requirements for robot motion within the manufacturing robotic cell.
2. Study and document the existing approaches for automatic planning of robot motion.
3. Test the automatic planning on given practical problem(s), perform simulations and practical experiments if possible.
4. Summarize the experimental results, conclude, and propose solutions.

Bibliography/Sources:

- [1] Robot Operating System, available documentation
- [2] Open Motion Planning Library, <http://ompl.kavrakilab.org/>
- [3] T. Chettibi, H.E. Lehtihet, M. Haddad, S. Hanchi: Minimum cost trajectory planning for industrial robots, European Journal of Mechanics - A/Solids, Volume 23, Issue 4, 2004, Pages 703-715, ISSN 0997-7538, <http://dx.doi.org/10.1016/j.euromechsol.2004.02.006>

Bachelor Project Supervisor: Ing. Vladimír Smutný, Ph.D.

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, October 24, 2016

Bachelor Thesis



**Czech
Technical
University
in Prague**

F3

**Faculty of Electrical Engineering
Department of Cybernetics**

Automatic Planning of Robot Motion

David Woller

**Supervisor: Ing. Vladimír Smutný, Ph.D.
May 2017**

Acknowledgements

I would like to thank my supervisor Ing. Vladimír Smutný, Ph.D., for guidance, eClub ČVUT and Ing. Jan Šedivý, CSc., for given opportunities to grow professionally and my parents for their support throughout my whole life.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within in accordance with the methodical instructions for observing the ethical principles in the preparation of university thesis.

Prague, May 25, 2017

Abstract

This thesis focuses on the available tools used for robotic cell design, especially automatic motion planning. It consists of a research on robotic cell design tools, motion planning tools and possible methods of human body motion tracking. Some of the motion planning algorithms are tested on an experimental task and used for comparing different robots usable for this task. Automatic motion planning applicability is also compared with traditional manual planning.

Keywords: robot motion planning, robot trajectory planning, motion tracking, motion capture, cycle time, design support tools, robot simulation tools, robotic cell

Supervisor: Ing. Vladimír Smutný, Ph.D.
Praha 6, Jugoslávských partyzánů 3, B-608b

Abstrakt

Tato práce se zabývá prostředky pro usnadnění návrhu robotické buňky, především pak automatickým plánováním trajektorií. Sestává z řešení, která zkoumá dostupné prostředky pro návrh robotické buňky obecně, prostředky pro automatické plánování trajektorií a možné metody záznamu pohybu člověka. Automatické plánování trajektorií je poté testováno na experimentální úloze, přičemž výsledky jsou použity pro porovnání vhodnosti různých robotů pro tuto úlohu. Zároveň je automatické plánování trajektorií porovnáno s tradičním manuálním programováním.

Klíčová slova: plánování pohybu robota, plánování trajektorií robota, záznam pohybu, sledování pohybu, čas cyklu, prostředky návrhu robotické buňky, simulace robotické buňky, robotická buňka

Překlad názvu: Automatický návrh trajektorie robotu v robotické buňce

Contents

1 Introduction	1
1.1 Motivation and goal of the thesis	1
1.2 Thesis structure	2
2 Existing solutions	3
2.1 Motion planning tools	3
2.1.1 Inputs and outputs of motion planning	3
2.1.2 ROS related tools	4
2.1.3 Commercial tools	6
2.1.4 Motion planning - conclusion	7
2.2 Existing design support tools	8
2.2.1 Decision tools	9
2.2.2 Simulation tools	9
2.3 Human body motion tracking	11
2.3.1 Optical-passive motion capture	11
2.3.2 Using inertial measurement units	12
2.3.3 Lighthouse technology	12
2.3.4 Structured light technology	14
2.4 Conclusion	15
3 Motion planning experiment	17
3.1 Task description	17
3.2 Cell description	18
3.3 Evaluated robots	21
3.3.1 Grippers	21
3.4 Experiment workflow	23
3.4.1 Robot and cell description	23
3.4.2 Trajectory planning	23
3.4.3 Cycle times determination	24
3.5 Simulation results and cycle times	24
3.5.1 Simple planning	24
3.5.2 Path length optimization	26
3.5.3 Cost of changing grip	28
3.5.4 Mechanical work optimization	30
3.5.5 Manual planning	32
3.6 Real cycle times determination	32
4 Conclusion	35
4.1 Results discussion	35
4.2 Comparison with standard planning	35
4.3 Applications of automatic planning	36
4.4 Limitations	36
Bibliography	37
A Content of the attached CD	41

Figures

2.1 VICON motion capture system - optical-passive[3]	11
2.2 Xsens IMU sensor suit[4]	13
2.3 HTC Vive set - lighthouse[1]	14
2.4 PrimeSense sensor - light coding[2]	14
3.1 Semi-finished and final product - thermometer tube	18
3.2 Original layout provided by Modia	19
3.3 Final layout - RViz visualization	20
3.4 Dual gripper with custom revolute joint attached to Epson G20 A01-S	22
3.5 Dual gripper attached to Kuka sixx	22
3.6 RRT expansion[40]	25
3.7 RRT expansion compared to RRT star expansion on the same data[45]	27
3.8 Pad for grip changing	29

Tables

3.1 Epson G20 A01-S cycle times, simple RRT	25
3.2 RRT star, minimal path length optimization criteria, all robots cycle times and custom joint angles for Epson robots	27
3.3 Grip change cycle times for all robots and custom joint angles for Epson robots	29
3.4 Kuka sixx - physical parameters of links - estimate	31
3.5 Kuka sixx - estimated inertial properties	31
3.6 Kuka sixx cycle times - minimal work optimization	31
3.7 Manual planning cycle times	32
A.1 Content of the attached CD	41

Chapter 1

Introduction

1.1 Motivation and goal of the thesis

This thesis was assigned in collaboration with Czech company Modia[5], that specializes in assembly and production automation, CNC production of exact parts and tool production. As Modia is rather small company focused on piece and small-batch customized production, it must use the potential and time of its designers very carefully and in the most efficient way. Thus, Modia formulated several repetitive difficulties they face during robotic cell design and asked about possible improvements.

The task was to examine ways how to facilitate the automation of existing production process. Let's assume that the production process is completely or partially human operated and that we want to fully automate it. In that case, an experienced design engineer analyzes the worker activities and empirically designs the new robotic cell layout. Then, actions and movements of the robot are manually programmed. It is not uncommon practice that the robot program is written for the first time only after physical installation. Therefore, efficiency of the final design depends greatly on the designer's experience and on the amount of time invested in development.

Two phases of the process described were selected for deeper analysis and search for facilitation. The first one is usage of human body motion tracking in problem definition - the idea is to capture the actions and movements performed by a worker and automatically or semi-automatically transform these data into geometric description (technical drawing with recorded positions and actions) of the production process. The second one is application of motion planning algorithms for robot movement planning, rather than manual programming.

Both topics were subject of a research and besides that, simulations implementing motion planning algorithms were carried out on a real process to be automated provided by Modia. Goals of these simulations were to:

- Test different motion planning algorithms on the given task
- Determine cycle times for different robots
- Compare manual planning with automatic planning

■ 1.2 Thesis structure

Chapter 2 is a research on existing tools for robotic cell design process facilitation. It lists and describes existing robot motion planning tools in section 2.1, design support tools in general in section 2.2 and examines technologies usable for human motion tracking in section 2.3.

Chapter 3 documents the motion planning experiment. It consist of task definition and experiment workflow description in sections 3.1 to 3.4, data from simulations in section 3.5 and results discussion, together with conclusion and suggestions, in chapter 4.1.

All source codes and data from simulations are recorded on the enclosed CD.

