



Norwegian University of
Science and Technology

Robot Control in Image-Guided Intervention

Andreas Østvik

Master of Science in Physics and Mathematics

Submission date: June 2016

Supervisor: Catharina de Lange Davies, IFY

Co-supervisor: Thomas Langø, SINTEF

Norwegian University of Science and Technology
Department of Physics

Abstract

Introducing autonomous robot systems in clinical medicine is deemed extremely challenging due to the complex scene involved and variations between patients. The majority of commercialized systems are controlled directly in a local telesurgical mode, requiring a high degree of human interaction. Research on robot control, vision and image-guided interventions has facilitated new integration possibilities, with potential of increased autonomy in multiple stages of patient care.

The work presented in this thesis is devoted to research and development of a framework for robot manipulation integrated with existing methods used in image-guided interventions. The framework employs the research platform CustusX, which is an open-source navigation software with implemented functionality aimed towards interventional use. Both CustusX and the robot framework is written in the programming language C++ utilizing several external libraries. The systems are well integrated creating a original nexus for robot manipulation and state of the art solutions for image-guided interventions.

Automatic calibration routines for spatially relating the robot manipulator with existing tools used for image-guided interventions is developed. The positional accuracy is in the sub-millimetre range, making it suitable for several clinical applications. Furthermore, the integrated framework offers a range of functionalities and extension points. This includes robot manipulation based on external input sources such as physical pointing instruments, but also interaction with patients registered with preoperative data using an implemented user interface. In addition, a vision-based robot control system using ultrasound is developed, to allow autonomous robot motion interpreting information obtained from image analysis. To investigate the performance of the implementations, several verification experiments are conducted with the robot manipulator UR5 from Universal Robots, together with other tools present in a typical operating scene. Evaluations have demonstrated that the integrated robot system has satisfying performance with extension possibilities towards clinical applications.

Sammendrag

Bruk av autonome robotsystemer i klinisk medisin anses å være utfordrende med tanke på operasjonsscenens kompleksitet, og variasjoner mellom pasienter. Majoriteten av kommersialiserte systemer kontrolleres direkte i et lokalt telekirurgisk modus, og krever høy grad av menneskelig interaksjon. Forskning på robotkontroll, -visjon og bildeveiledet intervensjon har lagt til rette for nye integreringsmuligheter, og økt autonomi i flere ledd av pasientforløpet.

Arbeidet i denne masteroppgaven er dedikert til forskning og utvikling av et rammeverk for robotstyring integrert med eksisterende løsninger for bildeveiledet intervensjon. Rammeverket benytter forskningsplattformen CustusX, som er et navigasjonssystem med åpen kildekode implementert med funksjonalitet rettet mot intervensjonell bruk. Både CustusX og robotrammeverket er skrevet i programmeringsspråket C++, og benytter flere eksterne biblioteker. Systemene er godt integrert, og danner et originalt bindeledd for robotstyring og “state of the art” løsninger for bildeveiledet intervensjon.

Automatiske kalibreringsrutiner er utviklet for å danne romlige relasjoner mellom robotarmen og eksisterende instrumenter brukt i bildeveiledet intervensjon. Den posisjonelle feilen har en størrelsesorden i submillimeter intervallet, noe som gjør systemet godt egnet for flere kliniske applikasjoner. Videre tilbyr det integrerte rammeverket en rekke funksjonaliteter og utvidelsesmuligheter. Dette inkluderer robotstyring basert på eksterne inngangskilder, slik som fysiske pekerinstrumenter, samt interaksjon med pasienter som er registrert med preoperativ data i programvarens brukergrensesnitt. I tillegg er et system for visjonsbasert robotstyring ved hjelp av ultralyd utviklet, noe som tilbyr autonom robotbevegelse ved å tolke informasjon hentet fra bildeanalyse. For å undersøke ytelsen av implementeringen ble det gjennomført flere verifikasjonseksperimenter med robotarmen UR5 fra Universal Robots, sammen med andre instrumenter typisk brukt i en operasjonsscene. Evalueringen har demonstrert at det integrerte rammeverket har tilfredstillende ytelse, samt utvidelsesmuligheter rettet mot kliniske applikasjoner.

Preface

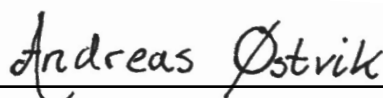
This paper is submitted as a master thesis in Biophysics and Medical technology and concludes my master's degree within the Applied Physics and Mathematics programme at the Norwegian University of Technology and Science.

Although I stand alone as the author, I do not consider this project a result of my sole contribution. First of all, a great thanks and respect goes to my supervisor Catharina de Lange Davies for her administrative contributions related to this project. I also would like to express my sincere gratitude to the knowledgeable scientists at SINTEF Dept. of Medical Technology. My stay at the department has been very pleasant, and I am privileged to have had the opportunity to work alongside and learn from such dedicated and skilled colleagues. I am indebted to the software developers, especially Ole Vegard Solberg and Christian Askeland for helping me with all sorts of computer related issues. Together they have introduced me to CustusX and software development as a whole. Thanks to Erlend Fagertun Hofstad and Daniel Høyer Iversen for our many discussions on robotics in medical research, and for insightful suggestions throughout the course of the year. If it had not been for Lars-Eirik Bø introducing me to Sketchup and providing some of the models used, quality of presented figures would probably be a lot worse - thank you.

Tremendous thanks are due to Tormod Selbekk and my supervisor Thomas Langø for giving me this chance and integrating me into the research environment at SINTEF. As a student, being invited to strategic meetings as well as being included in project planning with direct implications from this work has been very encouraging. Not to forget, your guidance and unrelenting support throughout the project has been invaluable.

Finally, I want to thank all my friends, especially at the study programme, for five fantastic years. This thesis would not be possible without them and constant support from my family. My dearest Oda Cathrine and Anna, for your unconditional love and understanding, I dedicate this thesis to you.

Trondheim, June 30, 2016

A handwritten signature in black ink that reads "Andreas Østvik". The signature is written in a cursive style with a prominent loop for the letter 'A' and a long horizontal stroke at the end.

Andreas Østvik

Contents

1	Introduction	1
1.1	Goal of the project	2
1.2	Outline	2
2	Background	3
2.1	Relevant work	3
2.2	Mathematical prerequisites	6
2.3	Serial manipulators	9
2.4	Calibration methods	18
2.5	Medical imaging	21
2.6	Image processing and analysis	23
2.7	Software	27
3	Methodology	31
3.1	Overview of the concept development	31
3.2	Relating objects of interest	32
3.3	Calibration procedure	35
3.4	Semi-automatic circle detection in US images	37
3.5	Integrated robot arm manipulation	44
3.6	Visualization	47
3.7	Experimental validation	48
4	Results	53
4.1	Calibration	53
4.2	Image processing and analysis	56
4.3	Robot manipulation	59
4.4	The integrated framework	63
5	Discussion	65
5.1	Registration and calibration	65
5.2	Image processing and analysis	69
5.3	Integration of robot manipulation	71
5.4	Robot framework	73

5.5	Future work	74
6	Conclusion	77
	Bibliography	79
A	Digital appendix	85
B	Definitions	87
	B.1 The skew-symmetric matrix	87
	B.2 Basic transformation matrices	87
C	UR5 Specifications	89
	C.1 Robot state messages	89
	C.2 Homogenous transformation matrix	89
	C.3 The Jacobian	90

Chapter 1

Introduction

Combination of increasing life expectancies and declining fertility has resulted in an ageing population. Disregarding greater longevity, the global median age is estimated to increase by nearly ten years before 2050 [1]. The median age in Western Europe is indicated to increase to 48 years. Accordingly, the proportion of the same selection aged over 60 will double during the period. This is likely to increase the demand of medical procedures [2], and as of now the accompanied challenges are yet to be resolved.

First use of medical robots date back to the mid-eighties [3], and despite being a relatively young field it has already evolved into an established part of clinical surgery. The majority of current commercialised systems are controlled directly by the operator in a local telesurgical mode [4], lacking autonomy and thus requiring a high degree of human intervention. The use of autonomous robot systems clinically is by many deemed extremely challenging due to the complex scene involved and variations between patients. Research within robotic vision and image-guided intervention has facilitated new integration possibilities, and will hopefully increase the autonomy in the operation scene. Creating sophisticated robot systems could potentially offer distinct advantages in several medical procedures, especially where great precision and repeatability is required. Among many, this includes guided needle insertions, standardized and user-independent ultrasound scans, therapeutic ultrasound, radiosurgical procedures, but also more complex soft-tissue surgeries.

Clinical feasibility is questionable, but in recent years collaborative industrial robots are becoming lighter and more flexible, which consequently would result in a smaller footprint in a potential clinical environment. In addition, trends in the price marked are making them economically viable [5]. Clinical routine can be improved by simplifying the clinical work flow using robot arms, as well as the workload of the surgeons and health personnel in general. With the presented forecast on demographic change, this could undoubtedly prove beneficial in the years to come.

1.1 Goal of the project

This master project is a sequel to the work conducted on the robotic framework in the research platform CustusX, designed and implemented by the author autumn 2015. The goal is to continue the development emphasising potential clinical applications. This includes deriving methods and setting up a laboratory scene for relating robot tools, such as medical instruments to an arbitrary domain of interest. A desired functionality is to manipulate the robot tool based on position and orientation data obtained through external sources such as pointing instruments, medical images or patient data registered to a laboratory scene. Furthermore, algorithms for following moving targets in those scenarios should be derived and verified. To relate the work to current research and solutions on the topic of autonomous robotic surgical systems, some relevant literature is presented initially.

1.2 Outline

This master thesis is organised in six chapters excluding appendices. The following outlines the remainder.

Chapter 2 - Background. Covers the theoretical prerequisites and the main technologies used in the system development. This should provide the unfamiliar reader the capability of understanding the task at hand, and the work presented in the following chapters. To relate this work in a broader context, relevant literature are also presented introductory to the best of the authors knowledge.

Chapter 3 - Methodology. Details methods used to relate objects of interest to the robot arm including proposed calibration routines. The image processing and integrated robot manipulation is also described. Essential implemented algorithms will be presented using pseudo codes with reference to the theoretical background. Lastly a description of the verification experiments will be given.

Chapter 4 - Results. Presents the results from the implementations and summarised status of the robot framework. In addition, results from experiments verifying the calibration method, image segmentation and robot manipulation is given.

Chapter 5 - Discussion. Discusses the extended framework, as well as the verification experiments. Considering the systems performance and overall precision, a outlook of the proposed clinical applications will be presented with suggestions for future work.

Chapter 6 - Conclusion. Concludes the project with closing remarks.

Chapter 2

Background

As presented in the previous chapter, one of the goals of this master project was to conduct a preliminary survey of relevant literature on state of the art solutions for medical robot systems. This is presented in the beginning of this chapter. A theoretical background introducing the basics and terminology used throughout the thesis will be presented in the following sections. Some of the background, especially the mathematical prerequisites, the section about serial manipulators and the software CustusX, is partially adapted from the authors specialisation project [6]. The chapter is concluded with an introduction to the framework for robot control implemented during the same project.

2.1 Relevant work

Initial integration of medical images with robot manipulators was conducted motivated by replacing the manual stereotaxic frame used in brain surgery with a robot arm guided by computed tomography(CT) images holding an instrument-guide for biopsy needle insertion [7]. The developed system was later used successfully on a patient for CT-guided brain tumor biopsy [8]. Initially, the accuracy was inadequate for small lesions and critical areas of the brain, but this research has arguably facilitated further development leading to present solutions.

Today, the most renowned surgical robot is Intuitive Surgical's da Vinci System [9], being discussed in thousands of publications [3]. The da Vinci is a surgical assistant directly controlled by a physician. The core functionality is augmenting the physicians ability of hand-movement by motion scaling and tremor reduction, providing the surgeons with better precision and enhanced dexterity during minimal invasive procedures. Most of the commercial available surgical robots, like the da Vinci, lack sophisticated autonomy being controlled directly by the surgeon. In this thesis this is of interest, thus will the remainder of this section concern a selection of state of the art solutions and research towards systems with a higher level of autonomy than the prior. Integration with medical multi-modal imaging in form

of computed tomography (CT), magnetic resonance imaging (MRI) and ultrasound (US) will be given prominence, with the latter being the main focus.

The future of autonomous robotic systems for surgical procedures is believed enabled by integration of robot responses based on analysis of real-time ultrasound [10] and preoperative modality images, such as MRI and CT. Combining ultrasound with robots for medical interventions, so called robotic ultrasound systems (RUS), has received increasing attention the past decade as it has the potential of enabling real-time visual servoing of the robot manipulator with respect to the interior of the body. Visual servoing is a technique utilizing vision based sensors and analysis of obtained data to control a robot, potentially by an automatic response. The most comprehensive research towards visual servoing using medical ultrasound is conducted at the National Institute for Research in Computer Science and Control (INRIA) and the Research Institute in Computer Science and Random Systems (IRISA) in France. In association, Krupa *et al.* has published several articles on interpreting robot motion response based on information obtained through ultrasound acquisition [11, 12]. They also presented a robotized tele-echography system with several modes such as autonomous scanning and positioning, including the possibility to maintain visibility of defined structures [13].

A common issue frequently noted for methods using standard 2D probes is tracking motion perpendicular to the two-dimensional ultrasound imaging plane. Mentioned publications rely on intensity based analysis, but they tend to lack robustness for so called out-of-plane motion. The concept of tissue speckle tracking is suggested as an alternative solution [14]. The presence of speckle noise is a typical characteristic of ultrasound images. Speckle is deterministic interference formed from coherent reflections from scatterers in the sub-resolution volume [15]. For small motions the speckle pattern is highly correlated, and this is what the authors base their method on. Recently a novel method using control laws employed with Shearlet coefficients shows great promise for determining featureless out-of-plane motion [16]. Among several additional solutions, utilizing 3D probes is an obvious way to cope with the problem. However, compared to 2D probes the frame rate is much lower making real-time tracking modes fragile [17]. Increasing the computational requirements by introducing more data through image volumes will also extend the calculation time, thus delaying estimates used for positioning. Nadeau *et al.* suggests using special probes [18], such as the bi-plane type, to solve both the out-of-plane motion and the frame rate problem.

A recurring pattern for developed systems and methodology is that they rely on implementing their own confined and task specific user interfaces. This makes exploitation with existing image-guided therapy solutions dreary, as extensions need to be created from scratch. Integrating robot manipulation into mature image-guided therapy platforms is believed advantageous as they generally provide basic layers already adopted for medical procedures. The renowned open source visual-

ization and medical image computing software 3D Slicer [19] is such a platform, and consequently it has been implemented with robot manipulation. Xia *et al.* has implemented collaborative control and visualization of a robot arm in the software for skull base drilling [20]. A more general contribution using OpenIGTLink [21] together with 3D slicer for state control and visualization of a robot is presented by Tauscher *et al.* [22]. Evolved into an extension module in the platform known as LightWeightRobotIGT¹.

Earlier, solutions for robot manipulation in IGT used ad hoc solutions, such as 3D printer setups or in-house robots. Such setups are often non-applicable to a clinical environment. Anyhow, as the publications above suggest, more recent research favors collaborative lightweight robots from large commercial firms. Collaborative, in this context, means a system able to perform tasks interacting with humans. These types of robots share common characteristics, such as restricted payloads and other safety-related specifications. Most also have built-in security functionality with ISO approval (ISO 10218-1:2011, ISO 13849-1:2015) [23, 24]. Currently a specific standard (IEC/NP 80601-2-77) for basic safety and essential performance of medical robots for use in surgery is under development [25]. The robot manipulator used in this project, the UR5 from Universal Robots² (Odense, Denmark), has not received much attention in the IGT community yet, but are well established in other industry and research. However, researchers at Johns Hopkins University have published several studies that use the robot in a robot-assisted tomography ultrasound system for deeper and faster scans [26, 27], while Bø *et al.* has used it for probe calibration [28] used in ultrasound-guided interventions. Recently, Mathiassen *et al.* presented a ultrasound robotic system using the manipulator holding an ultrasound probe [29]. The system has a sophisticated haptic sensor system with user interaction using a spatial navigator. Although the system relies on full human interaction rather than autonomous solutions, it proves the manipulator can be feasible for this type of tasks.

Though not completely related as it does not involve the mentioned modalities, first reports of an autonomous robot system successfully suturing and reconnecting *in vivo* pig intestines was announced and published early May this year [30]. The system is named Smart Tissue Autonomous Robot (STAR) and utilizes an industrial collaborative robot similar to the one used in this project, with seven degrees of freedom extended with a sewing tool. The vision system is based on a combination of Near Infrared Fluorescence (NIRF) molecular imaging and three-dimensional quantitative plenoptic imaging. Remarkably, it outperforms human experts performing a procedure known as intestinal anastomosis. Perhaps this preclinical evaluation could be the first phase in a new era of clinical practice where supervised autonomy

¹<https://www.slicer.org/slicerWiki/index.php/Documentation/4.5/Extensions/LightWeightRobotIGT>

²www.universal-robots.com

is admitted to soft-tissue surgery.

2.2 Mathematical prerequisites

This section is a short review of the mathematical background needed. The reader is assumed to be familiar with basic linear algebra, especially matrix operations.

2.2.1 Relating objects in space

To describe the position and orientation of rigid bodies a normal convention is to consider a orthonormal coordinate frame attached to the body of interest, and express it with respect to some reference frame. A pose is a notion of describing the translation and the orientation of a coordinate frame with respect to another [31], and can be used to transform vectors between coordinate frames. This is useful when considering multiple coordinate frames and relating them with a common reference. There are several ways to represent a pose, and in this text the concept of homogeneous transformation will be used primarily.

The position of a point P in the three dimensional space can be described by a translation vector $\mathbf{p}^i \in \mathbb{R}^3$ with respect to an arbitrary coordinate frame i . A translation vector between the origin of a coordinate frame j with respect to the reference frame i can be denoted by $\mathbf{t}_j^i \in \mathbb{R}^3$. To fully describe a rigid body in space the rotation must also be accounted.

Rotation matrix

In three dimensions the rotation matrices are (3×3) matrices of the special orthogonal group $SO(3)$. A matrix of this type can be used to describe the orientation of an arbitrary frame with respect to some reference, or to transform coordinates of a point from one frame to another. A general rotation matrix of frame j with respect to frame i is given by

$$\mathbf{R}_j^i = \begin{bmatrix} x_j \cdot x_i & y_j \cdot x_i & z_j \cdot x_i \\ x_j \cdot y_i & y_j \cdot y_i & z_j \cdot y_i \\ x_j \cdot z_i & y_j \cdot z_i & z_j \cdot z_i \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. \quad (2.1)$$

Matrices in the special orthogonal group $SO(3)$ are orthogonal with determinant equal to unity. Another useful property of the group is that $\mathbf{R}^{-1} = \mathbf{R}^T$, which in many cases simplifies calculations significantly.

Axis-angle representation

Another common rotation formalism in three dimensions is a result of the Euler rotation theorem [32], namely axis-angle representation. Two arbitrary and independent orthonormal coordinate frames in space can be related through a simple

rotation by an angle v about the axis \mathbf{k} . As stated above, the rotation matrix is orthonormal, and hence its determinant is equal to unity. This implies that one of the eigenvalues is equal to one, and that the corresponding unit vector \mathbf{k} satisfies $R_j^i \mathbf{k} = \mathbf{k}$.

One way to find the rotation matrix from the axis-angle parameters is using the Rodrigues formula [33], which is given by

$$\mathbf{R}_j^i = \cos v \mathbf{I} + \sin v \mathbf{S}(\mathbf{k}) + (1 - \cos v) \mathbf{k} \mathbf{k}^T, \quad (2.2)$$

where $\mathbf{S}(\mathbf{k})$ is the skew symmetric matrix. See appendix B for a definition of $\mathbf{S}(\mathbf{k})$. Inserting $\mathbf{k} = [k_x, k_y, k_z]^T$ into equation (2.2) gives

$$\mathbf{R}_{\mathbf{k},v} = \begin{bmatrix} k_x^2(1-c_v) + c_v & k_x k_y(1-c_v) - k_z s_v & k_x k_z(1-c_v) + k_y s_v \\ k_x k_y(1-c_v) + k_z s_v & k_y^2(1-c_v) - c_v & k_y k_z(1-c_v) + k_x s_v \\ k_x k_z(1-c_v) + k_y s_v & k_y k_z(1-c_v) + k_x s_v & k_z^2(1-c_v) + c_v \end{bmatrix}.$$

Here a prevalent abbreviation of the trigonometric functions sine and cosine is used, i.e $\cos v = c_v$ and $\sin v = s_v$. It is noticeable that $\mathbf{R}_{\mathbf{k},v} = \mathbf{R}_{-\mathbf{k},-v}$, thus the representation is not unique.

The inverse operation is also of interest. Given an arbitrary rotation matrix \mathbf{R}_j^i as in equation (2.1), the corresponding angle v are given by

$$v = \arccos\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right). \quad (2.3)$$

The unit vector \mathbf{k} , namely the axis of rotation, is given by

$$\mathbf{k} = \frac{1}{2 \sin v} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}, \quad (2.4)$$

and noticeably equation (2.3) and (2.4) become meaningless for $\sin v \neq 0$. Often used in this context is also the Euler vector, simply given as $\mathbf{v} = v \mathbf{k}$.

Homogeneous transformation

To effectuate the notation of position and orientation the homogeneous transformation is introduced by combining the concepts, and representing them in a matrix of the form

$$\mathbf{H}_j^i = \begin{bmatrix} \mathbf{R}_j^i & \mathbf{t}_j^i \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.5)$$

where $\mathbf{H}_j^i \in (\mathbb{R}^3 \times SO(3) = SE(3))$. The special Euclidean group of order three, $SE(3)$, can describe any angular and linear displacement of rigid bodies in the three

dimensional Euclidean space. Its worth noting that the homogeneous transformation matrix is not orthogonal, and thus $\mathbf{H}^{-1} \neq \mathbf{H}^T$. Using the fact that the rotation matrix \mathbf{R}_j^i is orthogonal, the inverse homogeneous transformation is given as

$$\mathbf{H}_j^{i-1} = \mathbf{H}_i^j = \begin{bmatrix} \mathbf{R}_j^{iT} & -\mathbf{R}_j^{iT} \mathbf{t}_j^i \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

A sequence of subsequent transformations can be calculated by multiplication of the homogeneous transformation matrices. The resulting matrix is then

$$\mathbf{H}_j^i = \mathbf{H}_{i+1}^i \cdots \mathbf{H}_j^{j-1}, \quad (2.6)$$

where the translation vector is

$$\mathbf{t}_j^i = \mathbf{t}_{j-1}^i + \mathbf{R}_{j-1}^i \mathbf{t}_j^{j-1},$$

and the rotation matrix is

$$\mathbf{R}_j^i = \mathbf{R}_{i+1}^i \cdots \mathbf{R}_j^{j-1}.$$

The homogeneous transformation matrix serves as a useful mathematical toolbox for the derivation of manipulator kinematics and relating objects of interest.