Chapter 2

Existing solutions

This chapter examines possible methods of simplifying the design process of industrial robotic cell. It is divided into three sections:

- motion planning tools;
- existing decision, simulation, and design supporting tools;
- human motion tracking.

2.1 Motion planning tools

This section focuses on existing motion planning tools for industrial robots. For better clarity it is further split into two subsections, of which former describes a set of tools based on Robot Operating System, whereas the latter focuses on commercial tools.

2.1.1 Inputs and outputs of motion planning

Before proceeding to the actual motion planning tools, let us elaborate a bit more on the topic of inputs and outputs of this task.

Inputs

There are three main requirements, no matter what kind of path planner one will be using. These are:

- Visualization and collision model

These models are needed both for the robot and for its surroundings. Visualization model is usually a detailed and quite accurate visual representation utilised in simulation. Nowadays is typically transferred from CAD tools and for industrial robots it is available on the web.

Collision model is needed in the planning process itself, to plan a collision-free path avoiding all obstacles and collisions. It is usually highly simplified version of visualization model, to increase planning speed. It should incorporate all components including cables. Usually it is a bit

In combination with Rviz visualization tool[32] it can serve as an interface between any robot and motion planning algorithms implemented in the libraries mentioned below.

■ Open Motion Planning Library

Open Motion Planning Library[28](OMPL) is very extensive ROS library containing many state-of-the-art sampling based motion planning algorithms. It doesn't contain any collision checking or visualization tools as these are provided by MoveIt!. It is also possible to implement and add custom planning algorithms using a predefined interface.

OMPL includes many different algorithms that can be suitable for various applications. The algorithms implemented can be classified into the following sets.

- Multi-query planners

These planners build a roadmap of the entire environment. After that, this map can be used for multiple queries.

- Single-query planners

Each time a query is made, these planners grow a new tree (or two trees) of states connected by valid motions. The trees are grown using various heuristics.

- Optimizing planners

This set of planner covers the planners with heuristic based on some optimization criterion. That can be e.g. length of the path, amount of mechanical work, overall cost of states etc. There is also a general framework for setting up custom criteria for optimization.

- Control-based planners

There is a significant subset of applications, where the trajectory of robot is partly defined by character of the application. Such trajectories are called "semi-constrained Cartesian paths" or with "differential constraints". An example of this might be sanding, polishing or spray painting. The control-based planners are designated specifically for this kind of applications.

■ ROS-industrial Scan-N-Plan

ROS-industrial[30] is an industry-developed project based on ROS platform. It is based on the desire to broaden the portfolio of tasks processed by robots. It's repository includes interfaces for common industrial manipulators, grippers, sensors, and device networks. There are also some software libraries for real time planning from 3D scan data - Scan-N-Plan[31]. This particular library is still in experimental state.

ROS-industrial can be considered as well established, reliable part of the ROS, so it can be used in the industrial applications.

trajectories of e.g. large and structured elements, that are being assembled together. Presumably, the trajectory of such an element is then transformed to trajectory of the manipulator's end effector and finally, feasibility of such a trajectory is checked in a simulation using robot inverse kinematics.

Price of this product remains unknown. Beneficial effect of the approach described in the previous paragraph is limited to very complicated assembly applications. IPS AB provides many references to their product users, most of whom are renowned car manufacturers.

■ Convergent Information Technologies

Convergent Information Technologies[6] is an Austrian company specialized in providing tools for automated programming of robots and processes. It's main product - AutomAPPPS - is directed to the following applications:

- Automatic off-line programming
- Automatic on-line programming
- Motion planning and bin picking

No details about the planning algorithms implemented or price of particular software modules are publicly accessible. However, the company itself claims, that more than 1 million robot programs were automatically programmed with their software. It is very unlikely, that number of robots actually running on this software would be anywhere near, as 1 million robots would be a significant portion of all the robots in use worldwide. Nevertheless, some references including concrete end user names and pictures from applications are provided on their website.

■ 2.1.4 Motion planning - conclusion

There are two major groups of tools for automatic robot motion planning - the open source ROS libraries and commercial products from either rather small automation-specialized companies (Industrial Path Solutions, Convergent Information Technologies), or even some large industrial corporations (Siemens).

■ ROS tools

The main advantages of planning trajectories in ROS are:

- State of the art algorithms
Especially the OMPL provides wide variety of implemented and debugged algorithms.
- Customizability
Implementation of one's own algorithms for trajectory planning is fully supported and their application on any robot is theoretically possible.

■ 2.2.1 Decision tools

Perfect instrument for a robotic cell designer would be an expert system, that could cover not only the process of trajectory planning and application analysis from human motion, but also other problems. These other problems are for example:

- (semi)automatic choice of components
- (semi)automatic design of robotic cell
- Virtual process simulation

There are some research activities focused in this direction (for example the German project called Manuserv[10]), but all of them are still at very early stage. The only mature part of the design process facilitation is a simulation.

■ 2.2.2 Simulation tools

Simulation tools are widespread and extremely useful, when it comes to designing a complex robotic system. They can be very helpful for several reasons, such as: feasibility assessment, comparison of different designs, estimation of cycle times etc., without having to spend money for experimental setups or even machines only assumed to be suitable. Many of the renowned robot manufacturers offer customers some simulation software compatible with their robot portfolio. Often, the customer is provided with not only environment for carrying out his process simulations, but also robot models (visual and collision) and the most common end effectors.

However, simulation preparation can be quite time consuming, especially when the customer combines products of more manufacturers or he uses some custom made elements. In such cases, visual and collision models can't be easily imported and have to be created manually. Some of the most common simulation tools available are described below.

■ KUKA Simulation software

KUKA provides a wide portfolio of simulation software in their product line called KUKA.Sim[14]. It consists of the following four partially independent tools.

KUKA.Sim Layout is a primary KUKA tool for robotic cell design and simulation. It contains libraries with visual and collision models of all the KUKA robots, end effectors, basic conveyor belts and protective fencing structures. It has a user friendly GUI oriented on easy collision and reachability checking. Its main purpose is to provide a virtual platform for easy comparison of different robotic cell designs.

KUKA.Sim Pro is an extension of Kuka.Sim Layout and therefore it has all of its functionality. In addition, it allows the user to carry out simulations of high accuracy with respect to time. Therefore, it should



Figure 2.1: VICON motion capture system - optical-passive[3]

■ 2.3 Human body motion tracking

Last section of this chapter is dedicated to technologies applicable to human body motion tracking. The idea is to capture trajectories of worker body and particularly his hands in a repetitive process and to transform this data into a technical description. This description should contain all important positions present in the workflow and would be used as an input of automatic robot trajectory planning.

The key parameters of motion tracking are:

- Accuracy of tracking
- Ease of installation
- Applicability in industrial conditions
- Price of hardware

The price of additional software is not considered, as our application is quite atypical. It will surely require some supplementary programming work and the commercial software usually used might not be needed. Below, some of the most relevant technologies are described.

■ 2.3.1 Optical-passive motion capture

This technology is the most common one in commercial applications of motion capture (e.g. Computer-generated imagery in movies or PC games). The capturing system uses (usually retroreflective) markers attached to the human body. Their position in space is then captured by multiple high resolution and high speed infrared cameras.



Figure 2.2: Xsens IMU sensor suit[4]

the controllers uses IMU sensors as well, therefore their position can be determined very accurately. The system is designed to work in a 4.6 by 4.6 meters large tracking space.

Accuracy of position measuring is stated to be sub-millimeter. That can be hardly obtained even by the state of the art optical-passive motion capture systems. However, a simple experiments carried out at CTU Prague achieved standard deviation of 3 millimeters. A great disadvantage might be, that the position measured is bound to a certain point in the middle of the controller, so the person captured could only pretend execution of his workflow.

Installation of this system is very simple - one only need to place lighthouse basestations into appropriate positions, so that controller visibility is at least partially ensured.

Applicability in industrial conditions is limited, as the tracking space 4.6 by 4.6 meters can be insufficient. Also the fact, that worker could only imitate workflow using handheld controllers might cause imprecision. However, the accuracy of sensors is very good and installation extremely simple.

Cost of the HTC Vive set is currently 25 000 CZK.



Figure 2.3: HTC Vive set - lighthouse[1]



Figure 2.4: PrimeSense sensor - light coding[2]

■ 2.3.4 Structured light technology

Similarly to previous lighthouse technology, light coding is a low cost motion tracking technology mainly used in game industry. It is based on structured light and camera technology of computer vision. It was developed by PrimeSense[22], an Israeli company currently owned by Apple. Structured light illuminates the scene observed by a camera. The distance from the sensor to the surface is then decoded from camera image. This technology is a basis of many gaming devices, such as Asus Xtion[18] or Microsoft Kinect[21].

Accuracy is directly proportional to distance from object, as the depth camera resolution is 640 by 480. With range from 50 centimeters to 5 meters, it theoretically leaves circa 0.75 millimeters per pixel at close range and 7.5 millimeters per pixel at maximal distance. However, the manufacturer states, that additional noise is nonlinear and assumes inaccuracy up to 5 centimeters at long range.

Installation is simple, however this solution is strictly for indoor use and

full visibility is necessary for complete 3D reconstruction.

Applicability in industrial conditions is right now very low due to significant inaccuracy at larger distance. However, it is likely that the resolution will improve over time and this technology is also the only one, that theoretically enables precise capture of hands and even fingers.

Cost of different sensor sets ranges from 5 000 to 25 000 CZK.

■ 2.4 Conclusion

Concerning motion planning tools, there is only a few commercial tools available and then some open source tools based on ROS. It appears, that research in this field is still in progress and therefore none of the major robot manufacturers provide any motion planning tool within their software portfolio yet. Open source tools based on ROS are at the moment more advanced and thus recommended for experimenting.

Talking about other design supporting tools, only simulation tools are mature enough to be helpful and widely used. No expert system capable of autonomous design of robotic cells exists yet.

Lastly, on the subject of human body motion capture - there are various methods and technologies available, usually from other fields of use than robotics (film industry, game industry). As their purpose is always different from our needs, their precision in capturing only certain body parts (most probably hands) in very high accuracy might be insufficient.

Chapter 3

Motion planning experiment

A practical problem was defined in cooperation with Czech company Modia[5]. Modia proposed a manufacturing process, that is going to be automated.

3.1 Task description

The object of this manufacturing process is a small copper tube with a length of 5 cm and a diameter of 1,5 cm. This tube is part of a thermometer used in automotive industry and every day several thousand pieces are produced.

During the process to be automated, the tube is flattened as seen in figure 3.1, calibrated and checked for surface defects and imperfect flatness. Currently, the first and second operation are performed on man-operated mechanical presses at two separate workplaces and all quality control is done manually.

Modia plans to increase production volume and simplify the entire process, thus it was decided that all the operations listed will be concentrated in one robotic cell and operated by robot. Before the actual design process and cell construction in Modia, several experiments implementing automated motion planning are to be carried out as a part of this thesis. Goal of these experiments is to make some of the decisions, that would normally have to be done by an experienced engineer, easier. The main issues are:

- Cycle time

As the production volume is to be increased, it is crucial to determine, whether or not a single robotic cell is more productive than the current workflow. Eventually, it is an important information for planning of cells: number of cells needed and also cost of automated solution.

- Choice of robot

There are various robot types applicable to the same task and it is desirable to choose the most appropriate one. This includes many parameters - cost, reliability, service support, performance etc. Exact simulation and optimally planned trajectories for different robots can help us with the evaluation of the last one - performance and suitability for a particular task.



Figure 3.1: Semi-finished and final product - thermometer tube

- Separation of simple operations

Even though all the "pick and place" operations present in the process could be performed by a robot with a suitable gripper, it may be beneficial to design some independent simple mechanisms (e.g. motors for tube insertion to forming or calibrating tool) to simplify the robot's task and therefore improve cycle time. However, such custom mechanisms significantly increase overall price of the robotic cell and unlike the robot itself can't be reused later in different application.

■ 3.2 Cell description

Layout of the robotic cell was provided by Modia and is shown in the Figure 3.2. This layout was later slightly modified, as some of the positions (camera and plane check control) turned out to be larger than originally intended. Final version of the layout is shown in the Figure 3.3 (which is a RViz visualization).

Positions present in the cell are:

- Input magazine

This position is not operated by the robot. In the input magazine, the freshly cut tubes are stored, put into vertical position and transported to the cleaning position one at a time.

- Cleaning position

Tubes can have burrs after cutting. In this position, tubes are cleaned with a deburring tool and placed into vertical position. Then, robot grips the tube here, so that the tube axis coincides with gripper axis.

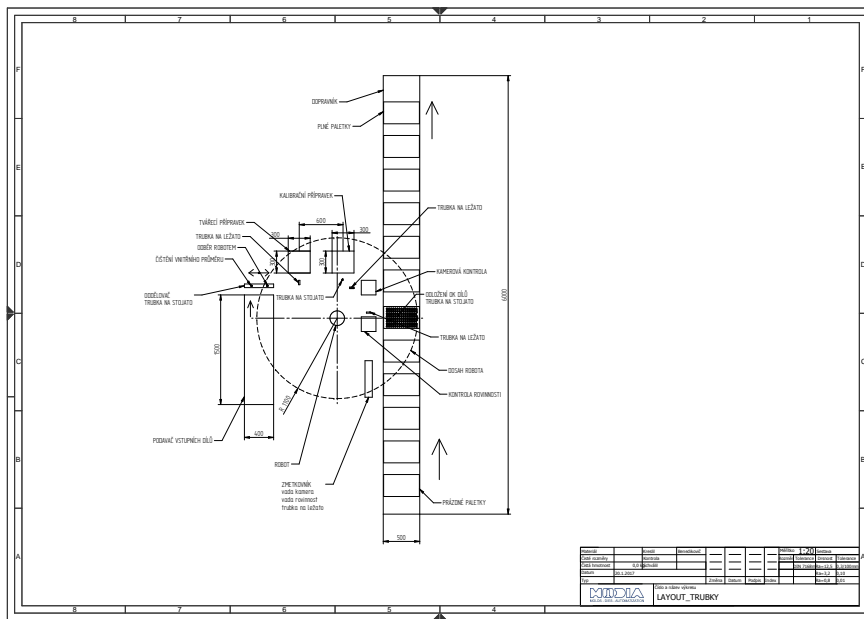


Figure 3.2: Original layout provided by Modia

■ Forming tool

The tube is placed to the forming position in a horizontal position. Then, it is pushed inside the forming press, secured against falling out and flattened lengthwise, as seen in Figure 3.1. After that, it is ejected from the forming tool and gripped again by the robot.

■ Calibrating tool

The tube is placed to the calibrating tool in a vertical position. Here, the inside of the tube is driven through by a calibration thorn. The purpose of calibration is to ensure, that the tube wasn't inadvertently deformed during forming procedure and that it meets the accuracy requirements. The tube is placed, inserted, calibrated, ejected and gripped again by the robot.

■ Camera check

Camera check serves to detection of surface defects such as scratches and pressure marks, that might have emerged during previous manipulation. The tube is positioned horizontally in front of a camera with a fish eye lens, illuminated and photographed. The tube is held by the robot all the time.

■ Plane check

This position inspects flatness of the flattened side of the tube. It consists of a laser source and a IR sensor placed on a moving platform. The tube held by the robot is placed horizontally above this platform, which then moves along the entire tube.