2.2.2 Kronecker product and vectorization

The Kronecker product is a mathematical operator mapping two arbitrary sized matrices into a block matrix [34]. The operator product, denoted as \otimes , between a matrix \mathbf{A} and \mathbf{B} with size $m \times n$ and $p \times q$ respectively is given as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11} \mathbf{B} & \cdots & a_{1n} \mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1} \mathbf{B} & \cdots & a_{mn} \mathbf{B} \end{bmatrix},$$

which is a matrix of size $mp \times nq$. Vectorization of matrices, i.e converting a matrix into a column vector by stacking each matrix column, often goes hand in hand with the Kronecker product. A useful formulations of them used together is

$$\text{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}), \quad (2.7)$$

which implies

$$\text{vec}(\mathbf{AB}) = (\mathbf{I} \otimes \mathbf{A}) \text{vec}(\mathbf{B}) = (\mathbf{B}^T \otimes \mathbf{I}) \text{vec}(\mathbf{A}).$$

As the equations suggest, the vectorization operation is used to express matrix multiplication as a linear transformation of matrices utilizing the Kronecker product.

2.2.3 Singular value decomposition

The singular value decomposition (SVD) is the factorization of a matrix into a product of three matrices [35],

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (2.8)$$

where \mathbf{U} and \mathbf{V} are orthonormal and $\mathbf{\Sigma}$ is diagonal with entries corresponding to the singular values σ_i of \mathbf{A} . SVD is more general than eigenvalue decomposition as it can be applied to arbitrary sized matrices. The matrix columns of \mathbf{U} and \mathbf{V} are called left- and right-singular vectors of \mathbf{A} , being denoted as \mathbf{u} and \mathbf{v} respectively. From the decomposition of \mathbf{A} it also follows that

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{V}^T \quad \text{and} \quad \mathbf{A}\mathbf{A}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{U}^T.$$

Thus will the vectors \mathbf{u} and \mathbf{v} of \mathbf{A} be a set of orthonormal eigenvectors of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$, with σ_i^2 being the eigenvalues of the symmetric matrices. Post multiplication of the rightmost orthogonal matrices in each case gives

$$\mathbf{A}^T\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{\Sigma}^T\mathbf{\Sigma} \quad \text{and} \quad \mathbf{A}\mathbf{A}^T\mathbf{U} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T. \quad (2.9)$$

Overall, the idea is to take a high dimensional data set, possibly highly variable, and reduce it to substructures of lower space while ordering it from most to least variation [36].

2.3 Serial manipulators

The serial link manipulator is an industrial robot composed of rigid links interconnected by joints. Each joint represents a degree of freedom [37], and are either revolute or prismatic. The configuration, denoted by the joint variable q_i , changes the relative angles or position of adjacent links respectively. The series of links and joints extends from a base, usually stationary, to an end-effector which can move in the robots reachable workspace [38].

As mentioned, the joint variable represents the relative displacement between two adjacent links, and for rotating joints the notational convention is $q_i = \theta_i$, where θ_i is the relative rotation. The joint configuration for a manipulator with n joints is $\mathbf{q} = \{q_i, i \in [1, \dots, n]\} \in \mathcal{C}$, where \mathcal{C} is the configuration space in a subset of \mathbb{R}^n . The velocities are then $\dot{\mathbf{q}} = \{\dot{q}_i, i \in [1, \dots, n]\} \in \mathbb{R}^n$. The space of all possible positions and orientation of the end-effector is called the task space \mathcal{T} , and in the three dimensional case \mathcal{T} is a subset of the special euclidean group of order three, $SE(3)$. Throughout the report the term task space and operational space will be used interchangeably.

2.3.1 Kinematics

Kinematics, in this context, describes the motion chains forming the structure of a robot system. Robot kinematics involves the mapping from joint configuration space to the operational space and vice versa. This can also include the velocity spaces, which will be covered briefly.

Forward kinematics

The forward kinematics are used to define the end-effectors pose $\xi_e \in \mathcal{T}$, as well as preceding joint poses, based on the joint configuration $\mathbf{q} \in \mathcal{C}$. As mentioned, the sequence of subsequent transformations can be calculated by multiplication of transformation matrices. Rewriting equation (2.6) gives

$$\mathbf{H}_j^i = \begin{cases} \mathbf{H}_{i+1}^i \mathbf{H}_{i+2}^{i+1} \cdots \mathbf{H}_{j-1}^{j-2} \mathbf{H}_j^{j-1}, & \text{if } i < j \\ \mathbf{I}, & \text{if } i = j \\ (\mathbf{H}_i^j)^{-1}, & \text{if } i > j \end{cases}$$

This can be used to describe the transformations between different manipulator joints. The end-effector pose for a n joint manipulator with respect to the base frame $b = \{0\}$ can then be written as

$$\xi_e \sim \mathbf{H}_n^b = \prod_{i=1}^n \mathbf{H}_i^{i-1} = \begin{bmatrix} \mathbf{R}_n^b & \mathbf{t}_n^b \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

For complex manipulators with several joints it is hard to derive the forward kinematics. Systematic conventions like the Denavit-Hartenberg method [39] should therefore be applied. Not only because it is easier to follow a defined method, but also introducing universality enables a common language for problems concerning forward kinematics. Other conventions exist, but this report is limited to only include this one. The transformation matrix are here described by a product of four basic transformation given by

$$\begin{aligned} \mathbf{H}_i^{i-1} &= \mathbf{R}_z(\theta_i) \mathbf{T}_z(d_i) \mathbf{T}_x(a_i) \mathbf{R}_x(\alpha_i) \\ &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \end{aligned}$$

where R_i and T_i is the transformation matrix for a rotation or translation with respect to an axes i respectively. The definition of the matrices used are given in Appendix B. The parameters θ_i , d_i , a_i and α_i are the joint configuration/angle, link offset, link length and link twist, respectively, and is associated with joint i

and the subsequent link as seen in figure 2.1. It is also important to note that the axis of manipulation is assigned to be the z_i -axis.

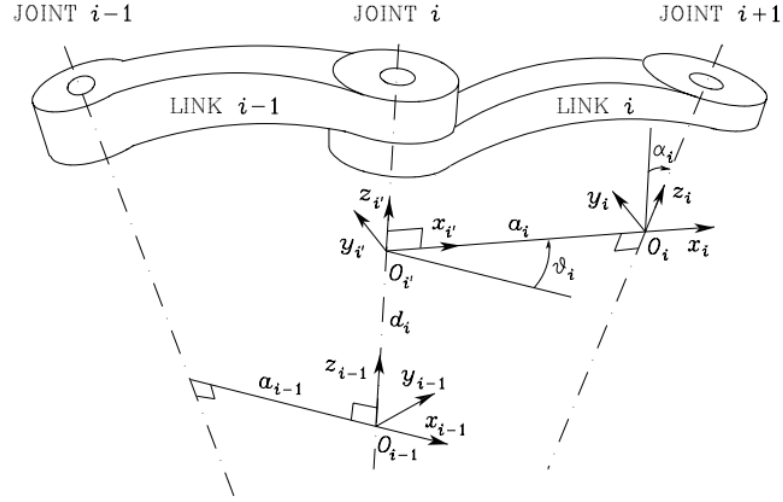


Figure 2.1: Sketch of the Denavit-Hartenberg kinematic parameters. Three revolute joints with rotation axis z is shown, with the corresponding joint configuration angle $\vartheta_i = \theta_i$. d_i and a_i is the link offset and link length respectively, while α_i denotes the link twist. Adapted from Siciliano *et al.* [37].

As mentioned, a robotic joint is either prismatic or revolute, which again means that θ_i or d_i are variables. According to this, three of the arguments are constant for every joint, and thus one can derive expressions for the forward kinematics with one type of variable. The constants are robot specific, and can be set based on physical interpretations of the robot at hand.

Velocity kinematics

In the preceding section the relationship between the joint configuration \mathbf{q} and the Cartesian positions and orientations, denoted by $\boldsymbol{\xi}$, was developed. In this section the mapping between the spatial velocity $\boldsymbol{\nu}$ and the joint configuration velocities $\dot{\mathbf{q}}$ are presented. The spatial velocity is proportional to the joint configuration velocities, and the term of relation is the Jacobian \mathbf{J} , which has a joint configuration \mathbf{q} dependency. The elegant relationship is described by

$$\dot{\boldsymbol{\xi}}_e \sim \boldsymbol{\nu} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}, \quad (2.10)$$

where $\boldsymbol{\nu}$ is the screw vector composed of the position $\dot{\mathbf{p}}$ and angular velocity $\boldsymbol{\omega}$, i.e $\boldsymbol{\nu} = \begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix}^T$. The Jacobian needs to be computed in terms of how the orientation is represented in the operational space. If we are using the homogeneous matrices, the geometric Jacobian is used. For the case when a minimal number of parameters describes the orientation, i.e axis-angle representation, we resort to the analytic Jacobian.

For a manipulator with n -joints in three dimensions the Jacobian is a $(6 \times n)$ matrix. For a revolute joint, the Jacobian matrix is calculated by

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1}^0 \times (\mathbf{t}_n^0 - \mathbf{t}_{i-1}^0) \\ \mathbf{z}_{i-1}^0 \end{bmatrix},$$

while for prismatic joints the Jacobian is given as

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1}^0 \\ 0 \end{bmatrix}.$$

The combination of all joints gives the final matrix

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \cdots & \mathbf{J}_{n-1} & \mathbf{J}_n \end{bmatrix}. \quad (2.11)$$

In case of rank deficiency, i.e. $\text{rank}(\mathbf{J}) < n$, the manipulator encounters singularities. Singularities are configurations in which mobility is reduced, and thus arbitrary motion of the end-effector is not possible. For square matrices this also means that its determinant is zero, making it non invertible, and an infinite number of solutions may exist. If not considered singularities can result in undesirable situations as small velocities in the operational space may result in large velocities in the joint space. Singularities are divided into boundary and internal singularities. As the name suggest, the prior occur when the manipulator is stretched to the outer boundary of the workspace or retracted to a locked position. The internal singularities may exist inside the reachable workspace, and are typically caused by alignment of two or more axes of motion.

2.3.2 Trajectory planning

Robot tasks, disregarded rare exceptions, involves motion of the joints. The necessity of understanding motion planning is therefore crucial. Trajectory planning involves the generation of time sequences of the desired values of motion defined by the user. In practical situations the user defines the parameters, and the controller system develops the trajectory through motion planning algorithms. Since the operational space remains more intuitive for the user in the case of planning, trajectory generation in the operational space will be the main focus here. Planning algorithms can be extended between spaces, and in general the usage are not space invariant, however the actual paths can deviate.

Though often used as synonyms, there is a difference between path and trajectory in this context. A path is the points in space in which the manipulator is assigned to follow, and is purely geometric. A trajectory is a path where a timing law is specified, and basically trajectory planning concerns the path description, but also the constraints, desired velocities and accelerations.

Operational space trajectories

Planning trajectories in the operational space not only prove more intuitive for the user but also simplifies the path constraints and determination of objects of interest inside the manipulators workspace. Some typical situations require the end-effector to move to a specific point or through sequences of points. Either way, the planning can be done in the operational space and algorithms involving inverse kinematics can be used offline before execution or real-time during motion.

Given a set of path points, the trajectory planning can be done by interpolating the sequence using cubic polynomials or commonly using linear segments with parabolic blends. A useful method is to express the motion analytically using motion primitives. This is done by representing the path of interest as a parametric representation of the arc length. The point \mathbf{p} thus corresponds to a arc length s . Each point \mathbf{p} can be utilized to express three unit vectors, namely the tangent vector \mathbf{t} , the normal vector \mathbf{n} and the binormal unit vector \mathbf{b} . The combined frame is right-handed as suggested by figure 2.2. The tangent vector \mathbf{t} is oriented along the direction of the path, together with the normal vector \mathbf{n} they constitute the tangential plane at each point \mathbf{p} . The binormal \mathbf{b} is chosen such that the frame becomes right-handed. The relations between the unit vectors can be expressed as

$$\mathbf{t} = \frac{d\mathbf{p}}{ds}, \quad \mathbf{n} = \frac{1}{\left\| \frac{d^2\mathbf{p}}{ds^2} \right\|} \frac{d^2\mathbf{p}}{ds^2}, \quad \mathbf{b} = \mathbf{t} \times \mathbf{n}.$$

Among several applications, these vectors can be used to construct rotation matrices, e. g $\mathbf{R} = [\mathbf{t} \ \mathbf{n} \ \mathbf{b}]$.

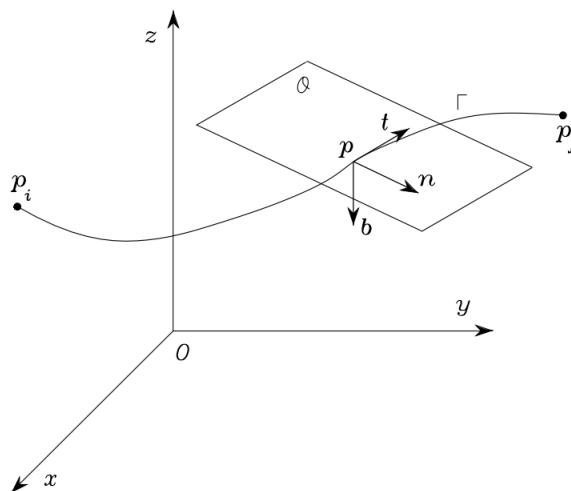


Figure 2.2: A parametric representation of an open path in the operational space. The right-handed system consisting of the tangent unit vector \mathbf{t} , the unit vector \mathbf{n} and the binormal unit vector \mathbf{b} corresponding to the point \mathbf{p} in space is illustrated. Adapted from Siciliano *et al.* [37].

Position trajectory

To generate a trajectory in the operational space in terms of position a timing law for the manipulators end-effector pose $\xi_e(t)$ must be determined. This can be done by referring to the above. The end-effector frame moves from an initial to a final point in a time $t_f - t_i$. The timing law can be assigned by the arc length function $s(t)$, and its analytical expression is based on chosen techniques, i.e cubic polynomials or sequences of linear segments with parabolic blends. For a given timing law, in terms of the arc length s , the velocity of the end-effector is given by the time derivative

$$\dot{\mathbf{p}}_e = \dot{s} \frac{d\mathbf{p}_e}{ds} = \dot{s} \mathbf{t}, \quad (2.12)$$

and varies with the chosen profile of $s(t)$ and the oscillating tangent unit vector along the path.

Orientation trajectory

The orientation of the end-effector along a trajectory are generally specified by the time-varying rotation matrix $\mathbf{R}_e(t)$. In order to characterize the end-effectors orientation along a trajectory, the transformations needed are

$$\mathbf{R}_e(t) = \mathbf{R}_i \mathbf{R}^j(t),$$

where $\mathbf{R}^j(t)$ is the transition matrix between the initial orientation \mathbf{R}_i and the final orientation \mathbf{R}_f at a given time t . In some cases the angular velocity of the end effector $\boldsymbol{\omega}_e(t)$ is needed for planning the trajectories. One way to interpret the angular velocities, as well as the angular acceleration $\dot{\boldsymbol{\omega}}_e(t)$, is by resorting to the presentation of the axis-angle representation in section 2.2.1. Using the convention, the transition matrix $\mathbf{R}^j(t)$ can then be written as $\mathbf{R}^j(v(t), \mathbf{k}^j)$. Since the axis of rotation \mathbf{k}^j is constant between every two orientations, it becomes sufficient to define a timing law for the angle of rotation v . The transitional velocity are then found by time differentiation of the rotation angle, i.e

$$\boldsymbol{\omega}^j(t) = \dot{v}(t) \mathbf{k}^j \quad (2.13)$$

The following relationships then describes the end-effector velocity as a function of time along the orientational trajectory

$$\boldsymbol{\omega}_e(t) = \mathbf{R}_i \boldsymbol{\omega}^j(t), \quad (2.14)$$

When the path and trajectory in the operational space in terms of position and orientation is given, inverse kinematics can be used to find the corresponding trajectories in joint space.

2.3.3 UR5 from Universal Robots

The UR5 manipulator is developed by the danish company Universal Robots, which is a world leading collaborative robotic technology firm. Today, the company has over 3500 UR robots installed worldwide, performing tasks in a wide range of sectors. Extensive research on applications using the manipulator is ongoing, mainly due to its specifications, cost and low threshold integration possibilities.

Universal Robots advertises the UR5 as a highly flexible robot arm. Some useful and relevant specifications for the manipulator is provided by the manufacturer and listed in their homepage [40]. The robot arm is a six degree of freedom serial link manipulator with low weight and size making it flexible in terms of placement. Additionally, the inbuilt sensor together with the controller will limit the force exerted by the arm, and therefore the manipulator can be operated with no additional safety guarding. The joints are installed with brushless servo motors and harmonic drive reducers estimated to have a long operating life. Noticeably the end-effector reach is 850 mm, while the repeatability is alleged to be 0.1 mm.

As mentioned, the UR5 manipulator has six revolute joints, and thus six degrees of freedom. This makes the end-effectors pose a subset of the special euclidean group $SE(3)$, and as a consequence all positions and orientation in its workspace are obtainable. A picture of the robot arm is shown in figure 2.3.

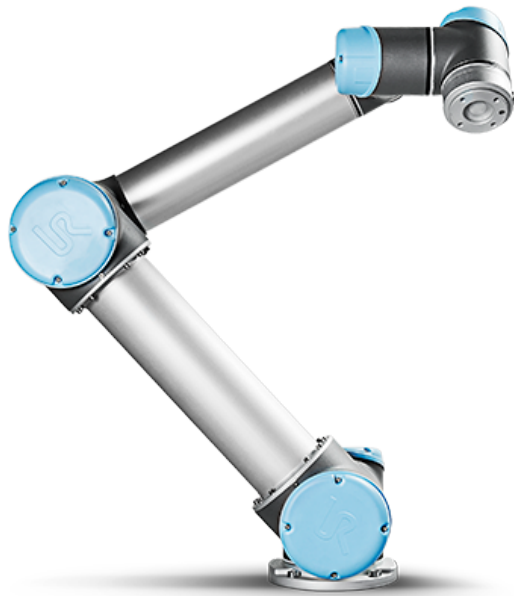


Figure 2.3: Real life picture of the UR5 manipulator in an arbitrary configuration. Adapted from Universal robotics homepage with permission. Source: <http://www.universal-robots.com/products/ur5-robot/>

The servo motors governing the joint configuration is placed behind the light blue plastic lids, while the aluminium links stretches between the motors. The end-effector is mountable, and can be extended with tools of own choice.

In order to manipulate the UR5 the forward kinematics needs to be derived. This involves the homogeneous transformation matrix describing the position and orientation of the end-effector with respect to some base reference. Relating the velocity spaces as described in section 2.3.1 requires the manipulators Jacobian, thus it is appropriately included. In figure 2.4 a geometric sketch of the manipulator with its revolute joints and assigned coordinate frames are shown.

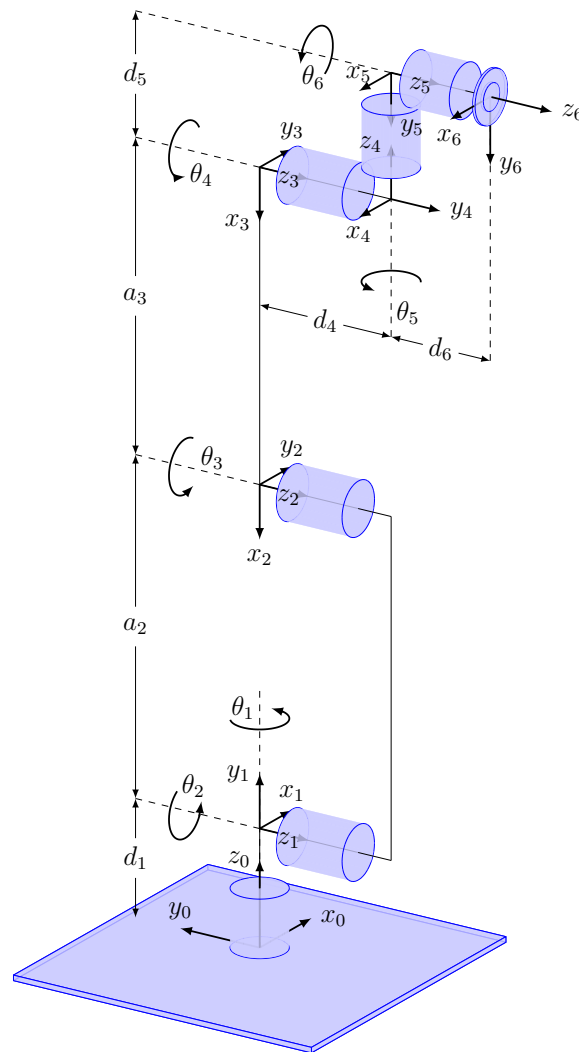


Figure 2.4: Illustration of the UR5 manipulator drawn with a joint configuration equal to $\mathbf{q}_r = [0 \ -\pi/2 \ 0 \ -\pi/2 \ 0 \ 0]^T$. The relevant Denavit-Hartenberg parameters are shown together with the coordinate frames corresponding to every joint.

The relevant Denavit-Hartenberg parameters are also illustrated, and these correspond to the values given in table 2.1.

Table 2.1: Denavit-Hartenberg parameters provided by the manufacturer. The parameter d_i represents the link offset, a_i the link length and α_i is the link twist.

Link i	d_i [mm]	a_i [mm]	α_i [rad]
1	89.16	0	$\pi/2$
2	0	-425.00	0
3	0	-392.25	0
4	109.15	0	$\pi/2$
5	94.65	0	$-\pi/2$
6	82.30	0	0

The homogeneous transformation matrices between every joint of the UR5 manipulator is calculated using the Denavit-Hartenberg convention described in section 2.3.1. Multiplying subsequent matrices from base to end-effector yields the homogeneous transformation matrix corresponding to the pose of the end-effector with respect to the robot base, generally noted as \mathbf{T}_{ee}^b , where $\{b\}$ and $\{ee\}$ corresponds to the base and end-effector respectively.