3. Motion planning experiment

- Garbage collector

Should any of the two control positions discover intolerable defects, the tube is dropped to the garbage collector.

- Conveyor with containers

Tubes that have successfully passed both controls are placed vertically to a container, placed on a conveyor.

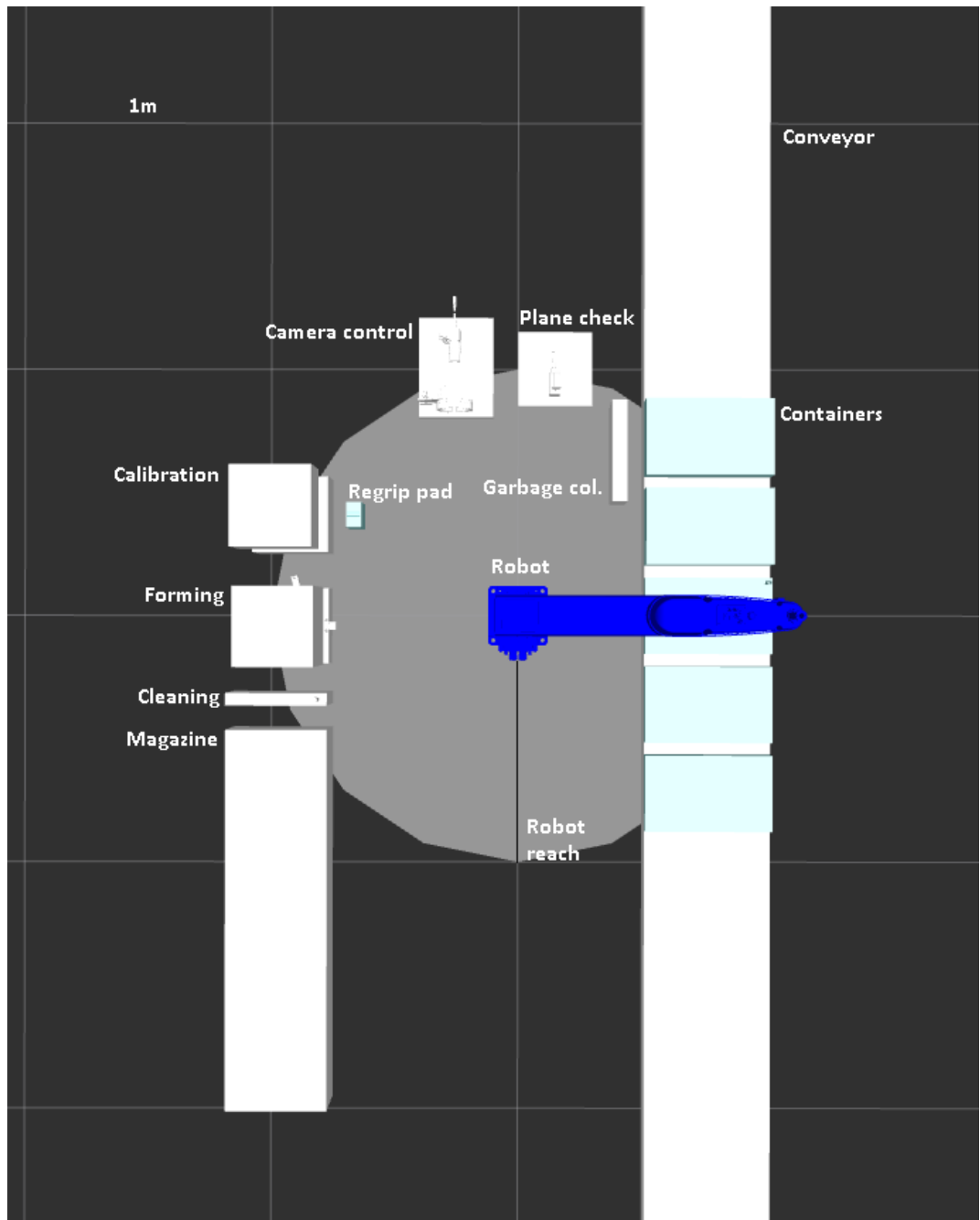


Figure 3.3: Final layout - RViz visualization

For best performance, the cell layout should be optimized in simulation as well. However, Modia required such deployment, that would enable easy operation by a person in case of long-term robot malfunction. Therefore, the locations of individual positions are fixed parameters and the layout remains almost unchanged for all tested robots.

Modia provided CAD drawings in .step format for some of the positions (forming, calibrating, camera and plane check). Visualization files for the robots are publicly available on manufacturers' sites. This enabled creation of accurate collision model and thus precise motion planning.

3.3 Evaluated robots

One of the main goals of the simulations was to determine, which robot can achieve the best cycle time. For this comparison, four robots (respectively two variants of two robots) were chosen in cooperation with Modia. The robots compared are:

- KR Agilus sixx R1100

KR Agilus sixx[35] is a Kuka six-axis robot. R1100 denotes the maximal reach in millimeters. It can handle payload up to 10 kg and was tested in tabletop mounting.

- KR Agilus five R1100

KR Agilus five[34] is the same structure as KR Agilus sixx, but it is only five-axis robot. Besides some minor differences in range of some joints, most of the parts are identical.

- Epson G20 A01-S

Epson G20 A01-S[38] is a SCARA (Selective Compliance Assembly Robot Arm[41]) robot with combined arm length of 1000 mm (thus slightly lower reach than the Kuka robots), z-axis stroke of 180mm and payload up to 20 kg. The S stands for standard version, tabletop mounting.

- Epson G20 A01-SR

Epson G20 A01-SR[38] is a variant of the previous Epson robot, with most of the parameters identical. SR stands for standard version, roof mounting - the robot structure is modified so that it can be placed above the cell and therefore enable better coverage of the robotic cell.

3.3.1 Grippers

Besides the robots selection, it was necessary to choose a suitable robot end effector. As the only requirement articulated by Modia was that the end effector should be dual (e.g. it should allow two parts to be gripped at the same time), the choice was arbitrary. It was decided to install two Schunk EGP 25-N-N-B[42] grippers pointing in opposite directions.

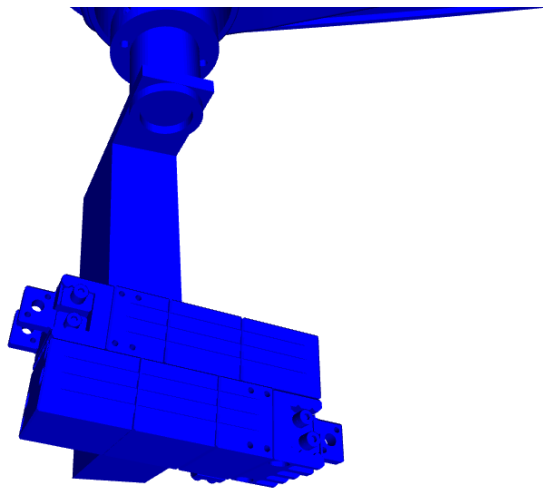


Figure 3.4: Dual gripper with custom revolute joint attached to Epson G20 A01-S

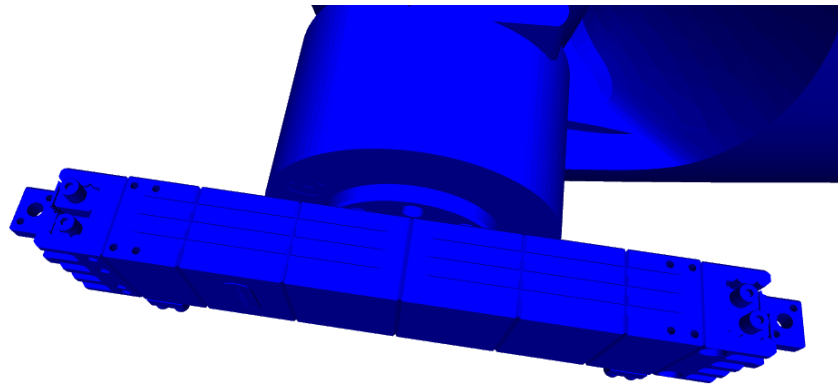


Figure 3.5: Dual gripper attached to Kuka sixx

As the SCARA robots didn't provide sufficient number of degrees of freedom, it was necessary to add another revolute joint between the last robot arm (vertical shaft) and the gripper pair. Properties of this unit were estimated as follows:

- Weight: 300 g
- Dimensions: $[x,y,z]=[0.04 \text{ m}, 0.025 \text{ m}, 0.01 \text{ m}]$
- Range of motion: $(-360^\circ, 360^\circ)$

Both dual gripper designs are shown in figures 3.4 (Epson variant with custom joint) and 3.5 (simple Kuka variant).

3.4 Experiment workflow

The motion planning experiment consists of several consecutive steps and each one of them requires use of different software tools, which may be confusing. This section is dedicated to slightly more detailed description of these steps.

3.4.1 Robot and cell description

Robot and cell were described in URDF (Universal Robot Description File), which is a XML file format used in ROS. Typical robot description consists of:

- Robot tree structure

Definition of joints, links and geometric relations between them. Denavit Hartenberg[36] notation can be used for an unambiguous description. However, DH notation wasn't used in this experiment, because it would cause deviations from coordinate systems defined by robot manufacturers. Therefore, the trajectories planned would require additional transformation of joint coordinates before use in manufacturer's simulators or robot control units.

- Visual robot model

This step is not strictly necessary, but is beneficial for visualization and inspection purposes. Visual model can be reconstructed from CAD files such as .step, usually supplied by the robot manufacturer.

- Collision robot model

Ideally, collision model would be identical with visual model, but that would put enormous demands on computing power during motion planning. Therefore, use of simple geometric shapes roughly wrapping the robot is inevitable.

- Robot physical properties

Robot physical properties are essential for optimal planning with other than geometric optimization criteria and for accurate determination of cycle times. It is perfectly possible to add them to robot .urdf, however most of the parameters necessary remains secret. In this experiment, an estimate for one of the robots is made and use of these parameters demonstrated.

3.4.2 Trajectory planning

For trajectory planning in ROS, it is necessary is to generate SRDF[43] (Semantic Robot Description Format). SRDF complements previously mentioned URDF with joint groups, default robot poses, additional collision checking etc. and can be generated using the MoveIt! Setup Assistant. Then, a planning

script for the motion plan definition has to be written, either in Python or C++. Individual OMPL algorithms can be configured either via the C++ ROS API, or directly in the `ompl-planning.yaml` file, which was generated together with the SRDF.

■ 3.4.3 Cycle times determination

As most of the robot physical and drive units parameters couldn't be obtained from manufacturers, accurate cycle times had to be determined while using official Epson and Kuka simulation software. Motion plans were therefore exported from ROS to `.txt` or `.tsv` files and simulated either in Epson RC+ 7.0 simulator[37] or in KUKA.Sim Pro[14].

■ 3.5 Simulation results and cycle times

In this section, the results of the realized simulations are presented. All trajectories data, URDFs, SRDFs and planning scripts are to be found on the CD enclosed.

For each of the algorithms, a rough estimate of run time needed for one task (movement from one position to another) is stated. These numbers were obtained on a laptop with a dual-core 2.2 GHz CPU and 4 GB 1333 MHz RAM.

■ 3.5.1 Simple planning

Before planning with various optimization criteria, the default planning algorithm RRT[39] was tested on Epson G20 A01-S.

■ RRT description

RRT (Rapidly-exploring Random Trees) is a tree-based motion planner, that works in the following steps:

1. sample a random state \mathbf{q}_r in the state space, within the defined range
2. find the state \mathbf{q}_c among the current states that is closest to \mathbf{q}_r and expands from \mathbf{q}_c to \mathbf{q}_r
3. repeat until goal state \mathbf{q}_g is reached

RRT expands equally in all directions and doesn't find optimal trajectories (thus, repeated planning does not improve the trajectory found in any way, but it is still useful should any of the planning attempts fail). However, when the range of random states sampling is defined sensibly, it can find a valid solution much faster than optimal planners described later.

RRT is therefore ideal for quick appraisal, whether or not the goal states defined are reachable from the current robot pose.

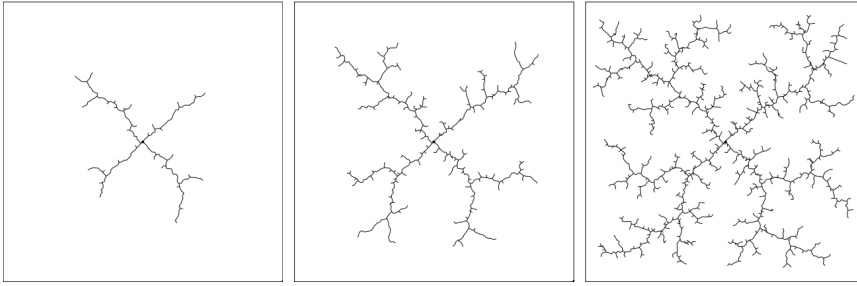


Figure 3.6: RRT expansion[40]

■ Simulation results

Table 3.1 shows the cycle times obtained for Epson G20 A01-S. For better comparison with other plans, the results are split into 8 steps:

1. Cleaning (1) - both grippers are empty, gripper 1 moves from container for finished products to cleaning position and grips new tube
2. Forming (2) - gripper 2 removes formed tube from forming position
3. Forming (1) - gripper 1 inserts tube to forming position
4. Calibrating (1) - gripper 1 removes calibrated tube from calibrating position
5. Calibrating (2) - gripper 2 inserts tube to calibrating position
6. Camera check (1) - gripper 1 places tube to camera check
7. Plane check (1) - gripper 1 places tube to plane check
8. Container (1) - gripper 1 drops tube to container, both grippers are empty

Goal position (gripper no.)	Cycle time [s]	Custom joint angle [°]
Cleaning (1)	8.3	143
Forming (2)	9.3	175
Forming (1)	4.7	183
Calibrating (1)	3.6	90
Calibrating (2)	2.1	90
Camera check (1)	4.9	136
Plane check (1)	3.3	23
Container (1)	7.9	238
Total	44.1	

Table 3.1: Epson G20 A01-S cycle times, simple RRT

Custom joint angle (CJA) column contains sum values of angular motion performed by the custom revolute joint,

$$CJA = \sum_{i=1}^{n-1} |\alpha_{i+1} - \alpha_i|, \quad (3.1)$$

where α_i is the current angular coordinate of the custom joint and n is the number of points in the particular trajectory planned.

Dynamic parameters of the custom joint weren't specified by Modia, only physical properties were estimated in section 3.3.1. Rather than influencing the robot cycle time with arbitrary values of speed, acceleration etc., it was decided to state the custom joint angle separately to get an idea. Time course of this joint coordinate is part of the trajectories data attached on CD. Modia could choose a specific drive later and asses, whether or not its performance would affect the cycle time.

The following times are not included in the cycle times stated:

- Gripper opening
- Gripper closure
- Camera checking
- Plane checking

These operations are common for all the robots simulated and their duration remains unknown. It is up to Modia to provide these values to get accurate cycle times, robot comparison is not influenced by this simplification. As for the remaining operations (cleaning, forming and calibration), it can be assumed that their duration does not affect the total cycle time thanks to the use of dual gripper.