Extending the manipulator with a tool is common, and requires a description of the tool coordinate frame in a known reference. Knowing the tool transformation matrix with respect to the end-effector frame permits us to describe the tool frame with respect to the robots base. This corresponds to multiplying \mathbf{T}_{ee}^b with the transformation matrix between the end-effector frame and the tool frame \mathbf{T}_t^{ee} , i.e $\mathbf{T}_{ee}^b \mathbf{T}_t^{ee}$,. Similarly, relating the base frame in operational space to some other arbitrary frame of interest is done by pre multiplying the base frame pose with respect to the arbitrary frame. In practical consensus, this can be used to relate the manipulator to other objects of interest.

During motion the velocity of the end-effector in operational space can be related to the joint velocity by the linear relationship of equation (2.10). According to equation (2.11), the geometric Jacobian for a six revolute joint manipulator is

$$\mathbf{J} = [\mathbf{J}_1 \quad \mathbf{J}_2 \quad \mathbf{J}_3 \quad \mathbf{J}_4 \quad \mathbf{J}_5 \quad \mathbf{J}_6],$$

where

$$\mathbf{J}_i = \begin{bmatrix} \mathbf{z}_{i-1} \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ \mathbf{z}_{i-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{i-1} \mathbf{z}_0 \times (\mathbf{p}_e - \mathbf{p}_{i-1}) \\ \mathbf{R}_{i-1} \mathbf{z}_0 \end{bmatrix}.$$

Bear in mind that the superscripts are omitted, and that in such cases the frame of reference is the base. The derived expressions for the transformations matrices between the different links, as well as the manipulators Jacobian is included in appendix C.

2.4 Calibration methods

A common problem setting up a robot system is relating the robot to other devices, such as tools and cameras. Several of these transformations can be unknown, but generally they constitute a circular relationship with static components. The problem frequently takes the form of $\mathbf{AX} = \mathbf{XB}$ or $\mathbf{AX} = \mathbf{YB}$, having one or two unknowns respectively. Consequently this has become a field of study in the robot community - the prior being well known as the ‘‘Hand-Eye calibration’’ [41]. The homogeneous matrices,

$$\mathbf{AX} = \mathbf{YB}, \quad (2.15)$$

can be rewritten as

$$\begin{bmatrix} \mathbf{R}_A & t_A \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_X & t_X \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_Y & t_Y \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_B & t_B \\ \mathbf{0}^T & 1 \end{bmatrix}.$$

Multiplying the matrices gives

$$\begin{bmatrix} \mathbf{R}_A \mathbf{R}_X & \mathbf{R}_A t_X + t_A \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_Y \mathbf{R}_B & \mathbf{R}_Y t_B + t_Y \\ \mathbf{0}^T & 1 \end{bmatrix},$$

which results in two equations, one purely orientational

$$\mathbf{R}_A \mathbf{R}_X = \mathbf{R}_Y \mathbf{R}_B, \quad (2.16)$$

and one containing the translational component

$$\mathbf{R}_A t_X + t_A = \mathbf{R}_Y t_B + t_Y. \quad (2.17)$$

So far, this is valid for both cases mentioned above. The further solution pattern can be divided into three categories, namely separable, simultaneous and iterative [42]. Separable solutions parts the task into solving the orientational from the positional component. Simultaneous solutions arise from solving the components simultaneously, while iterative solutions uses optimization techniques. Throughout the past decades, several methods for solving these problems are presented in literature. However, none are considered superior as there is always a reciprocity between accuracy and complexity which must be considered depending on the task at hand. The solution methods for both one and two unknowns have several similarities, the latter being of interest in this thesis.

To the best of the authors knowledge, Zhuang *et al.* presents the first closed-form separable solution of the $\mathbf{AX} = \mathbf{YB}$ problem utilized quaternions [43]. The proposed method is extended with increased accuracy in later work by Dornaika and Horaud [44]. Recently, Shah *et al.* proposed a separable method utilizing the Kronecker product and singular value decomposition with claims of being more

accurate and effective than the mentioned quaternion based methods [45]. A similar method, though simultaneous, has been presented earlier [46] by Li et al³. Both a separable and simultaneous solution utilizing the Kronecker product inspired by the mentioned work will be presented in the following.

2.4.1 Separable solution

Considering the orientation part first by rewriting equation (2.16) to

$$\mathbf{R}_A \mathbf{R}_X (\mathbf{R}_B)^T = \mathbf{R}_Y,$$

and vectorizing the matrix product by utilizing the property introduced in equation (2.7) we get

$$\begin{aligned} (\mathbf{R}_B \otimes \mathbf{R}_A) \text{vec}(\mathbf{R}_X) - \text{vec}(\mathbf{R}_Y) &= 0, \\ \begin{bmatrix} -\mathbf{I} & (\mathbf{R}_B \otimes \mathbf{R}_A) \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{R}_Y) \\ \text{vec}(\mathbf{R}_X) \end{bmatrix} &= 0. \end{aligned} \quad (2.18)$$

To obtain a unique solution for equation (2.18) it is necessary with at least $n > 3$ measurements of \mathbf{A} and \mathbf{B} . Assume n measurements are acquired, and that they are summarized. Multiplying the transpose of the resulting left sided column vector with itself gives

$$\begin{bmatrix} n\mathbf{I} & -\sum_{i=1}^n (\mathbf{R}_{B_i} \otimes \mathbf{R}_{A_i}) \\ -\sum_{i=1}^n (\mathbf{R}_{B_i}^T \otimes \mathbf{R}_{A_i}^T) & n\mathbf{I} \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{R}_Y) \\ \text{vec}(\mathbf{R}_X) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.19)$$

Setting $\mathbf{K} = \sum_{i=1}^n (\mathbf{R}_{B_i} \otimes \mathbf{R}_{A_i})$ and solving for both $\text{vec}(\mathbf{R}_X)$ and $\text{vec}(\mathbf{R}_Y)$ by separation results in two equations,

$$\begin{aligned} n^2 \text{vec}(\mathbf{R}_Y) &= \mathbf{K} \mathbf{K}^T \text{vec}(\mathbf{R}_Y), \\ n^2 \text{vec}(\mathbf{R}_X) &= \mathbf{K}^T \mathbf{K} \text{vec}(\mathbf{R}_X). \end{aligned} \quad (2.20)$$

Noticeably these expressions has the same form as the property introduced by equation (2.9) in the section about singular value decomposition. The solutions for $\text{vec}(\mathbf{R}_X)$ and $\text{vec}(\mathbf{R}_Y)$ is then proportional to the left-singular vector \mathbf{u} and right-singular vector \mathbf{v} of \mathbf{K} corresponding to the singular value n , which is also the number of measurements. This gives

$$\mathbf{R}_X = c_X \text{vec}^{-1}(v_i) \quad \text{and} \quad \mathbf{R}_Y = c_Y \text{vec}^{-1}(u_i). \quad (2.21)$$

³Beware of several algebraic mistakes in the method derivation in the article by Li *et al.* [46].

Since the rotation matrices is part of the special orthogonal group, the determinant is equal to 1. The proportionality constants can be determined accordingly,

$$\det(c_X \text{vec}^{-1}(v_i)) = c_X^3 \det(\text{vec}^{-1}(v_i)) = 1 \implies c_X = \frac{\text{sgn}(\text{vec}^{-1}(v_i))}{\det(\text{vec}^{-1}(v_i))^{1/3}},$$

with c_Y being solved similarly with u_i . Knowing the rotation matrix \mathbf{R}_Y we can solve for the translational part

$$\begin{bmatrix} \mathbf{R}_A & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{t}_X \\ \mathbf{t}_Y \end{bmatrix} = \mathbf{R}_Y \mathbf{t}_B - \mathbf{t}_A. \quad (2.22)$$

The expression takes the matrix form of a linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, and can thus be solved accordingly.

2.4.2 Simultaneous method

Performing vectorization on equation (2.16) directly and using the property of equation (2.7) gives

$$(\mathbf{I} \otimes \mathbf{R}_A) \text{vec}(\mathbf{R}_X) = (\mathbf{R}_B^T \otimes \mathbf{I}) \text{vec}(\mathbf{R}_Y),$$

which can be rewritten as

$$\begin{bmatrix} (\mathbf{I} \otimes \mathbf{R}_A) & -(\mathbf{R}_B^T \otimes \mathbf{I}) \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{R}_X) \\ \text{vec}(\mathbf{R}_Y) \end{bmatrix} = \mathbf{0}. \quad (2.23)$$

Vectorization of the translation part from equation (2.17), which already is an expression of vectors, yields

$$(\mathbf{t}_B^T \otimes \mathbf{I}) \text{vec}(\mathbf{R}_Y) + \mathbf{t}_Y + \mathbf{R}_A \mathbf{t}_X = \mathbf{t}_A.$$

Stacking the expression gives

$$\begin{bmatrix} (\mathbf{t}_B^T \otimes \mathbf{I}) & -\mathbf{R}_A & \mathbf{I} \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{R}_Y) \\ \mathbf{t}_X \\ \mathbf{t}_Y \end{bmatrix} = \mathbf{0}. \quad (2.24)$$

Consequently, a shared vector in equation (2.23) and equation (2.24) is identified making it possible to combine them and solve the problem simultaneously,

$$\begin{bmatrix} (\mathbf{I} \otimes \mathbf{R}_A) & -(\mathbf{R}_B^T \otimes \mathbf{I}) & 0 & 0 \\ 0 & (\mathbf{t}_B^T \otimes \mathbf{I}) & -\mathbf{R}_A & \mathbf{I} \end{bmatrix} \begin{bmatrix} \text{vec}(\mathbf{R}_X) \\ \text{vec}(\mathbf{R}_Y) \\ \mathbf{t}_X \\ \mathbf{t}_Y \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{t}_A \end{bmatrix}. \quad (2.25)$$

Again we end up with the form $\mathbf{A}\mathbf{x} = \mathbf{b}$ which can be solved using least-square methods. To describe the rotation as a matrix again, a devectorialization must be conducted.

2.4.3 Calibration error

Ideally inverting either side of equation (2.15) would yield identity. However, in a practical system where noise is present, inversion would give

$$(\mathbf{X}\mathbf{B})^{-1}\mathbf{A}\mathbf{X} = \boldsymbol{\epsilon} = \begin{bmatrix} \mathbf{R}_\epsilon & \mathbf{t}_\epsilon \\ \mathbf{0}^T & 1 \end{bmatrix},$$

where the matrix $\boldsymbol{\epsilon} \neq \mathbf{I}$ deviates from identity due to errors in the system. The translational error is then simply given by the length of the translation component in the error matrix,

$$\epsilon_{\text{trans}} = \|\mathbf{t}_\epsilon\|.$$

A common way to present the rotational error is utilizing the angle of rotation based on the axis-angle representation and Euler vector,

$$\epsilon_{\text{rot}} = \|\mathbf{v}\| = \|\mathbf{v}\mathbf{k}\|,$$

where \mathbf{v} is obtained from \mathbf{R}_ϵ using equation (2.3) and equation (2.4). When several measurements are obtained, an average errors is usually given.

In literature it is also common to avoid the Euler vector when determining the rotational error by simply expressing it as the sum of squares of the Frobenius norm[47], or alternatively the absolute error [44]. This is done summarizing the deviation in the equality using equation (2.16) for all the measurement,

$$E_{\text{rot}} = \sum \|\mathbf{R}_A \mathbf{R}_X - \mathbf{R}_Y \mathbf{R}_B\|_F^2.$$

Similarly, the translational error is given by the sum of squares of vector length in equation (2.17),

$$E_{\text{trans}} = \sum \|\mathbf{R}_A \mathbf{t}_X + \mathbf{t}_A - \mathbf{R}_Y \mathbf{t}_B - \mathbf{t}_Y\|^2$$

or eventually the relative error

$$E_{\text{trans}} = \left(\frac{\sum \|\mathbf{R}_A \mathbf{t}_X + \mathbf{t}_A - \mathbf{R}_Y \mathbf{t}_B - \mathbf{t}_Y\|^2}{\sum \|\mathbf{R}_A \mathbf{t}_X + \mathbf{t}_A\|^2} \right)^{1/2}$$

However, these methods of determining error is a selection of many, making it hard to compare experimental results with literature. In addition, many of the described methods is mainly investigated in a simulation setting with known noise, lacking ground in reality.

2.5 Medical imaging

Images are essential for medical interventions, and have a wide range of clinical applications. The techniques are mostly used to gain insight and information about

the interior of the human body through visualization of structures and physiology. Several modalities exist, each with their own strengths and weaknesses. Those encountered during the work of this thesis are ultrasound (US), magnetic resonance imaging (MRI) and computed tomography (CT). A simplified description is therefore presented briefly below.

Ultrasound

Ultrasound is defined as sound waves with frequencies above the audible range [48], and for medical diagnostic ultrasound it is usually operated in a megahertz interval. The basic principle is transmission of waves into the body using a transducer, followed by a reception of echoes returning from the internal structures. Echoes arise from the transmitted waves reflected on boundaries with different acoustic impedance. The signal are then processed and displayed. The most generic and used display-mode is arguably 2D B-mode, which is based on transmitting short pulses of ultrasound into the body, and utilizing the fact that they are reflected at different depths.

Magnetic resonance imaging

Magnetic resonance images are generated using strong magnetic fields, radio frequency (RF) pulses and smaller magnetic field gradients [49]. Roughly, the principle is based on utilizing that certain atoms have nuclear spin, and that humans consist of approximately 70 percent water (H_2O) which means an immense number of hydrogen atoms. As the patient enters the magnetic field, the hydrogen protons in which normally have randomly oriented magnetic moments will align with the field. A slightly larger proportion will align along the field causing a net magnetic moment. The protons precess around the axis of the magnetic field with a rate called the Larmor frequency. An alternating magnetic field causes a disturbance in the proton alignment, changing the net magnetisation. RF coils are then used to receive signals as protons tend back to their normal state in a process known as relaxation. The signal can then be used to produce images. It is also worth mentioning that gradient coils allow spatial encoding for MRI, and can be used for slice selection, allowing axial, coronal or sagittal images.

Computed tomography

Two dimensional X-ray images from a patient is obtained illuminating X-rays through the body and using a device to detect the transversed rays. Computed Tomography (CT) combines several X-ray projections taken from a range of angles and produces cross sectional images based on the acquisitions [50]. The technique enables high resolution three-dimensional images of internal structures of interest.

X-ray radiation attenuates differently with varying kind of matter, such as tissue types and bones, giving contrast in the obtained image data.

2.6 Image processing and analysis

The first step in a medical image processing and analysis system, at least in this context, is acquiring the digital images from the modality of investigation. After obtaining the digital image, the following, namely digital image processing, involves performing mathematical operations on the image aiming to enhance, filter, restore or extract useful information from it [51], the latter being called image segmentation.

Image segmentation is a process used to analyze images by partitioning it into individual segments based on common characteristics. In a medical context the property is usually segments belonging to biological structures, such as specific tissue types or organs. This can be useful, both for visualizing the object of interest, but also for registration and measurements in physical space. Multiple segmentation approaches exist, but a proper categorization is difficult since methods tend to be both very different and at the same time share multiple properties. Common approaches involve thresholding, region growing, morphology, watershed, active contours, level sets, atlas/registration-based, statistical shape models, centerline extraction and dynamic image segmentation (tracking) [52].

In this thesis several filtering and detection methods are combined in the image processing pipeline, including thresholding, Gaussian smoothing, edge detection, Hough transformation and Kalman filtering. The following will provide a brief introduction of the mentioned applied on two-dimensional images, but could just as well be used in a three-dimensional case. Real examples of application will be presented in later chapters.

2.6.1 Thresholding

Image thresholding is arguably the simplest segmentation method in its basic form. The method is based on distinguishing image objects based on intensity by creating a binary image [53]. Assigning the pixels with intensity over a given threshold τ as object points, and the remaining to background points, groups the image into distinct modes. The segmented image S can thus be given as

$$S_{\tau}(u, v) = \begin{cases} 1 & \text{if } I(u, v) > \tau \\ 0 & \text{otherwise,} \end{cases}$$

where I is the input image. This is called global thresholding, but intuitively one can assign different values of T to different regions of the image. Multiplying the original image $I(u, v)$ with the binary segmented image $S(u, v)$ would retain contrast

in the object part of the output image. We define the thresholded image I_τ as

$$I_\tau(u, v) = I(u, v) \times S_\tau(u, v) \quad (2.26)$$

Extending development of the basic concept has resulted in an immense amount of thresholding techniques and algorithms utilizing different information [54]. Some of the methods assign the threshold value τ automatically, making it suitable for segmentation without user interaction.

2.6.2 Gaussian smoothing

Noise is definitely prevalent in medical images and is generally caused by electronics involved in the acquisition. In mathematics, the Gaussian function has several and important applications, and can also be applied to image processing [55]. Gaussian smoothing, or blurring, is typically used to reduce unwanted noise and detail in an image. In two dimensions, the Gaussian function with standard deviation σ can be defined as

$$G(u, v | \sigma) = \alpha \exp\left(-\frac{u^2 + v^2}{2\sigma^2}\right),$$

where α is the weighting scale factor. Here, u and v denotes the image coordinates. The smoothing is done convolving the input image with the Gaussian function,

$$I(u, v) * G(u, v | \sigma) = (I * G_\sigma)(u, v). \quad (2.27)$$

Convolving the image with a Gaussian function effectually works as a low-pass filter as it attenuates the image high frequency components. The standard deviation σ acts as the smoothing factor, and increasing it blurs the image accordingly.

Edge detection are sensitive to noise, and Gaussian smoothing is commonly used with such detection algorithms. Assigning a significant smoothing factor is necessary to remove noise, but higher values result in a loss of image information which at worst could make it impossible to distinguishes object edges.

2.6.3 Edge detection

Edges are boundaries between different regions of the given image. Identifying these in a digital image is equivalent to finding the areas where image intensity/brightness changes rapidly. Several methods exist, and most utilizes first or second order derivatives of image intensity. From mathematics we know that the directional derivation, or gradient, gives us the rate and direction of most rapid increase, i.e

$$\nabla I(u, v) = \left[\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v} \right], \quad (2.28)$$

where $I(u, v)$ is the image intensity function and ∇ is the gradient operator. The gradient magnitude $\|\nabla I(u, v)\|$ is the edge strength with direction $\theta = \arctan(\frac{\partial I}{\partial u} / \frac{\partial I}{\partial v})$.

Recalling that convolution is both commutative and associative an useful implication is that

$$\nabla(I(u, v) * G(u, v | \sigma)) = I(u, v) * \nabla G(u, v | \sigma),$$

which saves one operation as the gradient of the Gaussian is known.

2.6.4 Circle Hough transform

A technique often used in image segmentation is the the edge-based Hough transform [56]. It can be used to detect lines, circles and other curves representable by an parametric equation. In a practical sense parametric equations are not well suited, but structures such as imperfect circles can be found using these techniques with the right parameters. A circle can be described by

$$(u - u_0)^2 + (v - v_0)^2 = r^2,$$

where r is the circle radius, and (u_0, v_0) is the center offset from the origin. In practice, thresholding, Gaussian filtering and edge detection is usually a preprocessing part of the Hough transform, followed by the actual circle search. At each identified edge pixel a circle with center at that point with radius r can be drawn, as seen by figure 2.5. This can be used to create a accumulator image $I_{A|r}$, often named voting space, containing the information about circle intersections. The intersection point of most drawn circles would correspond to the center of the real circle.

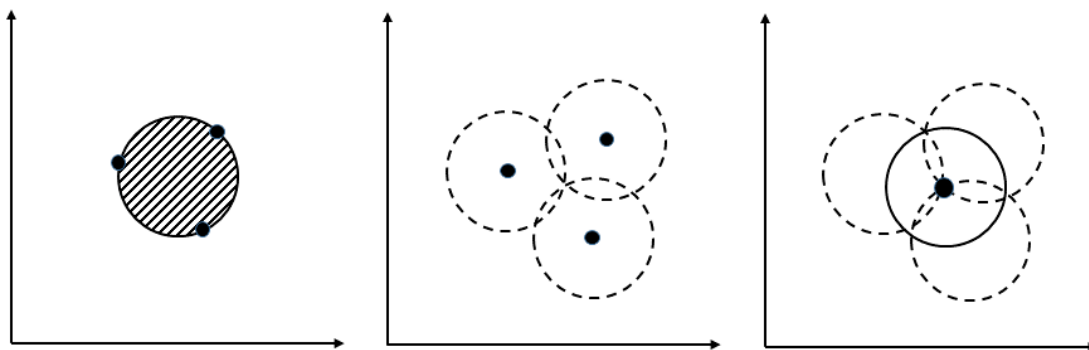


Figure 2.5: Example of the standard Hough transform searching for a circle with a given radius. The left image shows three samples from the edge of an actual circle with known radius. At every sampled point a voting circle with the same radius is drawn (middle). The point of most intersection is the proposed center of the actual circle as shown in the rightmost image.

2.6.5 Kalman filter

The Kalman filter method is an approach for recursive estimation of a process state from sequences of discrete and noisy measurements in the time domain [57]. According to the model, the state \mathbf{X}_k of a system at time k emerging from a previous state \mathbf{X}_{k-1} is given by

$$\mathbf{X}_k = \mathbf{A}_k \mathbf{X}_{k-1} + \mathbf{B}_k \mathbf{u}_k + \mathbf{v}_k,$$

where \mathbf{A}_k is the state transition matrix. The control input term is included if known and consists of the input matrix \mathbf{B}_k and vector \mathbf{u}_k . The process noise, here noted as \mathbf{v}_k , is assumed to have a normal probability distribution with zero mean and covariance \mathbf{Q}_k , i.e. $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{Q}_k)$. Measurements \mathbf{Y}_k at a given time is given by

$$\mathbf{Y}_k = \mathbf{C}_k \mathbf{X}_k + \mathbf{w}_k, \quad (2.29)$$

where the measurement matrix \mathbf{C}_k maps the state into measurement space. The noise from measurements, similarly but independent of the process noise, is given as $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{R}_k)$, where \mathbf{R}_k is the covariance. For algorithm conceptualization the Kalman filter is conventionally decomposed into two phases, prediction and update.

The prediction phase is preparatory, creating an estimation of the next state $\hat{\mathbf{X}}_{k|k-1}$ given the previous, expressly

$$\hat{\mathbf{X}}_{k|k-1} = \mathbf{A}_k \hat{\mathbf{X}}_{k-1|k-1} + \mathbf{B}_k \mathbf{u}_k. \quad (2.30)$$

Similarly, the estimation of the covariance \mathbf{P} is given as

$$\mathbf{P}_{k|k-1} = \mathbf{A}_k \mathbf{P}_{k-1|k-1} \mathbf{A}_k^T + \mathbf{Q}_k. \quad (2.31)$$

Upon a new acquisition the update phase incorporates the measurement correcting the estimation from the prediction phase. The first step, also known as the innovation step, calculates the the residuals of measurement $\tilde{\mathbf{Y}}$ and covariance \mathbf{S} . The prior is given by

$$\tilde{\mathbf{Y}}_k = \mathbf{Y}_k - \mathbf{C}_k \hat{\mathbf{X}}_{k|k-1}, \quad (2.32)$$

where \mathbf{Y} is the measurement, while the residual covariance is calculated from

$$\mathbf{S}_k = \mathbf{C}_k \mathbf{P}_{k|k-1} \mathbf{C}_k^T + \mathbf{R}_k.$$

An essential part of the algorithm is the Kalman filter gain \mathbf{K} which describes the relative importance of the measurement residual with respect to the previous state estimate. In other words, the optimal Kalman gain is given by

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{C}_k^T \mathbf{S}_k^{-1}. \quad (2.33)$$

The corrected and updated estimates for state and covariance are then

$$\hat{\mathbf{X}}_{k|k} = \hat{\mathbf{X}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{Y}}_k, \quad (2.34)$$

and

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_{k|k-1}, \quad (2.35)$$

respectively. The correction is only valid for the optimal Kalman gain.

2.7 Software

Several components constitutes the development environment in this project. The programming is mainly done using CustusX as a toolkit, which is written in C++ using CMake, Qt, CTK, Boost, VTK, ITK, Eigen and other libraries [58].

2.7.1 CustusX

CustusX is an open source research platform for image-guided therapy (IGT), which primary focuses on intraoperative navigation and ultrasound (US) imaging [58]. It was created and is maintained by SINTEF Medical Technology⁴ in cooperation with the Norwegian National Advisory Unit for Ultrasound and Image guided Therapy⁵. CustusX has been in continuous development for more than 15 years, with close collaboration between software developers, technical and medical researchers. Eventually it has become a mature research platform for use in the IGT research community, and has been used in several scientific studies with a wide range of clinical applications.

In terms of image modalities CustusX supports Ultrasound, but also pre-operative modalities such as computed tomography (CT) images, magnetic resonance (MR) images, positron emission tomography (PET) images and more. During interventional procedures instrument poses can be tracked and visualised in real-time within the software navigation window. According to the developers, validation experiments has shown that the platform has a navigation system accuracy (NSA) of < 1.1 mm, a frame rate of 20 fps and latency of 280 ms for a typical setup.

As of now CustusX is intended for research, and is not approved for use clinically. For such, the end user needs necessary approvals according to safety regulations and more. An extended version is used daily in the research environment at St. Olavs Hospital in Trondheim, Norway.

⁴<http://www.sintef.no/sintef-teknologi-og-samfunn/medisinsk-teknologi/>

⁵<http://www.usigt.org/>

Plugin extensions

CustusX is an integrating platform with a layered architecture. It can be used as a customizable navigation system, or as a toolbox for developing new applications. Internal resources and external libraries constitute the building blocks of the system, and combined these provide a lot of the systems functionality. The discrete plugins enables several of its core features, and serve as an extensibility point for further development.

One of the main goals from a developer point of view, was making the CustusX platform easy to extend while keeping the core stable. In practice this means that unstable extensions should not disrupt the rest of the system. As of 2014, CustusX uses the CTK Plugin Framework [59] as a dynamic component, based on the OSGi specifications [60]. This framework provides a simple and standardized architecture, making it easy for developers to extend with their own features.

The CTK Plugin framework introduces the concepts of services and plugins. The former are implementations of abstract interfaces available from the plugin repository, while the latter are physical components that can be loaded/unloaded at runtime. A plugin has one or more service implementations.

2.7.2 External libraries

CustusX, as well as the robot framework presented below, uses several external libraries throughout the application. CustusX is the integrating platform, with the libraries contributing with much of the functionality. The following will present some of the most central components. It is also worth mentioning that CustusX has cross-platform compatibility, meaning it can operate on several operating systems including MS Windows and Mac OS X, as well as Unix based systems such as Ubuntu. Being cross-platform and open-source makes the system more accessible for developers.

Qt

Qt is a cross-platform application development framework extending the standard C++ language with simplified graphical user interface development possibilities [61], as well as back end tools including the signals and slots mechanism. The signals and slots mechanism provides easy communication between the objects in the system, which has proven very beneficial for this sort of framework development.

Eigen

Eigen is a fast and elegant open source C++ template library with cross-platform support for linear algebra [62]. It supports matrix and vector operations, transformation methods, as well as having a large set of algorithms for numerical operations.

Several renowned software projects related to robotics, engineering and computer graphics uses Eigen including Google (Street view) and Willow Garage (Robotics operating system).

The Visualization Toolkit

The Visualization Toolkit (VTK) is an open-source software system and toolkit for computer graphics, image processing and visualization [63]. It is cross-platform compatible and implemented as a C++ class library. VTK has a large user group being one of the oldest and still actively developed open source toolkits. It integrates well with GUI toolkits such as Qt, and also data manipulation toolkits such as ITK.

The Insight Toolkit

The National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) is an open-source cross-platform library consisting of software components to perform data filtering, segmentation, and registration of data in two or more dimensions [64]. It is implemented in C++ designed as a complementary library, using templated code. It contains no visualization functionality, but integrates well with VTK. The toolkit contains several image processing algorithms renowned in the medical imaging community, being its focus area.

2.7.3 Robot Framework

The implemented robot framework is a result of the authors specialization project [6]. In short, it enables basic robotic control in CustusX without any extensions related to medical applications. However, the aim is to integrate the relevant methods of the two platforms and enable low threshold development possibilities emphasising medical applications. Some of the core functionality is summarized below.

Communication

A two-way communication between CustusX and the physical robot is established. While connected the received data packets are analyzed at a rate up to 125Hz, providing updated and relevant information about the robots current state. This limit is on the manufacturer side of the system, and practice it can be lower depending on workstation and general load. The packages sent to the robot contain information about desired pose, as well as other motion settings, such as velocity and acceleration. This enables real-time control of the manipulator directly from CustusX.

Kinematics and motion generation

The robot kinematics is used to calculate the transformation between the robot base and end-effector \mathbf{T}_e^b , as well as the Jacobian \mathbf{J} , given a joint configuration \mathbf{q} . The inverse kinematic solution, i.e calculating the joint configuration from the transformation matrix in operational space, is also implemented utilizing the Jacobian. The expressions used are derived symbolically in Matlab for the specific manipulator and the algebra operations are done using Eigen in the framework.

A sequence of operations, such as moving along a given path, is implemented as a queue system with timing laws. Two types of motion are possible, point to point or velocity profile motion. The prior is very precise, but is not well suited for scans when jerking is not desired. The latter, is less precise but enables smooth motion along complex curves.

Interactive graphical user interface and visualization

The framework also includes a graphical user interface from which users can operate the robot manipulator. The current pose of the robots end-effector, as well as the current joint configurations are updated in real time. Several applications exist, including manual and planned movement, as well as simulation methods for mimicking respiratory effects.

The end-effector can also be visualized in the navigation view area of CustusX as seen in figure 2.6. From a clinical perspective this is believed to have benefits in the following development since interaction with objects, and hopefully patients, through visualization would improve the users cognition.

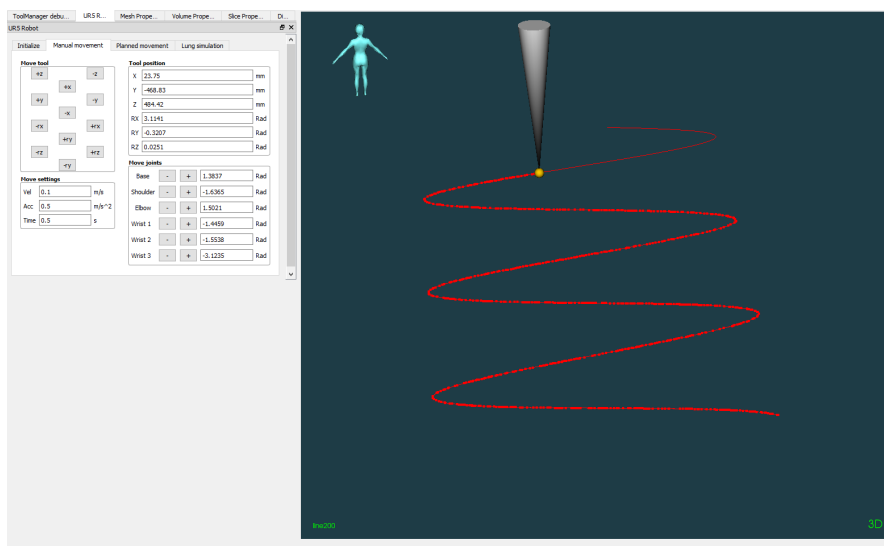


Figure 2.6: Visualization of the robot end-effector in the view area of CustusX. The end-effector pose is illustrated by the tip of the grey cone. The thick red dots is a tracked path in the navigation window, while the red thin line is the predefined motion path. A user interaction tab is also visible on the left hand side.

Chapter 3

Methodology

The implemented extensions with concept development proposing solutions for the project goals is presented in this chapter, followed by verification experiment methods and protocols. As explained, CustusX is programmed in the language C++ based on plugin extension possibilities using the Qt application framework, as well as several other external libraries. The robot framework is, and will continue to be based on the same dependencies.

The chapter will not serve as a stepwise guide to reimplementing the concepts and framework as a whole, but rather an abstraction with methods used for enabling the core functionality and essentials. Presenting the pseudo code is believed to increase the readability and readers understanding of the involved theory. This will also generalize the code, making re-implementations using other programming languages and libraries easier. The interested reader is of course welcome to view the raw code presented following links in Appendix A.

Additional expandability was remarked in the specialization project [6], and some are therefore implemented despite being a necessity achieving the submitted project goals. This includes better visualization in the user interface, which will be shown later. The following sections will describe the concept development and experiments used in the verification process.

3.1 Overview of the concept development

To fulfill the projects main goals, several distinct tasks are identified that could be solved and verified individually. First, the robot arm needs to relate spatially to other objects of interest, such as medical tools, pointing instruments and medical images. In practice this requires a flexible and fast calibration as the operational scene is an environment with varying display. By accomplishing this, spatial relationships between the systems relevant components is mapped. Controlling the robot appropriately, and eventually autonomous, requires additional feedback information and analysis. As mentioned in the background, visual servoing, or

vision-based robot control, is a technique used to manipulate the robot in response to data analysis from vision based sensors [65]. The proposed system uses two different vision based sensors. For external registration the optical tracker Polaris Spectra from NDI Medical¹ (Waterloo, ON, Canada) is utilized, while for inner body registration, real-time image acquisition using ultrasound is used. The tracker can register passive markers in its volumetric view, which can be distinguished by creating physical unique frames consisting of multiple markers. These frames are often mounted or integrated into instruments making them trackable in space. Knowing where the ultrasound probe is located in space as a result of the calibration process, enables location of the ultrasound image with respect to the robot. By autonomously identifying clinical relevant information in a medical image using processing routines, the robot arm can potentially manipulate a medical instrument with respect to the acquired information. Of course, the information needs to be interpreted such that the robot arm moves in an expedient way.

Summarized, the steps described above can be divided into

- Relating objects of interest
- Calibration
- Image processing and analysis
- Robot manipulation

These four pillars are presented below, followed by a proposed set of experiments aiming to validate the implementations. Despite being solved separately, the synergy effect of combining them is the higher purpose.

3.2 Relating objects of interest

Relevant coordinate systems for the robot framework are visualized in figure 3.1, with homogeneous transformation matrices relating them in space. The transformation from the robot base to the end-effector \mathbf{T}_{ee}^b is derived using the Denavit-Hartenberg convention, as explained in section 2.3.3. This functionality is implemented in previous work and can be obtained in real-time using the robot framework.

As mentioned, an optical tracker is used to register the patient and tool in space, the transformations being \mathbf{T}_{pr}^c and \mathbf{T}_t^c respectively. If both the patient reference frame and tool frame is visible in the trackers volumetric view, the transformation between them, namely \mathbf{T}_t^{pr} , is possible to calculate. \mathbf{T}_{ee}^b and \mathbf{T}_t^{pr} defines two dynamic transformations obtained using different sensors, and thus being independent

¹<http://www.ndigital.com/medical/>

in terms of accuracy. Mounting a tool registered by the tracker on the robots end-effector will relate the systems and cause mutual movement. Moreover, two spatial relationships from the robot arm to the rest of the core system is developed. The robot end-effector to tool transformation T_t^{ee} and robot base to patient reference T_{pr}^b . Initially these transforms are unknown, but using calibration methods as described in the next section can potentially give good estimations for the calibration matrices.

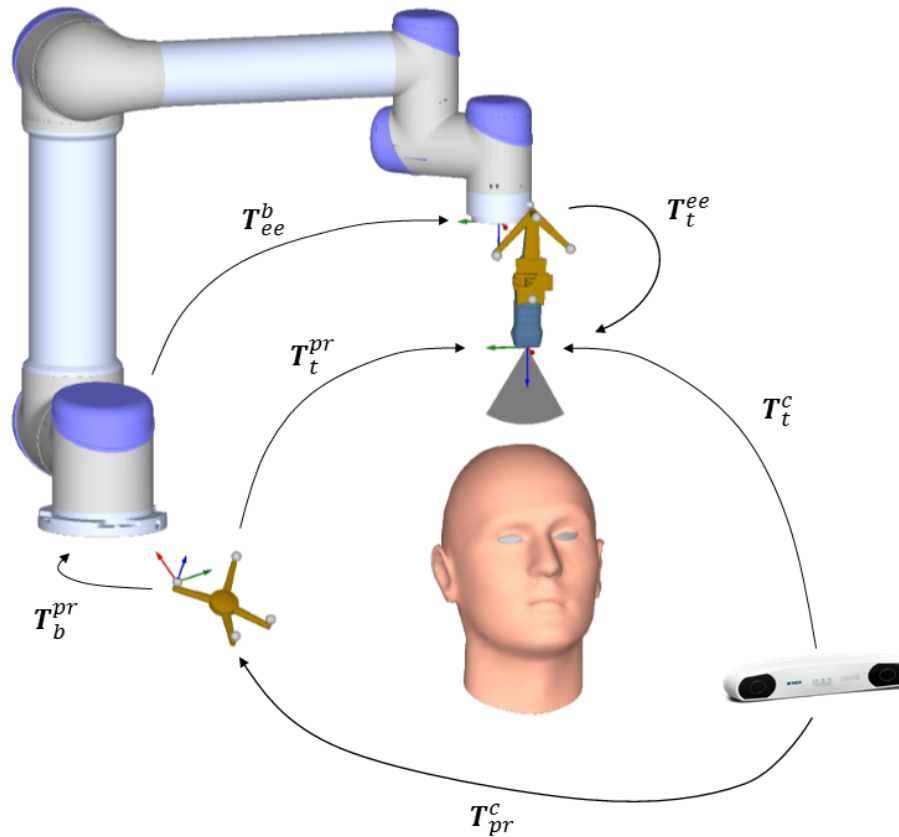


Figure 3.1: A simplified map of the spatial relationship between robot arm and relevant coordinate systems in the framework. A camera is registering both the patient reference frame and the tool frame mounted on the medical instrument. The robot base to end-effector transform T_{ee}^b is obtained utilizing information from the robots sensor system. The robot is connected spatially to the rest of the system from both the base and end-effector.

The patient reference is used as the frame of junction, or world frame, as can be seen from figure 3.2. It is used to relate all system relevant features spatially. This includes both real-time image data obtained from an ultrasound probe, the probes location in space, real-time pose data from a wireless pointer, as well as other data of interest. The data referred to in the same figure represents preoperative patient data, such as MRI and CT. Correlating the transformation from previously acquired patient data to the present patient frame requires a patient registration procedure.

Research on the topic of patient registration has resulted in several approaches for solving the issue [66], and the reader is referred to literature for a proper review. Here the concept of reference markers, or passive markers, are used.

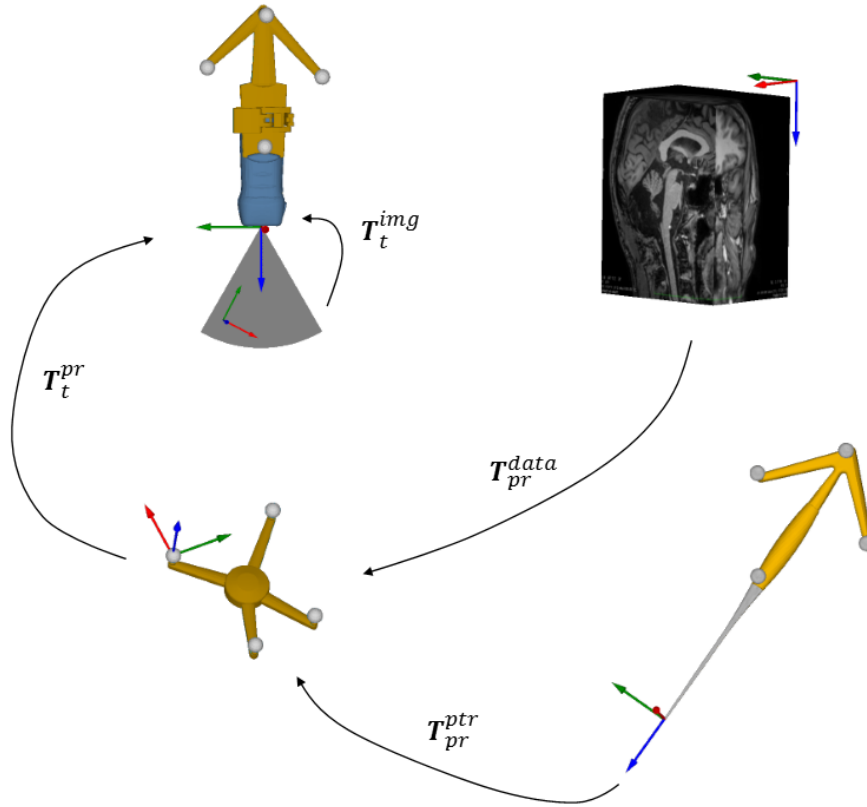


Figure 3.2: A simplified map of the spatial relationship between patient reference and different tools, image data, pointers and preoperative data. The patient reference frame, and possibly tool frame, is the connection point for the robot arm. A wireless pointing instrument, as well as the tool is tracked in real-time using an optical camera. Here the tool is represented by an ultrasound probe with image data. Additionally, preoperative data can be registered to the scene, but this requires synchronisation with real-time data.

Introducing an ultrasound imaging system as a vision based sensor for manipulating the robot arm requires a spatial connection between the robot task space and the ultrasound image space, as illustrated in figure 3.3. The ultrasound are transmitted and received at the probe head which is tracked in space by the optical camera. Signal acquisition and processing results in a digital image contain intensity data mapped in pixel space. Here, the image is defined with the origin in the top left corner with pixel coordinates u and v . There is a linear relationship between the Cartesian and pixel image space given by $x \cdot x_{max}^{-1} = u \cdot u_{max}^{-1}$, and knowing the image size in both systems results in an unambiguous solution. The relationship can be applied to other image dimensions as well. Image processing happens in the pixel domain, and relevant analysis leading to quantitative information such as

structure location and size can then be transformed to Cartesian coordinates and then be related to probe center. The full relationship can be expressed as the image to tool transformation \mathbf{T}_t^{img} .

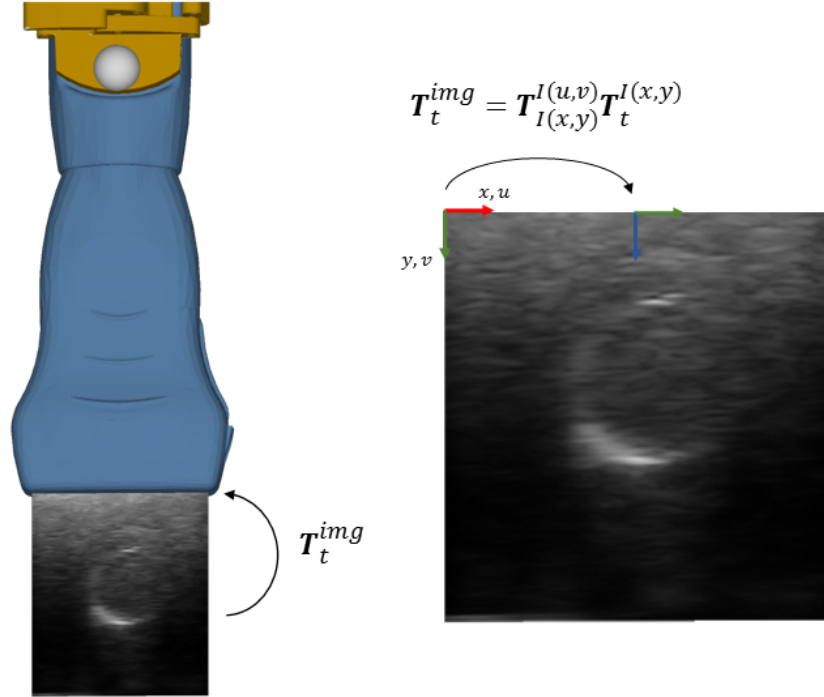


Figure 3.3: Visualization of an imaging ultrasound probe. Image analysis is performed in the pixel intensity domain of the ultrasound image, where u and v is the horizontal and vertical pixel coordinates respectively. The pixel coordinates shares the origin with the physical image coordinates x and y , and are proportionally related by $T_{I(x,y)}^{I(u,v)}$. The transformation $T_t^{I(x,y)}$ from the physical image space to the tool/probe center connects the image pixel domain to the robot framework.

3.3 Calibration procedure

Interaction between the robot arm and other objects of interest, such as medical tools and patients, requires a flexible and precise calibration method. From figure 3.1 two inner circular relationship are identified, with the patient reference to tool transformation \mathbf{T}_t^{pr} being a common junction. In other words,

$$\mathbf{T}_t^{pr} = \mathbf{T}_b^{pr} \mathbf{T}_{ee}^b \mathbf{T}_t^{ee} = \mathbf{T}_c^{pr} \mathbf{T}_t^c. \quad (3.1)$$

Initially, the robot arm has no information about the patient reference nor the tool location, which is crucial enabling interaction. The transformation between these coordinate frames, namely \mathbf{T}_t^{pr} , is known combining the measurements from the optical camera. Omitting the camera reference, the left part of equation (3.1) can

be rewritten as

$$\mathbf{T}_{pr}^t \mathbf{T}_b^{pr} = \mathbf{T}_{ee}^t \mathbf{T}_b^{ee}.$$

This circular relationship consist of two unknowns, i.e \mathbf{T}_b^{pr} and \mathbf{T}_{ee}^t , but keeping them constant during calibration makes it possible to solve for either of them. An unique solution can be obtained conducting several simultaneous measurements

$$(\mathbf{T}_{pr}^t)_{(i)} \mathbf{T}_b^{pr} = \mathbf{T}_{ee}^t (\mathbf{T}_b^{ee})_{(i)}, \quad \text{for } i \in [1, \dots, n]$$

where $n > 1$ is the number of measurements. Noticeably this takes the form $\mathbf{A}_i \mathbf{X} = \mathbf{Y} \mathbf{B}_i$, where $\mathbf{X} = \mathbf{T}_b^{pr}$ and $\mathbf{Y} = \mathbf{T}_{ee}^t$, and we can utilize the methods described in section 2.4. The following pseudo code is the basis of generating the calibration matrices between the patient reference and the robot base \mathbf{T}_b^{pr} and between end-effector and tool \mathbf{T}_t^{ee} . Algorithm 1 is based on the seperable method presented in section 2.4.1.