The total cycle time obtained in this section is apparently excessive. This confirms the already articulated assumption, that this algorithm is suitable only for fast assessment of feasibility of the motion planning task.

PC time sufficient for successful planning for each of the 8 steps was 30 seconds and 10 planning attempts.

■ 3.5.2 Path length optimization

In this section, the simulation was carried out for all of the robots, as the algorithm used - RRT star[44] with minimal path length optimization criteria set - is suitable for the task and can provide valuable data for comparison.

■ RRT star description

RRT star consists of the same three steps as the previously described RRT 3.5.1. The only difference is, that apart from connecting randomly generated child node **qr** to some of the current nodes **qc**, RRT star also inspects all of the current nodes within a defined neighborhood of the newly added child **qr**. If any of them can trace up to the tree root via child at a

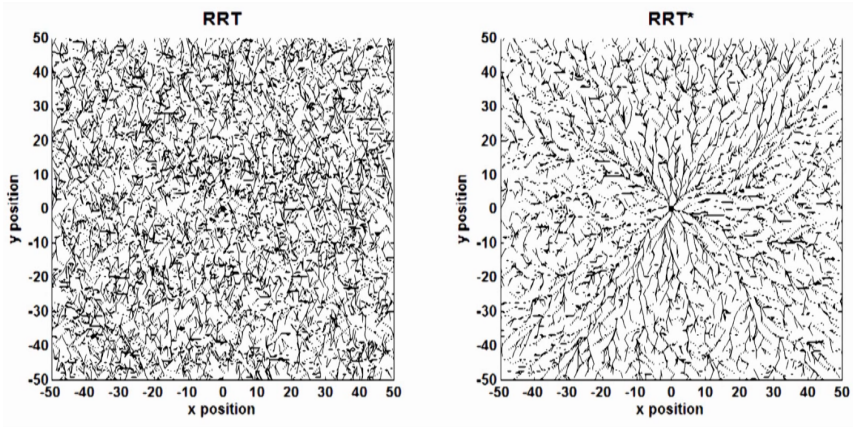


Figure 3.7: RRT expansion compared to RRT star expansion on the same data[45]

shorter distance than current connections, parent of this node is shifted to \mathbf{qr} .

Thus, RRT star smooths the tree branches in each step according to predefined metric and is asymptotically optimal.

However, its \mathbf{qr} neighborhood inspection consumes significant amount of the available computational resources and running time of RRT star is stated to be a constant factor of the running time required by RRT.

■ Simulation results

Goal position (gripper no.)	G20A01-S [s] ([°])	G20A01-SR [s] ([°])	Kuka fivve [s]	Kuka sixx [s]
Cleaning (1)	5.9 (159)	3.1 (104)	3.7	4.0
Forming (2)	3.9 (90)	1.4 (90)	1.7	2.3
Forming (1)	2.1 (180)	1.1 (180)	0.5	0.5
Calibrating (1)	1.3 (90)	1.1 (90)	1.2	1.2
Calibrating (2)	1.7 (180)	0.05 (180)	0.5	0.5
Camera check (1)	1.3 (90)	1.2 (90)	1.4	1.5
Plane check (1)	4.3 (30)	3.5 (69)	1.2	1.0
Container (1)	4.8 (95)	2.8 (116)	1.8	1.6
Total	25.2	14.4	12.0	12.6

Table 3.2: RRT star, minimal path length optimization criteria, all robots cycle times and custom joint angles for Epson robots

In table 3.2, cycle times and custom joint angles are listed together for Epson robots. For Kuka Robots, there are only cycle times, as they didn't require installation of additional drives.

There is a significant difference between the time achieved by floor mounted and ceiling mounted Epson robot. This is probably due to the fact that the (for SCARA robots fixed) vertical location of 2nd and 3rd robot joint differed

by 10 cm for the robots tested. Lower placed floor mounted robot was therefore forced to move between some of the obstacles, whereas the ceiling mounted robot could move above them. However, cell layout does not require any operation close to the robot base link. Thus, both robots should reach the same cycle time, were they placed equivalently.

For Kuka five robot, some of the goal positions were defined differently from the other three robots, because the five axis robot couldn't reach them. This applies to the following positions:

- Cleaning
- Calibration
- Container

For Cleaning and Container position, this meant only a small rotation of the tube position in the xy plane. Calibration position had to be rotated as a whole (both calibration tool and regrip pad).

Total cycle times obtained can be used for comparison of each robot suitability for the task, with Kuka five being the fastest.

PC time sufficient for successful planning for each of the 8 steps was 200 seconds and 30 planning attempts.

■ 3.5.3 Cost of changing grip

During previously realized planning, the tube axis coincides with gripper axis. However at the calibrating position the tube has to be inserted in an upright position and this can't be fully achieved while holding the tube as explained. Therefore, design of custom insertion mechanism was assumed. Such a solution might be effective, yet costly. An operation of changing grip, that would enable omission of insertion mechanism design, was simulated for all of the robots.

The algorithm used was RRT star optimized for minimal path length and the operation consisted of the following steps:

1. Gripper 1 is empty, gripper 2 holds formed tube and is placed at the forming position. Gripper 2 places tube on regrip pad
2. Both grippers are empty, gripper 2 grasps the tube from regrip pad, but this time the tube is perpendicular to gripper
3. Gripper 1 removes calibrated tube from calibrating position. Both grippers hold tubes perpendicularly
4. Gripper 2 inserts tube to calibrating position
5. Gripper 1 places calibrated tube on regrip pad. Both grippers are empty
6. Gripper 1 grasps tube so that the tube axis coincides with gripper axis

This process would replace operations Calibrating (1) and Calibrating (2) from the sequence described in 3.5.1.

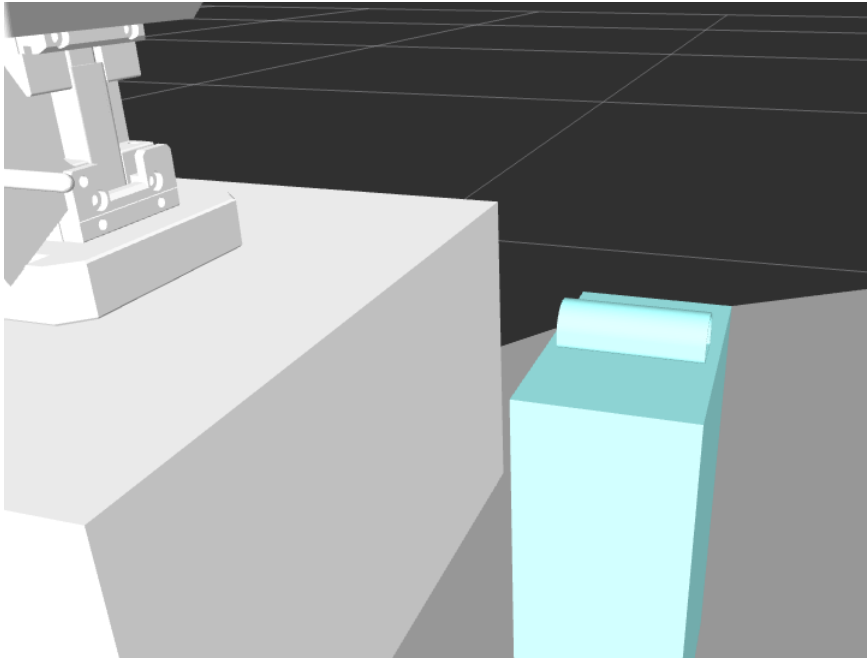


Figure 3.8: Pad for grip changing

Simulation results

Goal position (gripper no.)	G20A01-S [s] ([°])	G20A01-SR [s] ([°])	Kuka five [s]	Kuka six [s]
Step 1	1.4 (180)	0.4 (180)	0.7	0.6
Step 2	1.1 (270)	1.7 (270)	1.2	1.0
Step 3	0.2 (90)	2.6 (90)	1.3	1.3
Step 4	0.1 (180)	0.1 (180)	0.5	0.5
Step 5	0.2 (90)	0.2 (90)	1.2	1.2
Step 6	1.1 (90)	0.6 (90)	1.3	0.6
Total	3.9	5.6	6.2	5.2
No grip change	3 (270)	1.25 (270)	1.7	1.7

Table 3.3: Grip change cycle times for all robots and custom joint angles for Epson robots

Table 3.3 compares cycle times of grip changing for all four robots. Again, time needed for gripper opening and gripper closure is not included.