Algorithm 1 Separable calibration method

Require: A set of $n > 3$ corresponding transformations \mathbf{T}_t^{pr} and \mathbf{T}_{ee}^b

```

procedure CALIBRATESEPARABLE(  $\mathbf{T}_t^{pr}, \mathbf{T}_{ee}^b$  )
  for every pair of transforms  $\mathbf{T}_t^{pr}$  and  $\mathbf{T}_{ee}^b$  do
    Append  $\mathbf{K} \leftarrow \mathbf{K} + \mathbf{R}_{pr}^t \otimes \mathbf{R}_b^{ee}$  ▷ Equation (2.19)
  end for

  Calculate  $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V} \leftarrow$  SVD of  $\mathbf{K}$  ▷ Equation (2.8)
  Find  $\sigma_i^2$  corresponding to  $n^2$  ▷ Equation (2.20)
   $\mathbf{R}_b^{pr}, \mathbf{R}_t^{ee} \leftarrow$  Devectorize  $u_i$  and  $v_i$  ▷ Equation (2.21)

  for every pair of transforms  $\mathbf{T}_t^{pr}$  and  $\mathbf{T}_{ee}^b$  do
    Append  $\mathbf{A} \leftarrow [\mathbf{R}_b^{ee} \quad -\mathbf{I}]$  ▷ Equation (2.22)
    Append  $\mathbf{b} \leftarrow \mathbf{R}_t^{ee} \mathbf{t}_{pr}^t - \mathbf{t}_{ee}^b$ 
  end for

  Solve  $\mathbf{A} [\mathbf{t}_{pr}^b \quad \mathbf{t}_t^e]^T = \mathbf{b}$  ▷ Least squares methods

   $\mathbf{T}_b^{pr}, \mathbf{T}_t^{ee} \leftarrow \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$  ▷ Equation (2.5)

return  $\mathbf{T}_b^{pr}$  and  $\mathbf{T}_t^{ee}$ 
end procedure

```

As mentioned before, the separable methods have an error propagation of the orientation component into the translation. To investigate the effect of this, algorithm 2

is based on the simultaneous method presented in section 2.4.2.

Algorithm 2 Simultaneous calibration method

Require: A set of $n > 3$ corresponding transformations \mathbf{T}_t^{pr} and \mathbf{T}_{ee}^b

```

procedure CALIBRATESIMULTANEOUS(  $\mathbf{T}_t^{pr}$ ,  $\mathbf{T}_{ee}^b$  )
  for every pair of transforms  $\mathbf{T}_t^{pr}$  and  $\mathbf{T}_{ee}^b$  do
    Append  $\mathbf{A} \leftarrow \begin{bmatrix} \mathbf{I} \otimes \mathbf{R}_b^{ee} & -(\mathbf{R}_{pr}^t)^T \otimes \mathbf{I} & 0 & 0 \\ 0 & (\mathbf{t}_{pr}^t)^T \otimes \mathbf{I} & -\mathbf{R}_{pr}^t & \mathbf{I} \end{bmatrix}$   $\triangleright$  Equation (2.25)
    Append  $\mathbf{b} \leftarrow \begin{bmatrix} 0 \\ \mathbf{t}_{pr}^t \end{bmatrix}$ 
  end for

  Solve  $\mathbf{A} \begin{bmatrix} \text{vec}(\mathbf{R}_{pr}^b) \\ \text{vec}(\mathbf{R}_t^e) \\ \mathbf{t}_{pr}^b \\ \mathbf{t}_t^e \end{bmatrix} = \mathbf{b}$   $\triangleright$  Least squares methods

   $\mathbf{T}_b^{pr}, \mathbf{T}_t^{ee} \leftarrow \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}$   $\triangleright$  Equation (2.5)

  return  $\mathbf{T}_b^{pr}$  and  $\mathbf{T}_t^{ee}$ 
end procedure
    
```

Practically, it would be most appropriate keeping the patient reference frame at a static pose after the calibration. A possible scenario is the desire to change instruments, and keeping the patient reference constant with respect to the robot base enables automatic initialization of new instruments as $\mathbf{T}_t^{ee} = \mathbf{T}_b^{ee} \mathbf{T}_{pr}^b \mathbf{T}_t^{pr}$, with the right hand side being known after the calibration routine. On the downside, determining the transformation matrix without a proper calibration can induce additional errors as will be discussed later.

All the matrix calculations, including vectorization and Kronecker multiplication is performed using the external library Eigen. A small introduction to the library was given in section 2.7.2. In addition, Eigen has functionality for singular value decomposition and least squares methods which is utilized.

3.4 Semi-automatic circle detection in US images

Identifying structures of interest autonomously in an ultrasound image requires an image processing pipeline ending in a segmentation, or at least an interpretation of the structures location in physical space. In this project this structure is chosen to be

a sphere, but by substituting the detection part, other arbitrary structures could of course be used. Clinically, the object of interest would be anatomic structures such as specific organs, tissue or vessels, but could also be abnormalities like tumors, cysts and calcification. It is well known that the shapes of such structures vary, which makes it hard, or even impossible to generalize this step. Therefore it must be implemented based on the application at hand. Though perfect spheres and circles generally not exist in the human body, several structures tend to have a circular form. They are also commonly used in phantoms and laboratory calibration equipment as will be shown later on.

Depending on the application the first step of the image processing is optional cropping. If an ultrasound scan is performed over a defined volume in search for abnormal structures, the algorithm should search the whole image for abnormalities given that previously acquired information do not say otherwise. On the other hand, if a structure is detected, and the task is to track and keep it in the image plane, one would use cropping methods to decrease the size of the analysed image and thus increase the speed of processing. Decreasing the size of the analysed image can also decrease the chance of false positive identifications.

As described in section 2.6.4 about the Hough transform, searching for a circles in an image requires some prior knowledge about its radius r . Optionally this can be an interval, as it would be in a practical scenario. The cropping width is then assigned as $\Delta c = 2(\Delta d + r)$, where Δd is the motion limitation range. Figure 3.4 gives an illustration of the prior. This can potentially decrease the size of the analysed region by not considering irrelevant image areas, thus saving a significant amount of processing time.

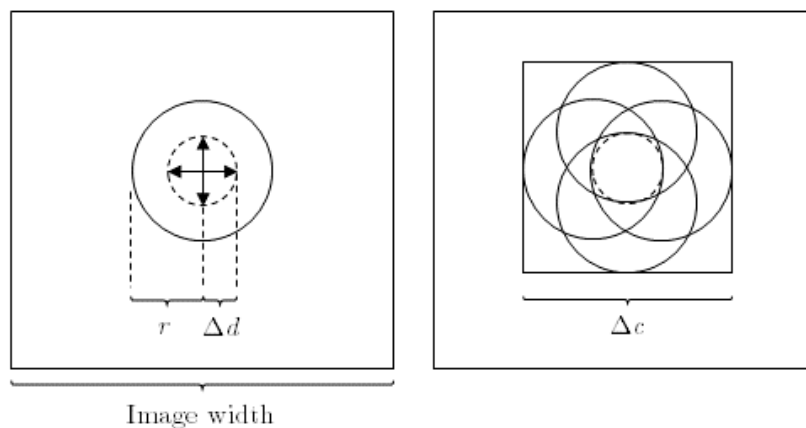


Figure 3.4: Illustration of the implemented cropping method. The left image shows a sampled circle with radius r . The maximum offset of the circle between samples is assumed to be limited by a value Δd set by the user. The cropping window Δc is calculated according to these parameters.

Motion of anatomic structures is mostly periodic and caused by respiratory effects, but quantification is difficult without any prior knowledge. This has also been an

area of extensive research as it constitute a major issue in several medical procedures, especially radiation based therapy and therapeutic ultrasound. Here, an assumption is that the total displacement of the structure is lower than the maximum displaced point on the thoracic diaphragm during deep breath. A literature summary of organ motion in the thoracic and abdominal regions show that the displacement generally is lower than 100 mm [67], being the organ with highest displacement caused by respiration. To translate this to a value of limited motion per frame, we also need to make assumptions on frame rate and displacement time, i.e inhalation/exhalation time. For this setup, the frame rate is assumed to always be better than 10 fps while the minimum displacement time is set to 1 s. This gives a motion limitation of $\Delta d = 10\text{mm}$ per frame.

After the cropping is finalized, the actual image processing pipeline is initiated. Several of the algorithms described earlier and used in the following are already implemented into the external library ITK. ITK is briefly described in section 2.7.2. The extensive set of algorithms includes several methods for thresholding, smoothing, image differentiation and even Hough transform. Consequently they were used in varying extent.

In figure 3.5 we see parts of an imaged liver mimicing model with a tumor implant. The left image is the original cropped image, where parts of the tumor boundary is clearly visible. To reduce the signal, thresholding is performed with a threshold value T equal to the non zero intensity average of the original image. The effect of increased threshold value is seen from left to right in the image sequence, which also proves that setting too high threshold values will remove the object entirely making it impossible to identify. Using thresholding is optional, but generally it is used here to decrease signal by removing some low intensity pixels, and consequently decrease the processing time.

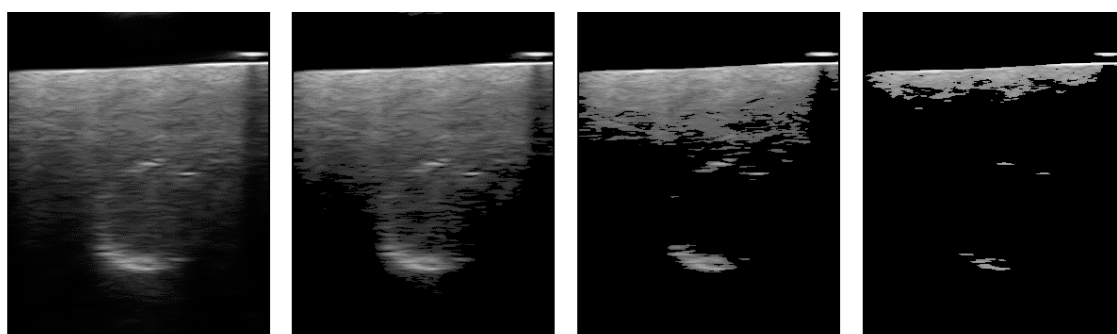


Figure 3.5: Effect of thresholding. The leftmost image is the original digital 8 bit grayscale image obtained from the ultrasound scanner. The second image from the left is thresholded with a value equal to the non zero intensity average of the original image keeping pixels with threshold value $T > 44$ where $T \in [0, 255]$. Similarly, the adjacent images are thresholded with values $T > 88$ and $T > 132$ respectively.

The following step involves Gaussian smoothing, which as mentioned in section 2.6.2

reduces noise in the image. This is necessary as the consecutive step involves directional differentiation, which are very sensitive to noise. Figure 3.6 shows the effect of smoothing with increasing value of the standard deviation σ .

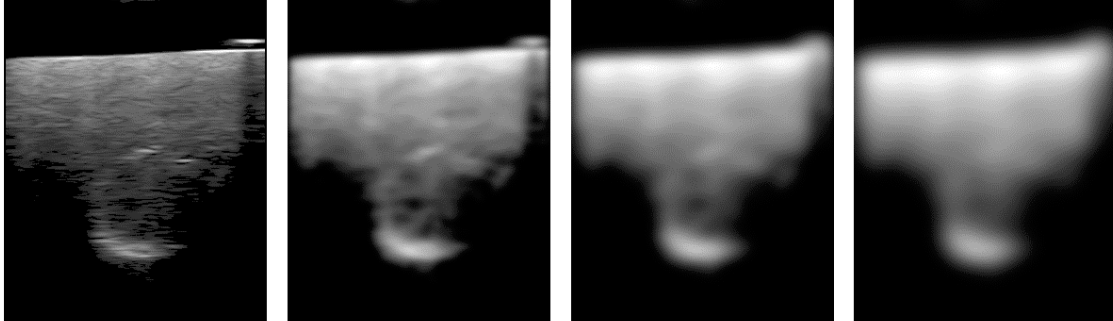


Figure 3.6: Effect of Gaussian smoothing. The left image is thresholded with $T > 44$. The following images is smoothed by discrete convolution of a Gaussian function with standard deviation $\sigma = \{0.5, 1, 1.5\}$ and the input image. Used to suppress noise.

To identify the edges the gradient magnitude of the smoothed images are calculated. The effect is shown in figure 3.7 for the same standard deviations as above.

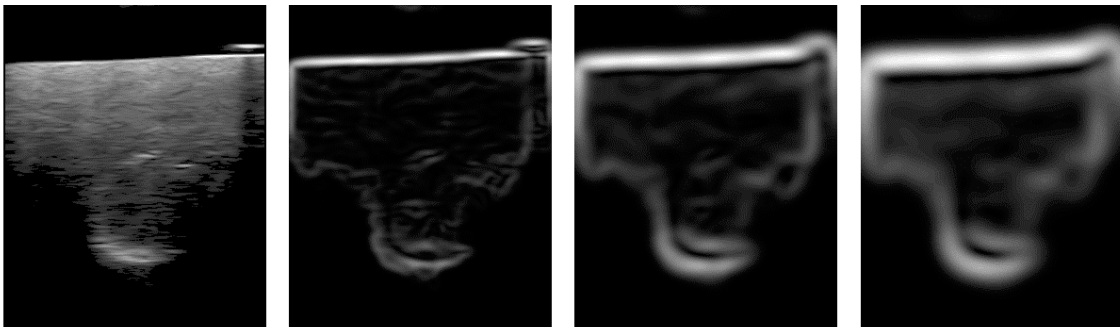


Figure 3.7: Effect of differentiating the Gaussian smoothed images and calculating the Gradient magnitudes. This serves as an edge detection filter. The left image is thresholded with $T > 44$, while the following images are the gradient magnitudes of the Gaussian smoothed images with standard deviation $\sigma = \{0.5, 1, 1.5\}$.

The edges can be used to produce an accumulator image, which corresponds to summarizing the number of intersection of drawn circles, as described in section 2.6.4. In this case it was known that the sphere had a radius of approximately 7.5 mm, and the search interval is set based on that. This is knowledge that to a certain degree is unknown, but qualified guess work is often possible in this context. The accumulator image generated performing Hough transform on the edge filtered image is shown in figure 3.8. The circles search is performed with an increasing radius interval from left to right.

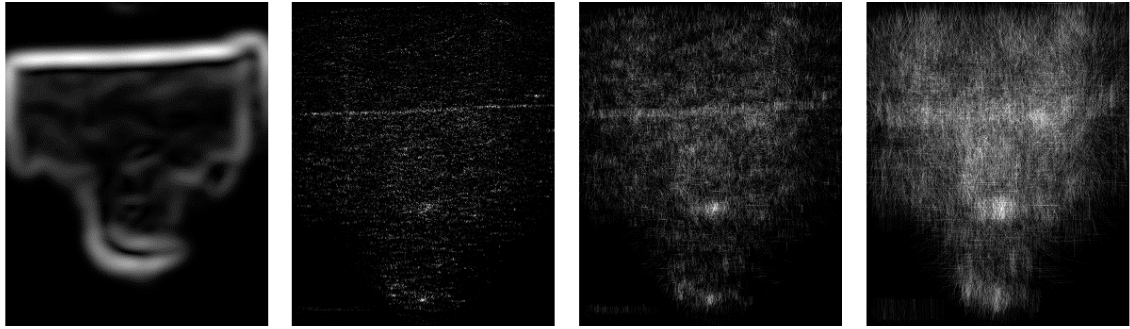


Figure 3.8: Hough transformation accumulator image for circle detection. The leftmost image is the Gradient magnitude of the Gaussian smoothed image with standard deviation $\sigma = 1$. The accumulator image intensity is formed drawing circles with a defined radius on the leftmost image. The intensity values is the number of circle intersections in each pixel making the maxima the most probable circle center. The second image is the result of a search from a circle with radius of 7.5 mm. The last images are formed by searching for circles in the intervals 7 mm - 8 mm and 6 mm - 9 mm respectively, where every vote is added to the same image. The size of the circular tumor model is estimated to equal 15 mm from a CT image.

The accumulator images is Gaussian smoothed with a standard deviation $\sigma = 1$, as seen in figure 3.9. The number of potential circles increases intuitively with larger interval, as can be seen. The brightest spot is then considered a measurement of the circle center.

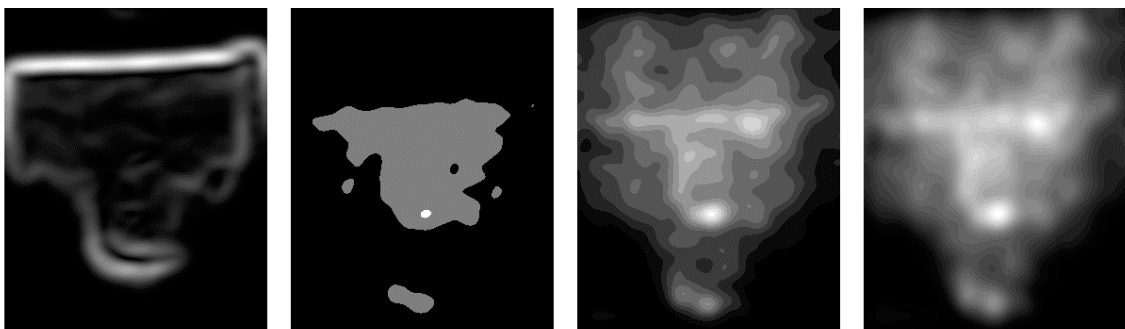


Figure 3.9: Effect of Gaussian smoothing of the accumulator images. The leftmost image is the Gradient magnitude of the Gaussian smoothed image with standard deviation $\sigma = 1$. The following images is Gaussian smoothed images searching for circles with radius of 15 mm, 14 mm - 16 mm and 12 mm - 18 mm respectively. The results are topographic images where the brightest spot is the most probable circle center.

The pseudo code below summarizes the implementation for finding a circle in the two-dimensional ultrasound image by utilizing the Hough transform.

Algorithm 3 Search for circle in a two-dimensional US image

Require: A digital image $I(u, v)$ and approximation of radius sphere r .

procedure SEARCHFORMOSTPROBABLECIRCLE(I, r)

if circle is detected in previous image **then**
 $c_i \leftarrow [u_0 \pm \frac{\Delta c}{2}, v_0 \pm \frac{\Delta c}{2}]$ ▷ Four corner points

 Crop image inside square of calculated corners c_i ▷ Figure 3.4
end if
for every pixel (u, v) **do**
if pixel intensity value $I(u, v)$ is non zero **then**
 $I_\Sigma \leftarrow I_\Sigma + I(u, v)$ ▷ Sum of intensity of non zero pixels
 $n_\rho \leftarrow n_\rho + 1$ ▷ Number of non zero intensity pixels
end if
end for
 $I_\tau(u, v) \leftarrow (I \times S_\tau)(u, v) \mid \tau > \frac{I_\Sigma}{n_\rho}$ ▷ Threshold image, equation (2.26)
 $I_G(u, v) \leftarrow (I_\tau * G_\sigma)(u, v)$ ▷ Gaussian smoothing, equation (2.27)
 $I_\nabla(u, v) \leftarrow \nabla I_G(u, v)$ ▷ Edge detection, equation (2.28)

 Create accumulator image $I_{A|r}$ as described in section 2.6.4. ▷ Figure 2.5
 $I_{A|r, G}(u, v) \leftarrow (I_{A|r} * G_\sigma)(u, v)$ ▷ Gaussian smoothing, equation (2.27)
 $I_{A, max} \leftarrow \max\{I_{A|r, G}(u, v)\}$ ▷ Find maximum intensity
 $u_h, v_h \leftarrow I(u_h, v_h) = I_{A, max}$ ▷ Find position of maxima
return (u_h, v_h) and $I_{A, max}$
end procedure

As the circle center measurements might contain both noise and uncertainties, a Kalman filter is applied at the end of the pipeline to get a statistical optimal value when tracking a moving sphere. The brief theoretical background describing the filter is presented in section 2.6.5.

To design the Kalman filter the first step would be to determine the state variables and transition function. For simplicity only one direction is considered, but the actual implementation is according to the two-dimensional images. The expansion to two dimensions is straightforward, but the matrix dimensions makes a literature presentation tedious. A predicted state model $\hat{X}_{k|k-1}$ for optimal localisation of the circle center assuming constant acceleration is given as

$$\underbrace{\begin{bmatrix} \hat{u}_k \\ \hat{\dot{u}}_k \\ \hat{\ddot{u}}_k \end{bmatrix}}_{\hat{X}_{k|k-1}} = \begin{bmatrix} \hat{u}_{k-1} + \hat{\dot{u}}_{k-1} + \frac{\hat{\ddot{u}}_{k-1}}{2} \\ \hat{\dot{u}}_{k-1} + \hat{\ddot{u}}_{k-1} \\ \hat{\ddot{u}} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & \frac{1}{2} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{A}_k} \underbrace{\begin{bmatrix} \hat{u}_{k-1} \\ \hat{\dot{u}}_{k-1} \\ \hat{\ddot{u}}_{k-1} \end{bmatrix}}_{\hat{X}_{k-1|k-1}}, \quad (3.2)$$

where the state variable $X = [u \ \dot{u} \ \ddot{u}]^T$ include the position, velocity and acceleration of the center. Noticeably, this can be interchanged with the Cartesian coordinates or other linearly related reference systems. The control input from equation (2.30) is excluded here, but can eventually prove useful for more complex scenarios where the tracked object is moving in a recognized pattern, i.e from respiration, and thus entailing an external force on the system. The transition matrix A_k is also used to estimate the predicted covariance as given by equation (2.31). The process noise covariance Q is assumed directional uncorrelated, and a diagonal matrix with variance set to 0.1 in normalized squared coordinates is used.