Last line in the table lists cycle times for the equivalent operation with no grip changing. Values listed are equal to the sum of values for Calibrating (1) and Calibrating (2) from table 3.2.

Cycle times for both Epson robots differs significantly. As already explained in section 3.5.2, this is due to different vertical placement of the robots and were they placed equivalently, they should achieve the same results.

Without knowing the parameters of the Epson robot last custom joint, the most suitable robot for this operation cannot be determined. It can be

observed, that whereas in the whole task the Kuka sixx performed 0.6 second worse than Kuka five, it is whole 1 second faster in this operation.

■ 3.5.4 Mechanical work optimization

During the previous planning experiments, the optimization criteria was minimal path length. For those, it was sufficient to describe the robotic cell geometrically. However for planning with other optimization criteria it is necessary to know physical properties and dynamic parameters of the robot. When optimizing for minimal mechanical work, the additional parameters required are:

- Weight of each link
- Location of center of mass for each link
- Moments of inertia for each link

These data are not publicly available. It was attempted to obtain these data from manufacturers, but our requests were not satisfied. Thus, in this section, a method of rough estimation of these parameters is shown on Kuka KR Agilus sixx R1100 robot. This estimate is based on the knowledge of total weight of the robot and its CAD model. The procedure described wasn't feasible for any of the Epson robots, as their CAD models does not allow performing of the following measurements.

■ Parameter estimation

Thanks to the robot documentation, the total weight m of the robot is known:

- $m = 55.0kg$

The total volume of the robot was determined using Autodesk Inventor[33].

- $V = 0.0342m^3$

Total weight of the robot was then split between all links proportionally to their volume. This doesn't relate to the gripper, as its weight is known from the documentation.

Knowing the weight of each link, it is possible to estimate location of the center of gravity and moments of inertia for each link. A simplistic assumption, that all of the links are not hollow and that their mass is evenly distributed, has been adopted.

Values listed in 3.5 are elements of the inertia tensor,

$$[I] = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \quad (3.2)$$

Because of easier implementation in ROS, these values are calculated with respect to the link origins defined in robot CAD drawing and include the information about center of gravity locations.

Robot link	$V[m^3]$	$m[kg]$
Base	0.0105	16.8
Link 1	0.00627	10.0
Link 2	0.0101	16.2
Link 3	0.00287	4.62
Link 4	0.00402	6.47
Link 5	0.000442	0.711
Link 6	0.0000140	0.0225
Gripper	0.0000341	0.11

Table 3.4: Kuka sixx - physical parameters of links - estimate

Link	$I_{xx}[kg \cdot m^2]$	$I_{xy}[kg \cdot m^2]$	$I_{yy}[kg \cdot m^2]$	$I_{xz}[kg \cdot m^2]$	$I_{yz}[kg \cdot m^2]$	$I_{zz}[kg \cdot m^2]$
Base	0.312	-0.000195	0.411	0.0809	0.0000657	0.186
Link 1	1.24	-0.000146	1.19	-0.0353	-0.00761	0,0912
Link 2	7.81	0.00140	7.78	-0.176	0.0459	0.0767
Link 3	4.37	-0.00000359	4.38	-0.186	-0.000112	0.0236
Link 4	6.42	0.00101	7.18	-2.11	0.00229	0.774
Link 5	0.704	-0.000729	0.924	-0.393	-0.00135	0.221
Link 6	0.0223	0.00	0.0307	-0.0137	0.00	0.00844
Gripper	0.000190	-0.00000245	0.00000998	0.0000000548	0.0000000935	0.000194

Table 3.5: Kuka sixx - estimated inertial properties

Simulation results

Cycle times for mechanical work optimization for Kuka sixx robot are listed in table 3.6.

Goal position (gripper no.)	Cycle time [s]
Cleaning (1)	3.1
Forming (2)	3.7
Forming (1)	0.5
Calibrating (1)	1.7
Calibrating (2)	0.5
Camera check (1)	2.3
Plane check (1)	5.5
Container (1)	1.9
Total	19.2

Table 3.6: Kuka sixx cycle times - minimal work optimization

Total cycle time of 19.2 seconds is much worse than the cycle time obtained with path length optimization criteria for the same robot (12.6 seconds). This kind of planning could be advantageous for manipulation with heavy loads, close to the robot's limits, as it would reduce the wear of the robot and energy demands. However, in the simulated task the total weight of the dual gripper and product is less than 0.3 kg for Kuka robots (payload

up to 10 kg) and less than 0.5 kg for the Epson robots (payload up to 20 kg). Therefore, the load is rather negligible and mechanical work optimization is not demanded.

3.5.5 Manual planning

The task was also planned semi-manually, to assess the quality of the automatically planned trajectories. As the manual planning is much more time-consuming and it wasn't primary goal to solve the task traditionally, it was carried out only for Epson G20-A01SR and Kuka sixx.

During the planning process, a set of waypoints (typically one in front of each processing position such as forming or calibration) was defined. Motions from these waypoints to the positions and back again were planned automatically, as it is necessary to define the operations (picking and placing) by Cartesian paths.

Motions between the individual waypoints were executed while using interpolation in joint coordinates. Thus the Cartesian path of end effector is not predictable and the plan has to be subsequently checked for collisions. Motion based on interpolation in joint coordinates typically allows higher speeds and accelerations than interpolation in Cartesian coordinates.

Simulation results

Goal position (gripper no.)	Epson cycle time [s] (CJA [°])	Kuka sixx cycle time [s]
Cleaning (1)	4.7 (0)	1,4
Forming (2)	0.9 (90)	1.7
Forming (1)	1.7 (180)	0.5
Calibrating (1)	0.2 (90)	2.1
Calibrating (2)	0.3 (180)	0.5
Camera check (1)	0.9 (90)	1.7
Plane check (1)	1.1 (0)	1.1
Container (1)	2.1 (90)	2.1
Total	11.9	11.1

Table 3.7: Manual planning cycle times

Total cycle times are significantly better than the cycle times obtained in section 3.5.2 for the same robots. Possible reasons for this are discussed in chapter 4.1.

3.6 Real cycle times determination

As was explained in section 3.5.1, some of the operations present in the manufacturing process were not included in the cycle times stated, as their

values are unknown and they are common for all the plans generated in the same section. The operations omitted are:

- Gripper opening
- Gripper closure
- Camera checking
- Plane checking

Real cycle t_{real} time can be determined as

$$t_{real} = \sum_{i=1}^8 t_{mo_i} + t_{cc} + t_{pc} + 3 \cdot t_{go} + 3 \cdot t_{gc}, \quad (3.3)$$

where

- t_{mo_i} . . . time needed for i-th manipulation operation (e.g. Cleaning (1), Forming (2) etc.)
- t_{cc} . . . time needed for camera checking
- t_{pc} . . . time needed for plane checking
- t_{go} . . . time needed for gripper opening
- t_{gc} . . . time needed for gripper closure

For the workflow variant with grip changing, the formula is as follows:

$$t_{real} = \sum_{i=1}^8 t_{mo_i} + t_{cc} + t_{pc} + 5 \cdot t_{go} + 5 \cdot t_{gc}. \quad (3.4)$$

Chapter 4

Conclusion

4.1 Results discussion

Data obtained in simulations carried out lead to the following conclusions:

Choice of robot

The most suitable data for comparing all robots are in table 3.2, which lists cycle times for the whole task, while using RRT star algorithm with minimal path length optimization criteria.