The output of the Hough transformation, i.e the measured circle center coordinates, represents the Kalman input measurement Y_k as given by equation (2.29). Mapping the state into measurement space is accomplished setting the measurement matrix $C_k = [1 \ 0 \ 0]$. The measurements normalized noise covariance forming R_k is set to unity. These parameters constitute the necessary building blocks of running the entire filter and receiving a optimized estimate for the circle center position. The returned values are then used to visualize the segment, and as a basis of cropping the next acquired image. The pseudo code for the filter given an estimate of the circle center from the Hough transform is seen in algorithm 4.

Algorithm 4 Kalman filter on circle center with noise

Require: The measurement of the circle center (u_h, v_h) with possible noise.

procedure FINDOPTIMIZEDCIRCLECENTER((u_h, v_h))

if Kalman filter is uninitialized **then**

 Predict previous state $\hat{X}_{k-1|k-1}$ and covariance $P_{k-1|k-1}$

end if

$\hat{X}_{k|k-1} \leftarrow A_k \hat{X}_{k-1|k-1}$ ▷ Predicted state, Equation (3.2)

$P_{k|k-1} \leftarrow A_k P_{k-1|k-1} A_k^T + Q_k$ ▷ Predicted covariance, Equation (2.31)

$\tilde{Y}_k \leftarrow [u_h \ v_h]^T - C_k \hat{X}_{k|k-1}$ ▷ Measurement residual, Equation (2.32)

$K_k \leftarrow P_{k|k-1} C_k^T (C_k P_{k|k-1} C_k^T + R_k)^{-1}$ ▷ Kalman gain, equation (2.33)

$\hat{X}_{k|k} \leftarrow \hat{X}_{k|k-1} + K_k \tilde{Y}_k$ ▷ Corrected state estimate, equation (2.34)

$P_{k|k} \leftarrow (I - K_k C_k) P_{k|k-1}$ ▷ Corrected covariance, equation (2.35)

 Store $\hat{X}_{k|k}$ and $P_{k|k}$ for next iteration

$(u_0, v_0) \leftarrow (\hat{u}_{k|k}, \hat{v}_{k|k}) \mid \hat{X}_{k|k} = [\hat{u}_{k|k} \ \hat{v}_{k|k} \ \dots]^T$

return (u_0, v_0)

end procedure

The Kalman filter is applied while tracking a structure. Both the circle detection

and the Kalman filter calculations are threaded outside the application main thread, and utilizes the CPU. The matrix manipulation is performed using Eigen.

3.4.1 Parameters

As shown, several parameters are used throughout the image processing pipeline. A summary is provided in table 3.1. The radius is the only modifiable variable in the clinical GUI, while the rest are set in the technical back end. The full covariance matrices of the Kalman filter is omitted, but the initial and normalized diagonal variance of the process and measurement matrices are given. These values must of course be optimized for various scenarios.

Table 3.1: Summary of parameters used in experiments. The radius is varied, but is usually set in the interval 5 mm - 35 mm. SI units are given when applicable, but in implementation pixel/image units is used interchangeably knowing the scaling factor.

Description	Interaction	Symbol	Value	Unit
Radius of potential spheres	Clinical	r	$r_{\min} - r_{\max}$	[mm]
Standard deviation Gaussian	Technical	σ_g	1	[mm]
Image threshold	Technical	τ	0-255	
Maximum sphere motion per frame	Technical	Δd	10	[mm/s]
Process variance	Technical	σ_Q^2	0.1	
Measurement variance	Technical	σ_R^2	1	

3.5 Integrated robot arm manipulation

Desired functionality of the robot framework includes the possibility of manipulating the robot arm extended with a tool based on external input sources. Ultrasound probes are examples of such tools, and for this project this is exclusively used. Herein integration of moving the center of the probe head using the robot is described. On a high level, the robot framework, or controller for that matter, understands motion of the end-effector with respect to the robot base. Most commonly input measurements occurs with respect to other coordinate frames, which in turn requires extensive translation using the relation mapping described in section 3.2. A typical example is the desire to move a tool to a specific pose in the patient reference, as can be understood from figure 3.2. Given a transformation \mathbf{T}_t^{pr} it is possible to determine the required base to end effector transformation \mathbf{T}_{ee}^b using the predetermined calibration matrices, i.e

$$\mathbf{T}_{pr}^b \mathbf{T}_t^{pr} \mathbf{T}_{ee}^t = \mathbf{T}_{ee}^b.$$

Offset transformations are relevant for probe motion with respect to ultrasound image analysis, and is possible post-multiplying \mathbf{T}_t^{pr} with the offset defined in tool

space.

After determination of end-effector motion in the robot base frame, we can finally start giving the robot arm commands. Details of timing law synchronisation and trajectory generation is performed using the robotic framework and robot controller. A conclusion from prior development was that sequential point-to-point motion using the UR5 manipulator resulted in jerk. Jerk, or the time-derivative of joint acceleration, influences how smooth and efficient the robot moves. A method introduced in the specialization project was proved to minimize this unwanted behaviour by utilizing velocity kinematics. Sequential manipulation of the robot arm consequently happens in the joint velocity space, i.e the input to the controller system is the joint velocity $\dot{\mathbf{q}}$. The issue with jerk is not relevant for single movements, which usually is performed in configuration space.

To find the necessary expression, we need to invert the Jacobian of equation (2.10). This gives

$$\dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \boldsymbol{\nu}, \quad (3.3)$$

where the Jacobian matrix \mathbf{J} is a function of the joint configuration \mathbf{q} . The Jacobian is calculated for every state update in the framework. To determine the joint velocity, we need to develop a velocity vector $\boldsymbol{\nu} = [\dot{\mathbf{p}} \ \boldsymbol{\omega}]$. This can for instance be achieved using equations (2.12) and (2.14), and generating a trajectory based on interpretation of configuration space. A parent pseudo code for motion between arbitrary poses with respect to a coinciding coordinate frame is given in algorithm 5.

Algorithm 5 Move robot tool from pose \mathbf{T}_i to \mathbf{T}_f

procedure MOVEFROMFRAMETOFRAME($\mathbf{T}_i, \mathbf{T}_f, v_c$)

$$\mathbf{R}_f^i \leftarrow (\mathbf{R}_i)^T \mathbf{R}_f$$

$$v_f, \mathbf{k} \leftarrow \text{Angle-axis representation of } \mathbf{R}_f^i \quad \triangleright \text{Equation (2.3) and (2.4)}$$

$$\mathbf{t}_f^i \leftarrow \frac{[\mathbf{t}_f - \mathbf{t}_i]}{\|\mathbf{t}_f - \mathbf{t}_i\|}$$

$$\mathbf{t}, v \leftarrow \text{Append desired timing laws on } \mathbf{t}_f^i \text{ and } v \in [0, v_f]$$

$$\boldsymbol{\nu} \leftarrow [(v_c \mathbf{t} \ \mathbf{R}_i \dot{v} \mathbf{k})]^T \quad \triangleright \text{Equation (2.12), (2.13) and (2.14)}$$

$$\dot{\mathbf{q}} \leftarrow \mathbf{J}^{-1}(\mathbf{q}) \boldsymbol{\nu} \quad \triangleright \text{Equation (3.3)}$$

while not at \mathbf{T}_f **do**

 Move joints according to $\dot{\mathbf{q}}$

end while

end procedure

The algorithm can be used to move the robot arm based on input from any external

source with pose information spatially related to the patient reference. This can be utilized when implementing framework functionality allowing robot response based on ultrasound image analysis. Figure 3.10 shows a structure in the ultrasound image with a labeled centroid.

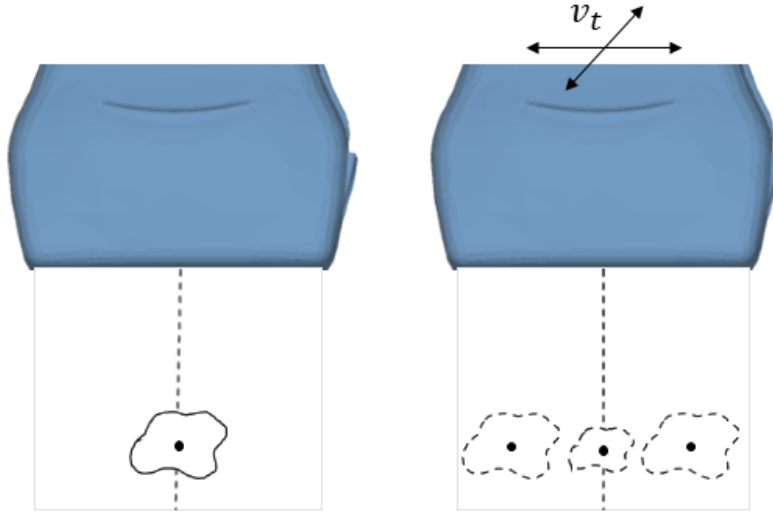


Figure 3.10: Illustration of the probe with respect to a moving structure in the ultrasound scene. The goal is to keep the centroid, here sketched as a black dot, in the azimuth center of the imaging plane. This is achieved manipulating the probe with the robot arm based on image analysis.

Suppose that this structure can move as shown, either in the image plane with approximately constant area or out-of-plane motion with variable area. Using the robot arm it should be possible to servo the probe in order to keep the structures centroid along the radial direction. In practice, the probe has restricted motion in this direction due to acoustic contact on a surface barrier. This leaves horizontal in-plane and perpendicular out-of-plane motion. The position of the structure with respect to the probe head is a result from the image analysis and therefore used as a basis for the robot motion. The transformation from robot base to structure can be derived using figures (3.1), (3.2) and (3.3), giving two transformations paths, one using optical tracker and the other the internal robot sensors. This is summarised as

$$\mathbf{T}_{\text{str}}^b = \mathbf{T}_{ee}^b \mathbf{T}_t^{ee} \mathbf{T}_{\text{img}}^t \mathbf{T}_{\text{str}}^{\text{img}} = \mathbf{T}_{pr}^b \mathbf{T}_t^{pr} \mathbf{T}_{\text{img}}^t \mathbf{T}_{\text{str}}^{\text{img}}. \quad (3.4)$$

Moving the probe head over the structures centroid should thus be performed by servoing to the given transformation, while keeping the axial distance between head and centroid constant. Eventually, if the surface is known this should be included in the manipulation for determining the axial displacement between probe head and structure centroid. If the structure moves perpendicular to the image plane its area

in the image plane will eventually vary. In this specific scenario a sphere is tracked, making the area proportional to the calculated radius. If the structure is identified, the robot slightly moves to determine the gradient of the structures radius while searching for the largest radius. The imaged structure with largest radius is used as the pinpoint, and will serve as the desired ultrasound section. A pseudocode for centering the probe over a circular structure utilizing several of the algorithms presented earlier is presented in algorithm 6.

Algorithm 6 Center probe over structure

Require: Digital image $I(u, v)$ and radius search interval $[r_{\min}, r_{\max}]$

procedure CENTERPROBEOVERCIRCULARSTRUCTURE($I, [r_{\min}, r_{\max}]$)

for every r between r_{\min} and r_{\max} with a suitable increment **do**

$(u_h, v_h), I_{A, \max} \leftarrow \text{SEARCHFORMOSTPROBABLECIRCLE}(I, r)$

 Determine proposed center and radius r_s based on maximum $I_{A, \max}$

end for

$(u_0, v_0) \leftarrow \text{FINDOPTIMIZEDCIRCLECENTER}(u_h, v_h)$

$\mathbf{t}_{str}^I \leftarrow [u_0 \quad v_0 \quad 0]^T$ ▷ Coordinates of center

 Create \mathbf{T}_{str}^t ▷ Probe specific transform from image to physical space

 Create \mathbf{T}_{str}^b ▷ Equation (3.4)

 MOVEFROMFRAMETOFRAME($\mathbf{T}_t^b, \mathbf{T}_{str}^b$)

return r_s

end procedure

The current transformation between base and tool \mathbf{T}_t^b is updated in real-time. The returned radius r_s is used to center the probe over the structures largest cross-section. The proposed method involves slight motion of the probe along the direction of elevation, recalculating the proposed radius and placing the probe such that the differentiation of the radius is negative to either side. The motion is defined according to accuracy in the system.

3.6 Visualization

Improving the visualization of the robot arm was issued in the specialization project. This can be understood by figure 2.6, where the center of the robots end-effector is displayed as the tip of a grey cone. The manipulator is deformable in a sense that it consists of six rotational joints with rigid links. A computer-aided drafted(CAD²) model of the arm is provided by the manufacturer, and this was manually sub-structured using CAD software into links according to the geometric representation

²Use of computer programs to create constructions and technical drawings of physical objects

drawn in figure 2.4. Mutual relationships between sub-models is developed by deriving the transformation between the robot base and every joint frame using the Denavit-Hartenberg convention. Real-time display of the manipulator in the navigation window is implemented using the external library VTK, which is described in section 2.7.2. Every link model is placed correctly in navigation space and position and orientation is calculated in real-time based on decoded state updates from the robot controller. Pictures showing the implemented results is used throughout the rest of this thesis.

3.7 Experimental validation

The verification experiments are divided into three parts, and are used to identify the system performance and accuracy in order to validate the implementations described above. First the calibration routine and image analysis is investigated, which both constitute a building block in the integrated system. To examine the integrated robot system three experiments will be conducted. They are chosen with the goal of covering several components interacting at the same time. Simple protocol schemes are used to describe the experiments, to enable easier reevaluation.

All experiments are carried out at the same workstation using a computer installed with an Ubuntu 14.04 operating system. The hardware consists of an Intel(R) Core(TM) i7-6700K CPU with 4.00 GHz, 32 GB RAM and a NVIDIA GeForce GTX 980 Ti GPU unit. The robot arm is an UR5 from Universal Robots with six revolute joints. Additionally, the calibration process and tool tracking utilizes the optical tracker Polaris Spectra from NDI Medical. The ultrasound imaging is performed with the ultrasound scanner SONIX MDP from Ultrasonix. Additional materials and equipment used for specific experiments will be mentioned in the protocol.

3.7.1 Calibration procedure

Purpose. Investigate the precision in the calibration process and the necessary number of measurements for satisfactory system accuracy. Potentially relevant information also includes the variance between calibrations and the time consume.

Method. The procedure involves several systematic and corresponding measurements of \mathbf{T}_t^{pr} and \mathbf{T}_{ee}^b in different positions and orientations. The calibration routine is performed in an area equal to a clinical workspace volume. The routine is run several times to evaluate robustness.

Data interpretation. Post measurement data interpretations involves plots of error metrics, such as the residual sum of squares of the orientation, the orientation and translational error versus number of measurements. The translational compo-

nents are investigated separately. The time consume running the procedures is also included.

3.7.2 Image processing and analysis

Purpose. Investigate the speed and accuracy of the image processing and analysis to verify real-time requirement and robustness for robot manipulation. The actual size of the sphere will be compared to estimates from calculations.

Method. The procedure will involve a comparison between the circle detection from image analysis and the physically registered sphere using a calibration arm. The arm is placed in a water tank, as seen in figure 3.11, and tracked by the optical camera. Two transformation paths from the sphere to the tool is generated, one via the calibration arm and patient reference, and the other from image to tool.

Data interpretation. Visualization of segmented sphere spatially projected onto the image plane. Difference in center location and size will be calculated, as well as the processing time.

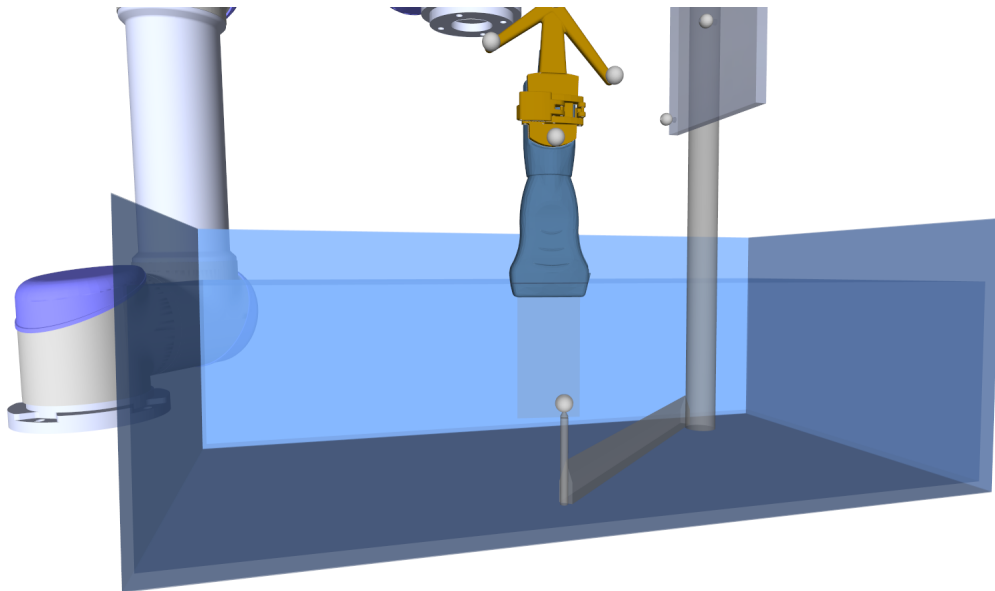


Figure 3.11: Experiment setup for validation of image processing and analysis. A calibration arm is placed in a water tank together with an imaging ultrasound probe. The sphere location and size in physical world is compared to the information obtained from image analysis.

3.7.3 Integrated robot manipulation

I. Visual servoing based on ultrasound image analysis

Purpose. Investigate the accuracy and robustness of autonomous centering of the probe over a identified structures centroid in the ultrasound image.

Method. A calibration arm is placed in a water tank as shown in figure 3.12. Initially, the probe is manually displaced with the structure randomly located in the ultrasound image. An autonomous visual servoing procedure is then conducted, with the goal of centering the probe over the centroid of the sphere using ultrasound image analysis.

Data interpretation. The final displacement between physically registered sphere and proposed location from image analysis. Comparison between tracking using robot controller or optical tracking system.

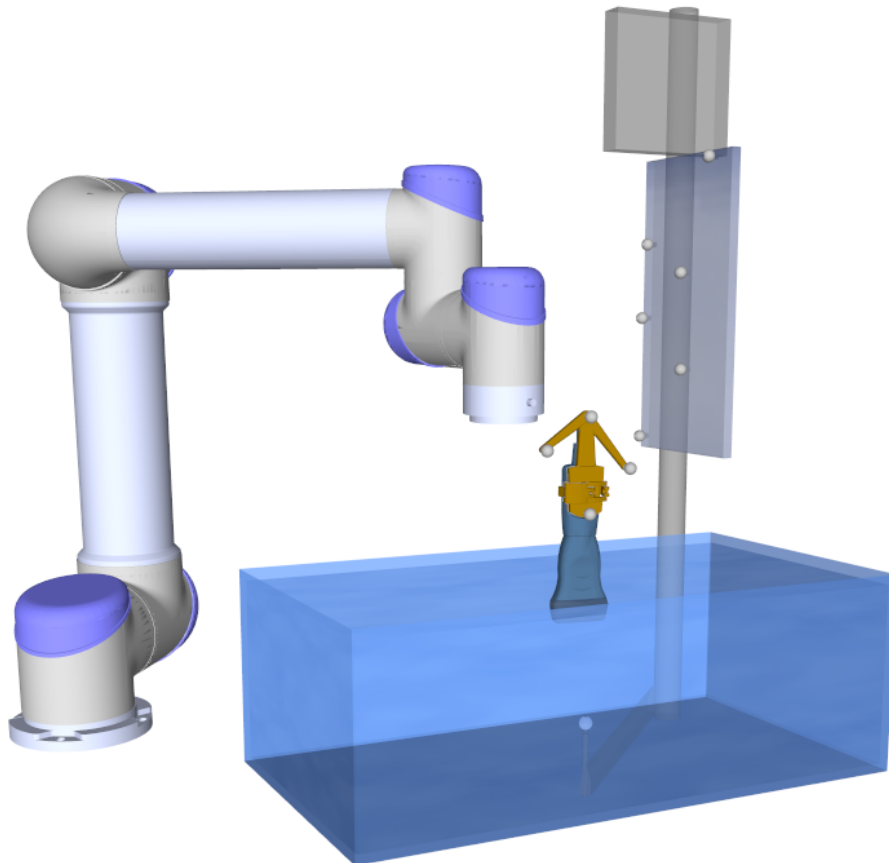


Figure 3.12: Illustration of the setup for validating visual servoing based on ultrasound image analysis. A calibration arm is placed in a water tank together with an imaging ultrasound probe held by the robot arm.

II. Mimic moving target: Wireless pointer

Purpose. Investigate how the robot mimics motion sequences from external input sources, in this case a wireless pointer.

Method. A wireless pointer is moved arbitrary in space and tracked as can be seen by figure 3.13. The robot moves the center of the probe head based on the tracked path.

Data interpretation. Comparison of planned path and actual motion of the probe head registered both using the internal robot sensors and the optical tracking system. Plot of predefined path together with path generated by robot motion. Investigation of distance between every point of sampled path from wireless pointer and the tracked probe.

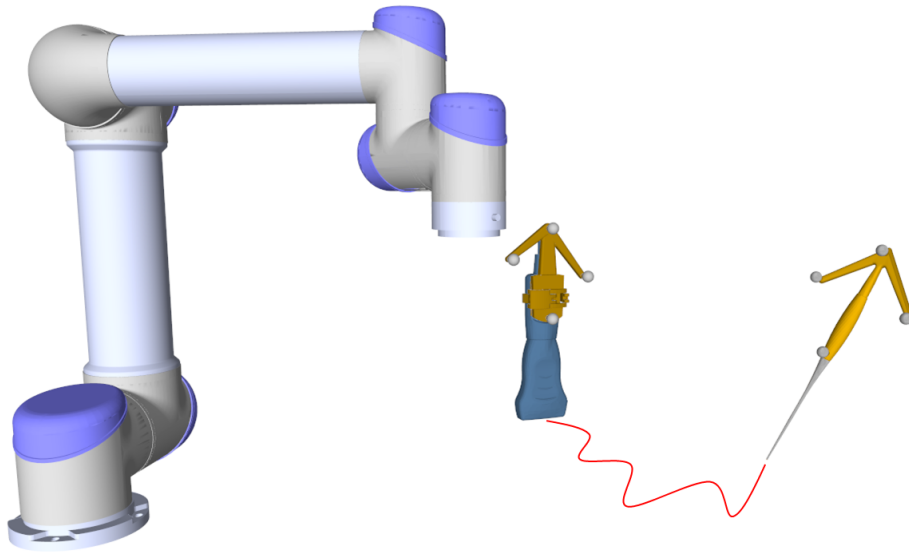


Figure 3.13: Illustration of robot extended with a tool mimicking the motion of a wireless pointer. The red curve is the tracked path from the pointing tip, which the robot arm follows.