According to these results, Kuka five robot achieved the best cycle time of 12.0 seconds. However, as explained in section 3.5.2, the task definition had to be slightly modified for Kuka five, as it couldn't reach the same positions as the other three robots. Therefore, it is recommended to consider installing Kuka six instead, as it achieved very similar cycle time of 12.6 seconds and is more versatile due to its six axis structure.

Separation of simple operations - grip changing

One of the questions was whether to design custom insertion mechanism for calibration tool (details in section 3.5.3). If not, the robot would need to change grip before and after operating the calibration position - this solution would be cheaper to build, yet would take longer.

Cycle times for grip changing are listed in table 3.3. Here, final recommendation can't be given, because the best cycle times were achieved by Epson robots and these may still be influenced by the last custom joint. Choice of this joint is up to Modia.

For Kuka robots, the extra motions needed for this operation would take 3 (Kuka six), respectively 4 (Kuka five) extra seconds (in addition to 1.7 seconds for the simple variant).

4.2 Comparison with standard planning

In section 3.5.5, the robot motion was planned semi-manually for Epson G20A01-SR and Kuka six and cycle times were determined. It turned



Bibliography

- [1] Htc vive picture source. http://media.bestofmicro.com/Y/I/571482/gallery/14-vive-parts-Edit-developed-fixed2_w_600.jpg.
- [2] Primesense sensor picture source. <http://photos2.appleinsider.com/13.07.16-Carmine.jpg>.
- [3] Vicon mocap picture source. http://www.cgadvertising.com/media/NEWS/2/IMG_0814B.jpg.
- [4] Xsens suit picture source. http://www.edinburgh-robotics.org/sites/default/files/mvn_biomech-430x484.jpg.
- [5] Modia. <http://www.modia.cz/cs>, (accessed April 28, 2017).
- [6] Convergent information technologies. <http://convergent-it.at/drupal/>, (accessed December 13, 2016).
- [7] Industrial path solutions. <http://industrialpathsolutions.se/#software>, (accessed December 13, 2016).
- [8] Siemens kineoworks. https://www.plm.automation.siemens.com/en_us/products/open/kineo/kineoworks/, (accessed December 13, 2016).
- [9] Siemens product lifecycle management. <https://www.plm.automation.siemens.com/>, (accessed December 13, 2016).
- [10] Manuserv project. <http://www.manuserv.de/manuserv/>, (accessed December 17, 2016).
- [11] article about mottoman simulation tool. <https://www.yaskawa.eu.com/en/news-events/news/article/news/simulation-and-offline-programming-of-motoman-robot-systems/>, (accessed December 19, 2016).
- [12] Factory automation company. <http://www.fanuc.eu/>, (accessed December 19, 2016).

- [13] Industrial robotics simulation. <http://industrial-robotics.co.uk/simulations.htm>, (accessed December 19, 2016).
- [14] Kuka simulation software portfolio. <http://www.kuka-robotics.com/cs/products/software/simulation/>, (accessed December 19, 2016).
- [15] Motion controls robotics. <http://motioncontrolsrobotics.com/robotic-simulations/>, (accessed December 19, 2016).
- [16] Octopuz simulation software. <http://octopuz.com/>, (accessed December 19, 2016).
- [17] Robot simulation software overview. <https://www.smashingrobotics.com/most-advanced-and-used-robotics-simulation-software/>, (accessed December 19, 2016).
- [18] Asus xtion. https://www.asus.com/3D-Sensor/Xtion_PRO/, (accessed December 20, 2016).
- [19] Example of motion capture markers. <http://www.qualisys.com/accessories/passive-markers/lightweight-marker/>, (accessed December 20, 2016).
- [20] Htc vive, virtual reality headset. <https://www.vive.com/eu/>, (accessed December 20, 2016).
- [21] Microsoft kinect. <http://www.xbox.com/en-US/xbox-360/accessories/kinect>, (accessed December 20, 2016).
- [22] Primesense. <https://en.wikipedia.org/wiki/PrimeSense>, (accessed December 20, 2016).
- [23] Vicon motion capture. <https://www.vicon.com/>, (accessed December 20, 2016).
- [24] xsens motion capture - imus. <https://www.xsens.com/>, (accessed December 20, 2016).
- [25] 3d automatic path planning for surface machining, ros-industrial library. <http://rosindustrial.org/news/2015/8/2/3d-automatic-path-planning-for-surface-machining>, (accessed December 9, 2016).
- [26] Descartes planning library. <http://wiki.ros.org/descartes>, (accessed December 9, 2016).
- [27] Moveit! library. <http://moveit.ros.org/>, (accessed December 9, 2016).
- [28] Open motion planning library. <http://ompl.kavrakilab.org/>, (accessed December 9, 2016).

- [29] Robot operating system. <http://www.ros.org/>, (accessed December 9, 2016).
- [30] Ros-industrial. <http://rosindustrial.org/>, (accessed December 9, 2016).
- [31] Ros-industrial scan-n-plan. <http://rosindustrial.org/scan-n-plan>, (accessed December 9, 2016).
- [32] Rviz 3d visualization tool. <http://wiki.ros.org/rviz>, (accessed December 9, 2016).
- [33] Autodesk inventor. <https://www.autodesk.cz/products/inventor/overview>, (accessed May 10, 2017).
- [34] Kuka fivve r1100. <https://www.kuka.com/en-us/products/robotics-systems/industrial-robots/kr-agilus-fivve>, (accessed May 2, 2017).
- [35] Kuka sixx r1100. <https://www.kuka.com/en-cn/products/robotics-systems/industrial-robots/kr-agilus-sixx>, (accessed May 2, 2017).
- [36] Denavit-hartenberg parameters. https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters, (accessed May 7, 2017).
- [37] Epson rc+ 7.0 simulator. <http://robots.epson.com/software-support>, (accessed May 7, 2017).
- [38] Epson scara robots, g20 series. <http://robots.epson.com/product-detail/5>, (accessed May 7, 2017).
- [39] Rrt algorithm - ompl class reference. http://ompl.kavrakilab.org/classompl_1_1geometric_1_1RRT.html, (accessed May 7, 2017).
- [40] Rrt expansion picture source. <http://mappingignorance.org/fx/media/2013/02/Blanco1.png>, (accessed May 7, 2017).
- [41] Scara structure description. <https://en.wikipedia.org/wiki/SCARA>, (accessed May 7, 2017).
- [42] Schunk egp-25-n-n-b gripper. https://de.schunk.com/de_en/gripping-systems/product/39758-0310900-egp-25-n-n-b/, (accessed May 7, 2017).
- [43] Srdf description. http://picknik.io/moveit_wiki/index.php?title=SRDF, (accessed May 7, 2017).
- [44] Rrt star algorithm - ompl class reference. http://ompl.kavrakilab.org/classompl_1_1geometric_1_1RRTstar.html, (accessed May 8, 2017).

- [45] Rrt vs. rrt star expansion comparison. https://youtu.be/JeEk_CWcRFI, (accessed May 8, 2017).



Appendix A

Content of the attached CD

Filename	Description
./Epson	Epson RC+ 7.0 simulator programs
./Kuka	Kuka.Sim Pro simulator programs
./ROS	ROS workspace - .urdf and .srdf files
./trajectories data	raw trajectories data exported from ROS
david-woller-bachelor-thesis.pdf	bachelor project documentation

Table A.1: Content of the attached CD