III. Phantom scan

Purpose. Compare CT data with 3D reconstruction of an ultrasound scan performed by the robot arm. Investigate how location and size of segmented tumors from ultrasound are with respect to the CT.

Method. Predefined scan over a liver tissue phantom registered based on CT data as suggested by figure 3.14. 3D reconstruction is performed from acquisition using an existing method in CustusX known as the Pixel Nearest Neighbour [68]. Segmentation of proposed tumors in real-time using the circle detection method.

Data interpretation. Comparison of number of found tumors, CT is used as ground truth. A 3D reconstruction of the scan is constructed, together with plots of sum of image intensity and Hough accumulator intensity.

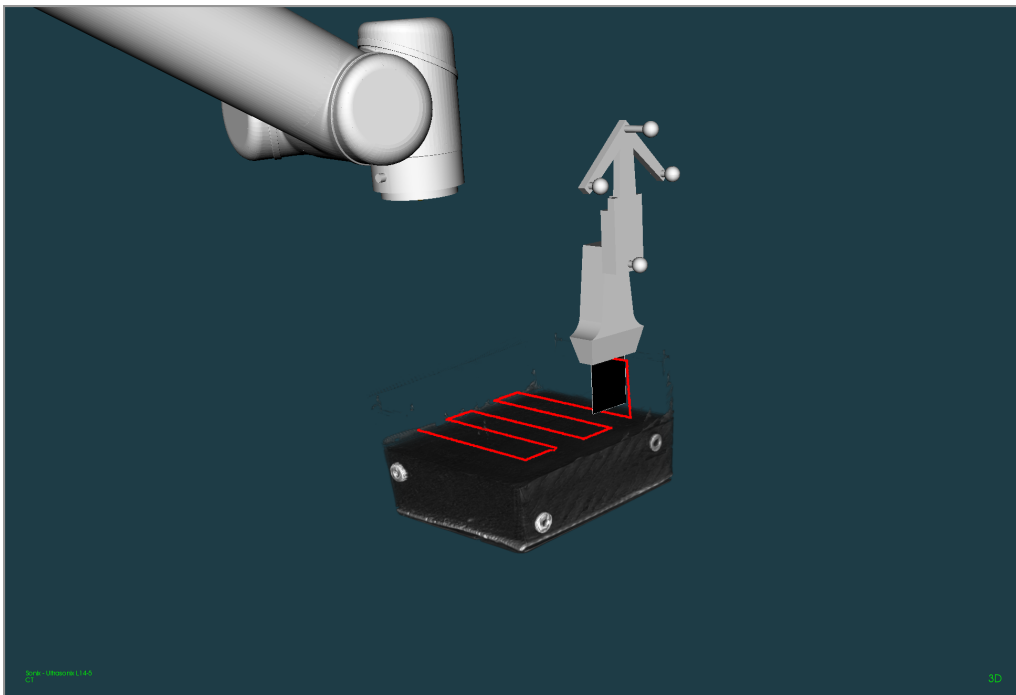


Figure 3.14: Illustration of a scan line on a liver tissue phantom in the navigation window of CustusX utilizing the robot framework. The robot arm is holding an ultrasound probe and is about to perform a predefined scan on the phantom. The liver phantom model is created from CT data.

Chapter 4

Results

In this chapter the key features of the robot framework and the results from the verification experiments is presented. The integrated robot framework is a product of extensive system development, with several distinct components. The implemented code is found following links in appendix A, with pseudo code for essential functionality presented in the previous chapter.

4.1 Calibration

The results from the calibration routine using either the separable or simultaneous approach can be divided into rotational and positional error metrics. In the following the behaviour of the different calibration methods as a function of pose measurements will be presented. The rotation error of a typical calibration routine is given in figure 4.1.

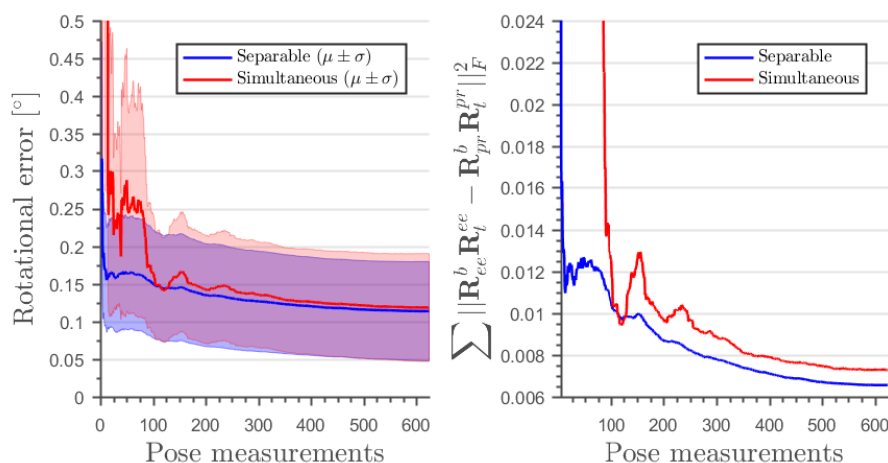


Figure 4.1: Rotational error metrics from the the separable and simultaneous calibration routine. The left figure shows the rotational error as a function of pose measurements with the shaded area corresponding to the standard deviation σ . The residual sum of squares of the orientation is shown to the right.

Residual sum of squares is often used in literature and are therefore included in the figure. In this calibration routine, 620 measurements was conducted in an volumetric area of approximately 6 dm^3 . The three-dimensional translational error from the same routine is shown in figure 4.2 for both the separable and simultaneous approach.

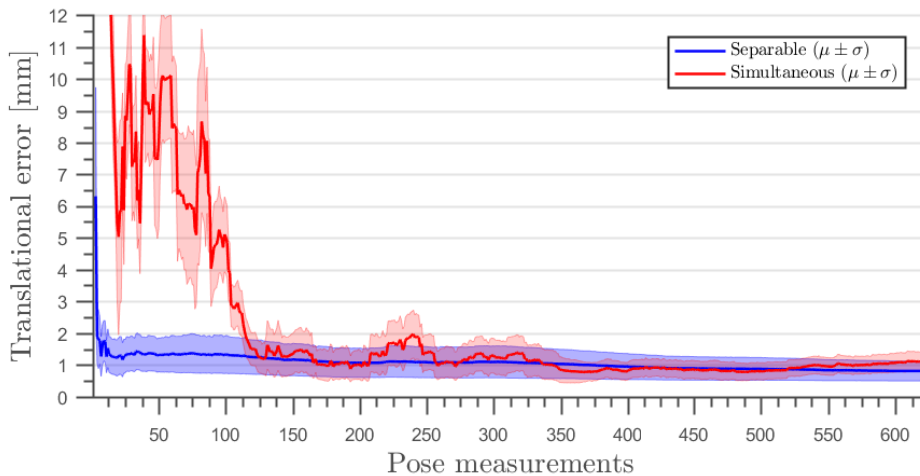


Figure 4.2: The positional error as a function of pose measurements. The blue and red line are results from the separable and simultaneous calibration methods respectively. The shaded regions corresponds to the area bounded by the standard deviation σ of the measurements.

To gain more detail about the directional components of the translational error they are evaluated separately, and shown in figure 4.3. A running average of ten measurements is applied to attenuate fluctuation in each component.

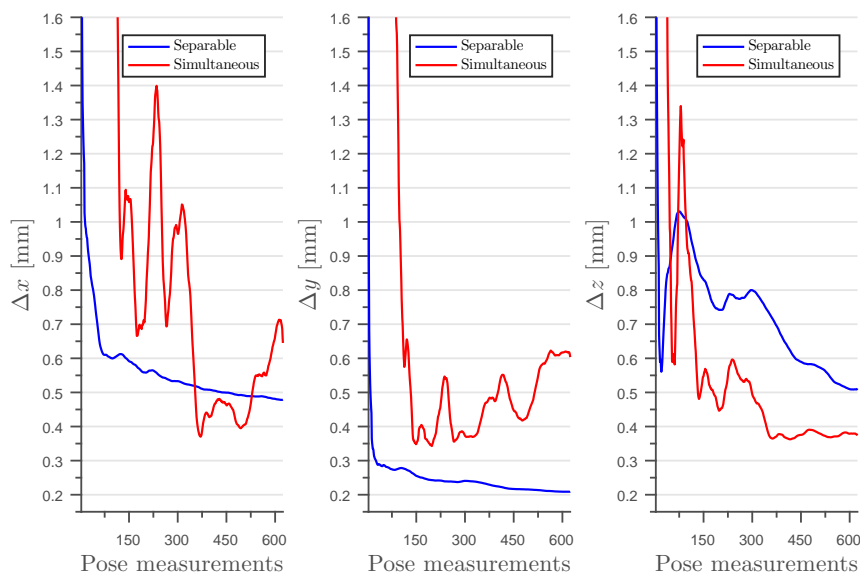


Figure 4.3: The translation error of the components x, y, z from left to right as a function of pose measurements. The red and blue lines are from the simultaneous and separable calibration methods respectively. A running average of ten measurements is applied.

As a summary of the calibration routine, key values after 620 pose measurements is given in table 4.1.

Table 4.1: Summary of the results from two calibration routines after 620 measurements.

Information	Separable	Simultaneous
Rotational error [°]	0.11 ± 0.07	0.12 ± 0.07
Sum of squares of Frobenius norm	0.0066	0.0073
Translational error [mm]	0.82 ± 0.31	1.05 ± 0.29
Δx [mm]	0.48 ± 0.30	0.65 ± 0.37
Δy [mm]	0.21 ± 0.17	0.60 ± 0.27
Δz [mm]	0.51 ± 0.34	0.37 ± 0.25
Relative error	0.0018	0.0022

Additionally, series of calibrations are conducted with equal number of pose measurements to investigate variance in the routine. The calibration error from ten consecutive runs with 620 measurements each is shown in figure 4.4 for both methods.

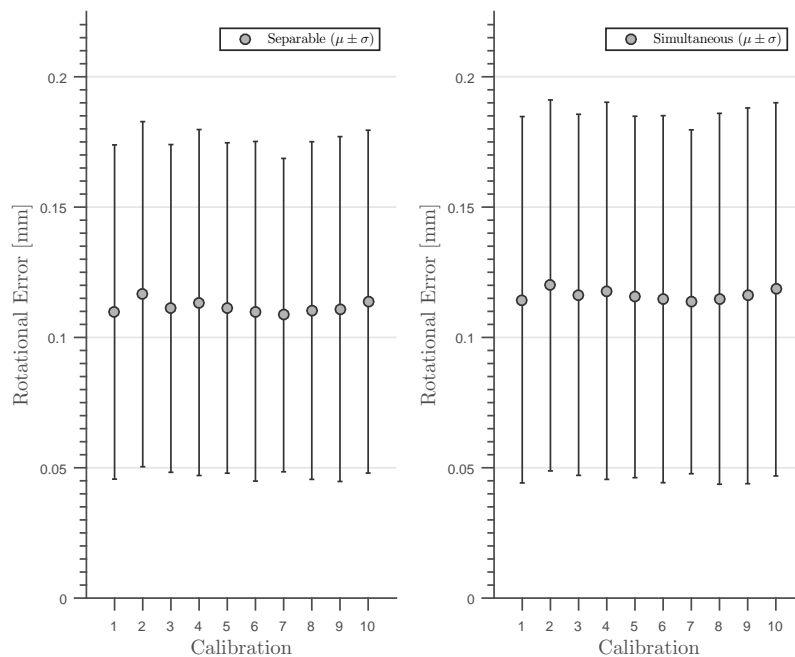


Figure 4.4: The rotation error with standard deviation of ten calibration routines. The left plot are the results from the separable method, while the right plot is the simultaneous method. 620 measurements was conducted for each calibration.

Similarly, the translational error from the same ten procedures is given in figure 4.5.

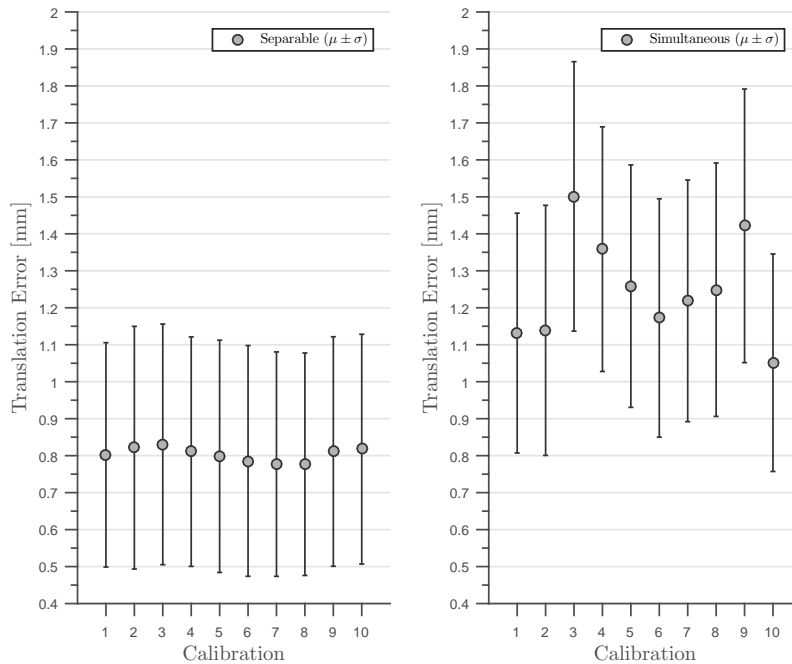


Figure 4.5: The translational error with standard deviation of ten calibration routines. The left plot are the results from the separable method, while the right plot is the simultaneous method. 620 measurements was conducted for each calibration.

The calibration time consume is also of clinical relevance. The calculations times is negligible compared to the motion routine, and individual investigation is thus omitted here. For this routine the measurements per minute is approximately 58, making the full routine last 11 minutes. This is of course very dependant of the manipulators motion parameters, such as velocity and acceleration, which are set as high as possible.

4.2 Image processing and analysis

In a visual servoing robot system, the image analysis is usually independent of the robot, but not the other way around. Image analysis can thus be investigated individually. Here, the speed and accuracy of the processing and analysis for a typical probe in water tank setup is presented. The imaged structure in the water tank is a sphere, as shown in figure 3.11, and the size from analysis is compared to actual size provided by the manufacturer.

Deviation between circle center location from image analysis and optically tracked sphere is shown in figure 4.6. Each measurement is an average of 170 measurements conducted in seven different locations.

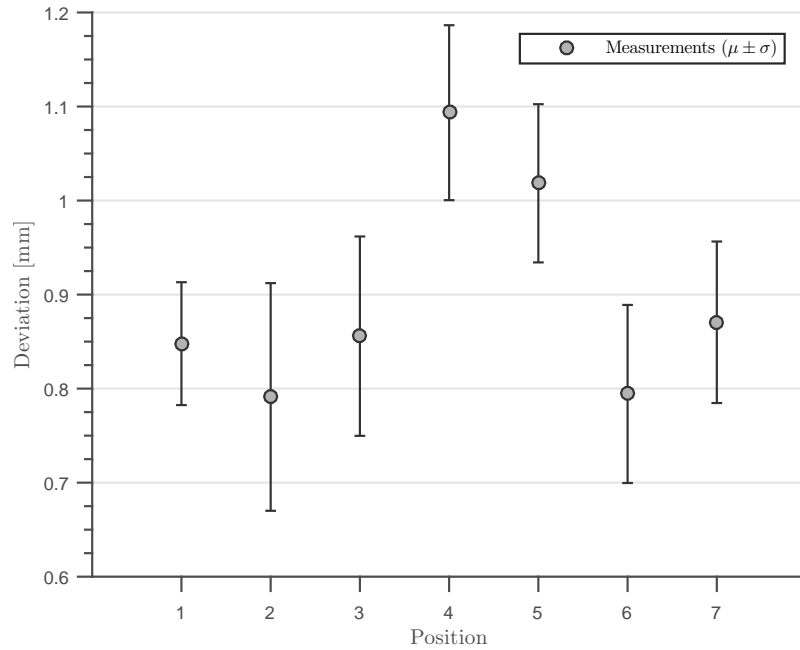


Figure 4.6: Plot of the deviation between circle center location from image analysis and optical tracked sphere (calibration arm). The mean and standard deviation is calculated from 170 measurements in seven different locations.

The estimated radius of the circle from image analysis compared with the real radius equal to (11.5 ± 0.1) mm, can be shown in figure 4.7.

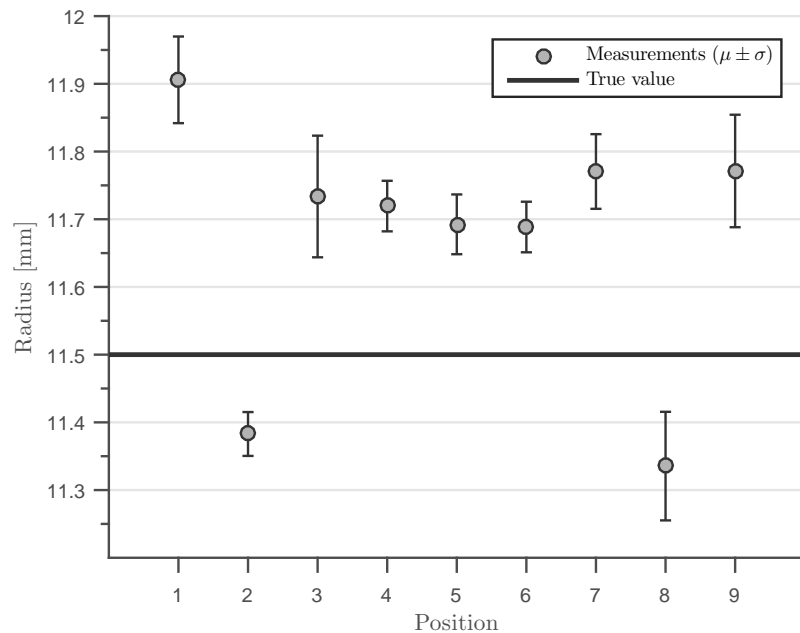


Figure 4.7: The estimated diameter from the the circle detection. The mean μ and standard deviation σ is calculated from 170 measurements in nine different locations. The true value $d_{\text{true}}(11.5 \pm 0.1)$ mm is provided by the manufacturer.

The real-time visualization of the process in the navigation window is seen in figure 4.8. Here, a white overlay on the proposed circle is added to the ultrasound image section. In the three-dimensional navigation window the robot holding an imaging probe is shown, with a red sphere segmentation corresponding to the estimated location and size is updated in real-time.

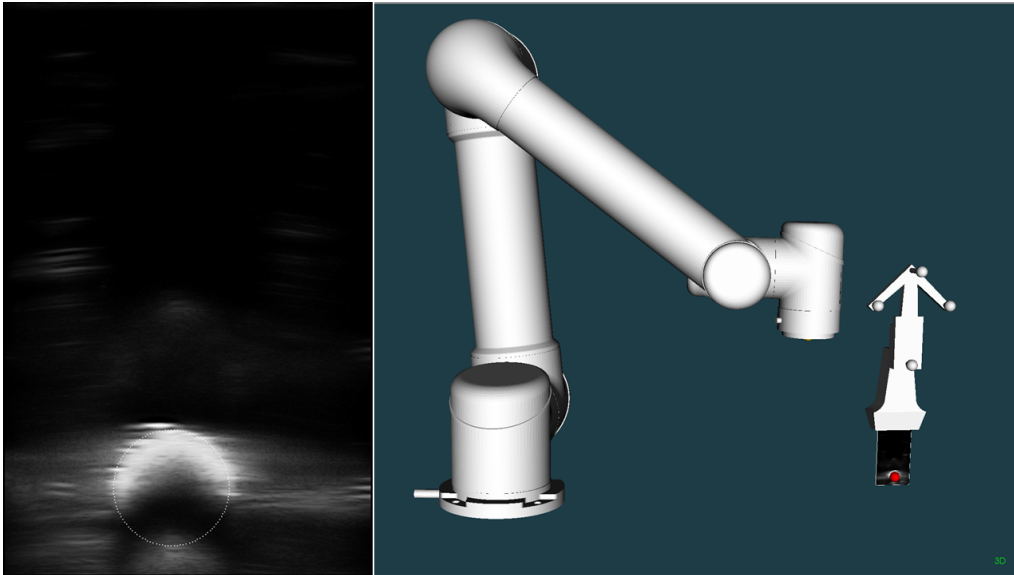


Figure 4.8: The segmentation of the circle on top of the original ultrasound image. The left image shows the proposed solution by a white circle, while the right right image shows the three-dimensional navigation window and the qualitative spatial relationship to the robot arm.

The real-time criteria is a rather vague definition. Here, we have assumed that the analysis satisfies the criteria if the time between frames is longer than the time of analysis. The main components influencing the speed in this pipeline is image size and intensity, and are thus investigated with the analysis runtime. Table 4.2 shows the result as an average of ten runs.

Table 4.2: Speed of analysis. The image size is given in pixels, which consists of 8 bit intensity components. The total intensity is the sum of every pixel intensity in the image, and each runtime measurement is an average of ten image analysis’.

Image size	Intensity [$\times 10^6$]	Runtime [ms]
360×600	3.3	86 ± 2
320×512	3.1	74 ± 2
300×440	2.6	57 ± 2
280×360	2.5	48 ± 2
256×256	2.3	39 ± 2

4.3 Robot manipulation

I. Visual servoing based on ultrasound image analysis

As mentioned earlier, visual servoing using ultrasound has the potential of being useful in several clinical applications. As a proof of concept, an autonomous method for centering a probe over a tracked structure in an ultrasound image was implemented, and the accuracy was investigated in the verification experiment. Figure 4.9 shows the results from the probe centering for thirty different cases measured using image analysis, optical tracking sensors or internal robot sensors.

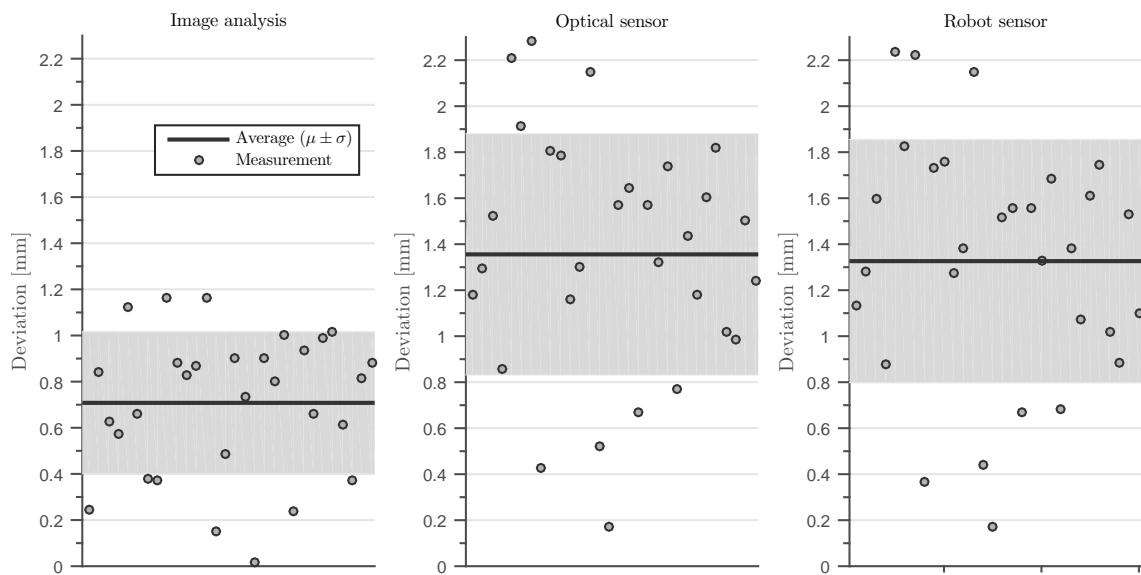


Figure 4.9: Results from the verification of probe centering functionality measuring the distance from the sphere to the probes axial axis using image analysis, optical sensors and robot sensors. The probe is placed such that the structures centroid is placed arbitrarily in the ultrasound image at least 10mm from the axial axis. The probe is aligned autonomously using the robot arm.

II. Mimic moving target: Wireless pointer

The previous section revealed results from a verification experiment where an ultrasound probe was centered over a tracked structure. The robot-arm was displaced while the structure was held in a constant position. In other scenarios, the structure would also be moving, and the goal of the visual servoing would be to mimic the moving target, and maintaining the visibility of the structure in the ultrasound image section. The concept of mimicking a moving target is of general nature, and here a wireless pointer is used to define the motion of the probe held by the robot-arm. Figure 4.10 shows a defined path by the wireless pointer and the tracked motion of the probe using both the optical and the robot sensor systems.

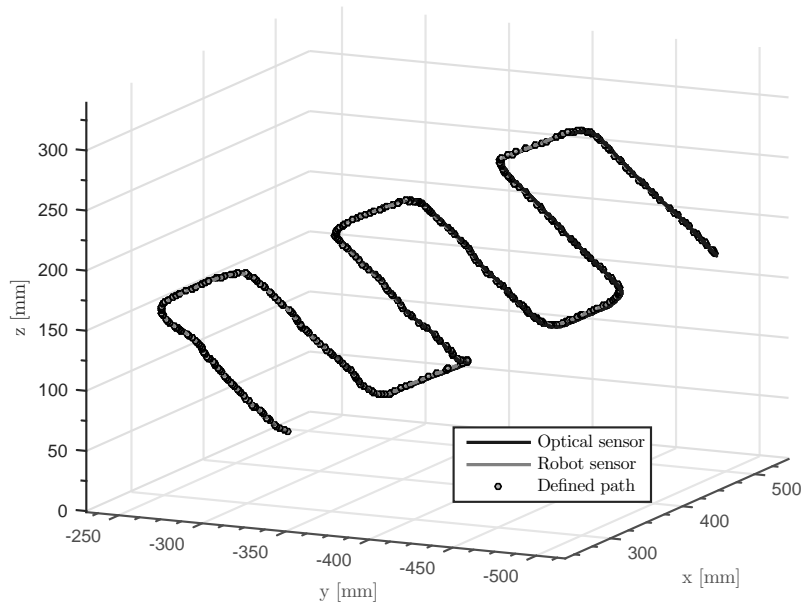


Figure 4.10: Defined path sampled from wireless pointer in space. The black and gray lines correspond to the probe center moving through the predefined points measured using the optical and robot based sensors respectively.

The deviation between the defined path and mimicking motion is shown in figure 4.11.

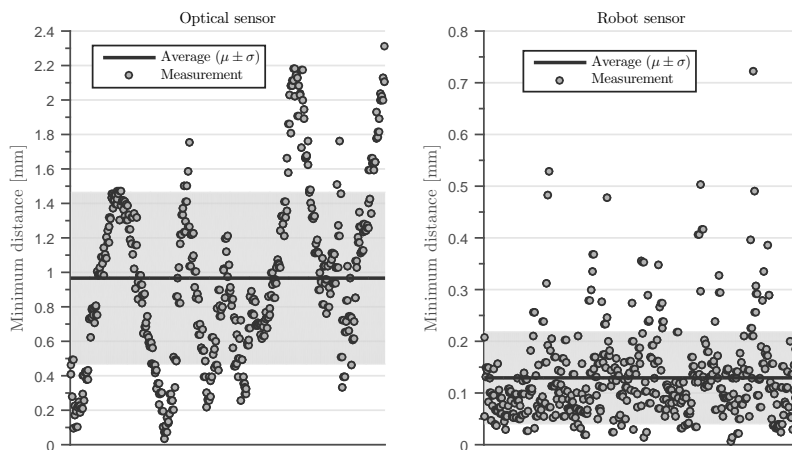


Figure 4.11: The minimum distance between every point on the sampled path from the wireless pointer and the closest point on the tracked probe serving through the points. Measured using both the optical tracker and the internal robot sensors.

For further investigation the standard deviation in static positions was calculated for both the optical and robot sensors. The probe was held by the robot arm, and kept still for approximately 4000 samples in ten different positions. For the optical tracker the standard deviation of the orientation was $(18.7 \pm 0.8) \text{ m}^\circ$, while the translation was $(0.21 \pm 0.04) \text{ mm}$. The same measurements conducted with the

robot tracking sensors gave a rotational deviation of $(0.3 \pm 0.3) \text{ m}^\circ$ and a positional deviation of $(3 \pm 3) \mu\text{m}$.

III. Phantom scan

An ultrasound scan over a liver tissue-mimicking phantom was conducted followed by a 3D reconstruction of the recorded data. An image of the navigation window after the scan is shown in figure 4.12. A real-time segmentation of one of the tumor models is shown as a red sphere in the reconstruction inside the ultrasound image sector. The right hand side shows the ultrasound image overlay on the CT data.

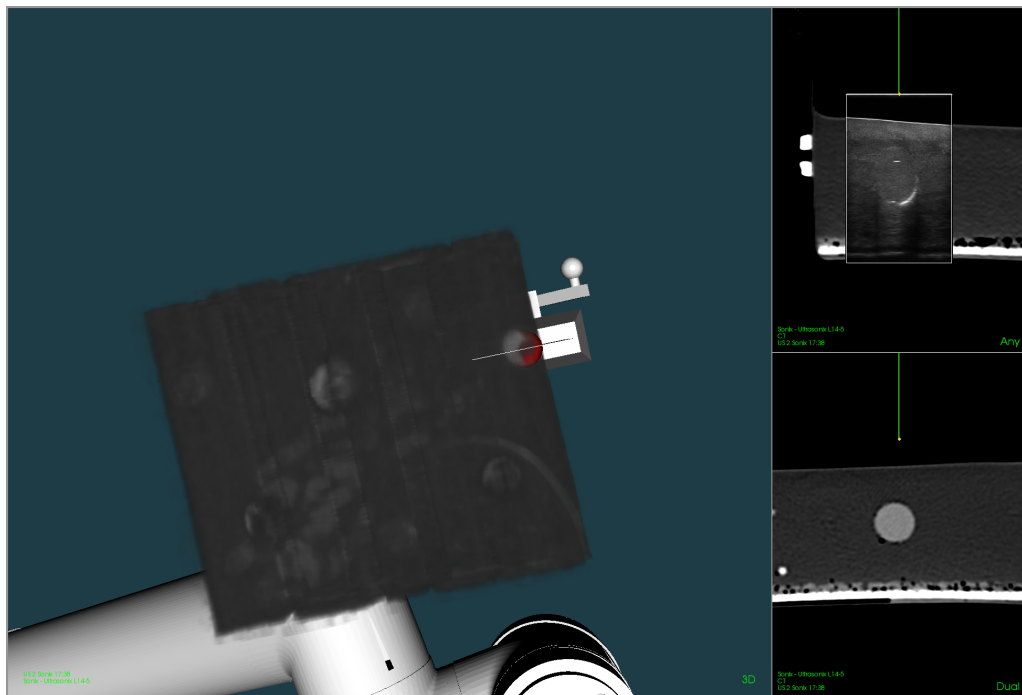


Figure 4.12: 3D reconstruction of ultrasound scan with the red sphere being the real-time segmented tumor in the ultrasound plane. Several tumor models are visible in the 3D view. The upper right image show the ultrasound plane overlay in the CT data, while the bottom right shows the CT data alone.

The scan was conducted with four scan lines, as illustrated in figure 4.13. By inspection seven tumor models was identified, which coincides with the CT data. The lower part of the figure shows a running average of the normalized total image intensity and the number of intersections from the Hough transformation. The peaks correspond to the tumor models found from inspection of the 3D reconstruction.

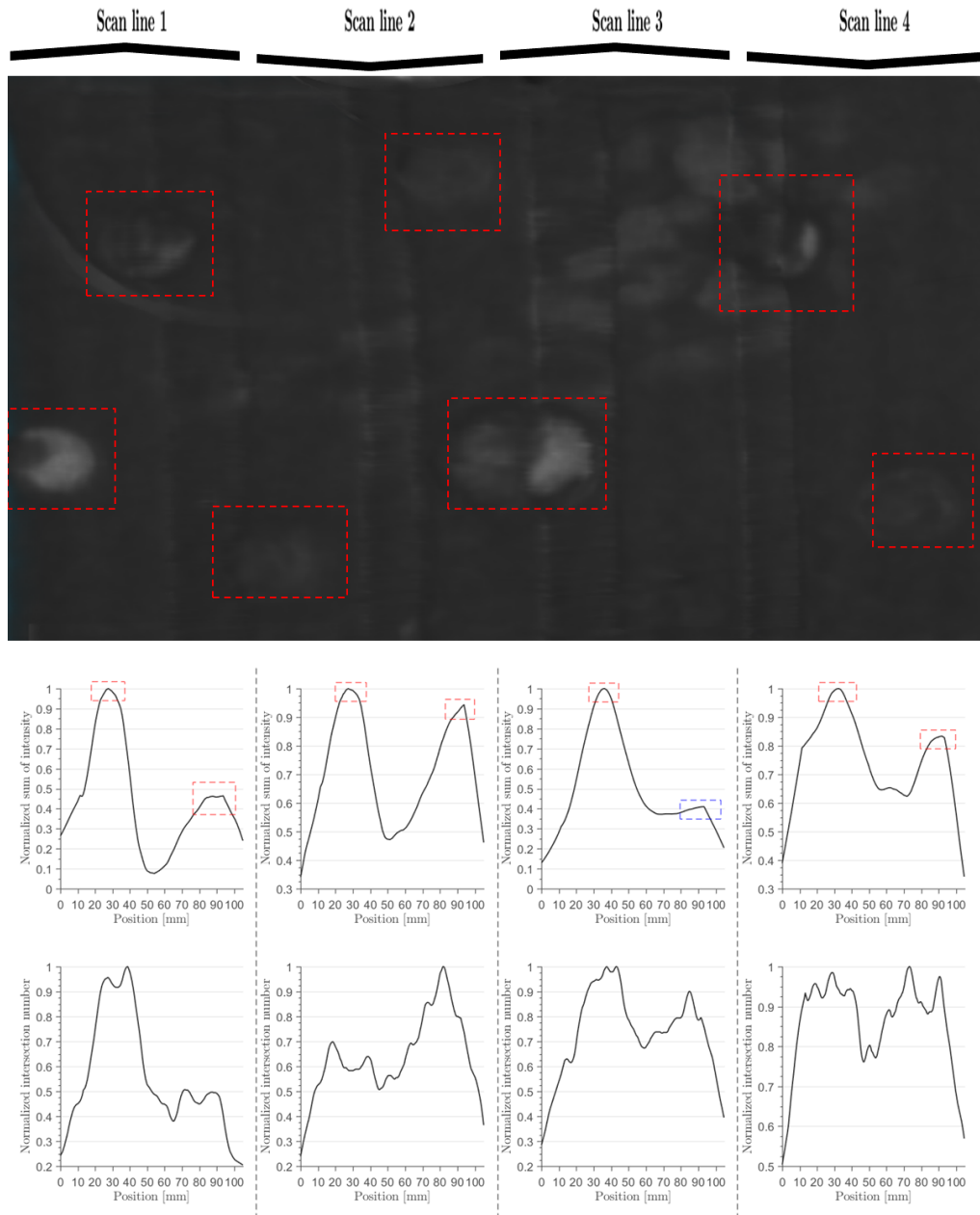


Figure 4.13: 3D reconstruction of a liver tissue phantom scan with approximately 100 mm length per scan line. The scan line direction is indicated. The upper plot is the normalized image intensity along the scan lines. The lower plot is the normalized maxima of the accumulator intensities from the Hough transformation. Both are smoothed with running averages. The red squares indicate abnormalities in the tissue.

The normalization is based on data from individual scan lines, which is approximately 10 cm. The motion of the probe is along the elevation direction, and the transition between the lines, i.e. the azimuth motion, is removed from the results to avoid incoherence.

4.4 The integrated framework

The verification experiments was carried out using the implemented framework, which naturally is considered the main result of this thesis. A summary of features extending the framework presented in section 2.7.3 is given herein, and should provide the main back-end components necessary to build a clinical application.

Calibration

A fully automatic closed-form calibration routine for spatially relating clinical components with the robot arm is developed. It is relatively fast, and has a system error in the sub-millimeter range as can be seen from the results presented previously. The calibration can be performed using either a simultaneous or a separable method, and is modifiable both in terms of measurements and calibration area. In addition to the robot arm, the routine utilizes an optical tracker commonly used in the operation room to spatially track ultrasound probes, surgical instruments and more.

Integrated robot manipulation

The calibration allows robot manipulation integrated with existing tools used for image-guided interventions. This allows precise motion of the robot arm based on input from external pointing instruments, but also with respect to patient registered preoperative data. Implemented functionality includes a click-able navigation window with possible motion planning, such as defined scans or simple motion to navigation points. In addition, a visual servoing functionality, or vision-based robot control, using ultrasound is implemented as an extension point. Together with advanced image analysis, the latter can be used to autonomously navigate a probe held by a robot arm.

Graphical user interface and visualization

An user interface is built on top of the framework to allow user defined interaction with the robot arm directly from the software CustusX. Several types of motion modalities are included, such as manual, planned, ultrasound tracking, i.e visual servoing and more. In addition, the initialization settings, such as connection to the robot, calibration and visualization of one or all links are implemented and modifiable while running the application.

The visualization was greatly improved compared to the previous solution by adding a real-time updated dynamic model of the robot arm in the navigation window. This is shown in figure 4.14, where the robot arm is holding an ultrasound probe. A phantom is registered to the scene with preoperative data from CT and MRI, while a wireless pointer is held above the head by a human operator. The

different components are integrated and interactable.

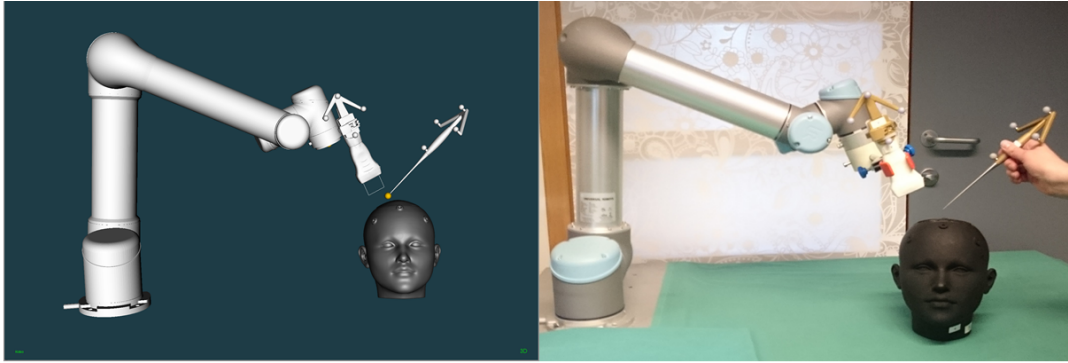


Figure 4.14: The left image shows the navigation window in CustusX with the robot manipulator holding an ultrasound probe. Additionally, a wireless pointing instrument is pointing towards a preregistered phantom head generated from data obtained using preoperative modalities such as CT and MRI. The right image shows the corresponding real-life laboratory scene.

The resulting work has been able to demonstrate the integrated robot system to the research team at SINTEF in a satisfactory manner. Demonstrations have been conducted and the functionality has been validated from the researchers as necessary for further development and projects in order to transfer the technology towards clinical applications. As a proof of concept videos used in presentations earlier this year showing the framework and robot in action are included following the link in appendix A.

Chapter 5

Discussion

To recap, the goal of this project was to derive relationships between the robot manipulator and objects relevant to an image-guided intervention. This involved development of automatic calibration routines feasible for clinical use. Desired functionality included robot manipulation as response to information obtained from external sources. This chapter is devoted to discussing the methodology, the results from verification experiments and the resulting framework. Suggestions for further development concludes the chapter.

5.1 Registration and calibration

Integrating a robot manipulator with existing and common tools used in image-guided interventions requires a shared reference frame relating every relevant object in space. Figure 3.1 and 3.2 shows two simplified maps sketched to illustrate the spatial relationships. The patient reference was chosen as the junction frame, often called world frame, opposed to what is used in CustusX, which introduces an additional parent reference [58]. In this scenario, introducing such a coordinate system would be redundant and are thus omitted. The patient reference is also the intuitive choice, as a hypothetical user, such as the clinician, would likely prefer planning the procedure with respect to the patient. Planning however, is performed in the user interface, and ideally the user should not need to be concerned about the world frame as it is handled in the software back end. The figures, which is greatly simplified in terms of the immense amount of coordinate frames that is actually present, suggests the crucial need of seamless mapping and transformations between frames. Moving a robot arm holding a medical tool with respect to the robot base instead of the patient reference can at worst be fatal. A systematic mapping of frames of interest is important in practice, but also helpful in the development process.

As shown in figure 3.1, the robot is spatially related to the rest of the system by two transforms, the patient reference to robot base \mathbf{T}_b^{pr} and the robots end-effector to the tool center point \mathbf{T}_t^{ee} of a medical instrument. Both the base of

the robot and patient reference is typically fixed during a procedure. In computer assisted surgery the latter is often used to correlate preoperative data of the patient spatially for interaction in the intraoperative scene. This restricts motion of the frame with respect to the patient during a procedure, without conducting a new patient registration. The robot base is usually fixed also, but contradictory scenarios are prevalent, such as moving the robot base with multi-axes Cartesian actuators to allow more flexibility. However, in this calibration routine the assumption is that the robot base is fixed with respect to the patient reference and similarly the robot end-effector to the medical tool. In practice this means that the patient needs to be fixed with respect to the patient reference, and consequently the robot base as well.

Among several possible approaches, a separable and simultaneous calibration method based on using the Kronecker product and vectorization was implemented. The reason for introducing two types of methods were to investigate their difference in error and effectiveness. Both methods require several measurements to be valid, and also access to the proposed work area for a limited amount of time. Further investigation should thus be conducted to improve clinical feasibility, as the full calibration routine would not be suitable with a patient present in the calibration volume. A practical solution would be mounting the robot base on a movable rack allowing more flexibility to the calibration process.

The calibration routines show several interesting results. The general tendency for the calibration error is that it decreases with increasing number of pose measurements conducted in the calibration volume. However, after a significant amount of measures the error saturates. This can be seen for both orientation and translation inspecting figure 4.1 and figure 4.2. Both methods approach approximately the same error, but the simultaneous case is significantly more fluctuating than the opponent. Apparently it is more fragile to noise than the separable method, which becomes quite clear from plots of the translational error components in figure 4.3. Compared, the simultaneous method is unpredictable, at least conducting less than 150 measurements. After this limit, the tendency is more congruent.

The two independent sensor systems used in the calibration, the optical tracker and the internal robot sensors, induces varying amount of error to the routine. Investigations conducted after the mimic moving target experiment showed that the robot sensor system had a negligible deviation compared to the optical in static positions. Experience with the optical tracker indicates several factors affecting the noise in the measurements. One discovery is that the markers on the tracked frames corrodes gradually causing a increasing spatial error over time. The tracking software provides error estimates, and investigation showed that using worn markers gave up to 0.5mm increased error in the calibration. Clearly, this is significant, and using new markers were thus endeavored. It is also unknown when the optically tracked frames were calibrated and how well the actual calibration was. The tracked frames are fragile to physical tension, and over time this may have caused

material deformation obviously affecting their initial calibrations. Since the acquisition method of the optical tracker is similar to a camera, the angle of incidence, as well as trembling will effect the routine. Placing the optical tracker for optimal depth resolution in the work area was sought. Another related observation is found inspecting the separable method in figure 4.3, which shows a slightly higher error in the z -direction. Gravity is believed to be one of the reasons for this discrepancy, as this is the direction of most gross force and probably most influenced from environmental factors, such as vibrations and more. A possible improvement of the issues related to noise mentioned above would be conducting several pose measurements in each position and averaging them before including them in the calibration.

As theory suggests, the separable routine cause a propagation of rotational error into the translational part. However, this is hard to identify from the results as the translational error do not appear to coincide with the rotation error. The saturation occurs significantly earlier for the translational case, which do not suggest a strong dependency. The motivation for introducing both a separable and a simultaneous method was to determine if the effect of propagating error in the prior method was considerable. However, the conclusion is that performance of the separable method is preferred, both due to its predictable behaviour, but also because of low error in position and orientation. As can be seen by table 4.1 the separable method almost exclusively outperform the simultaneous. Results from several independent calibration routines is presented in figure 4.4 and figure 4.5, makes it apparent that the superior choice in terms of both accuracy and reliability is the separable method. The rotation errors is hardly distinguishable, the the translational error is both lower and less fluctuation for the separable case.

The time consumption of the calibration routine is dependant on the number of pose measurements. Here, up to 620 measurements were conducted. This was an arbitrary number, and could be set differently dependant on situational requirements. In image-guided interventions, a positional error in the navigation system below 1 mm is by many accepted as a threshold good enough for most clinical procedures. Assuming this, only 174 measurements would be necessary for the simultaneous solutions, while 373 was needed for the separable. However, since the calibration is performed from one side of the work space center towards the other, less pose measurements would also reduce the volume of validity. A possible solution would be running the calibration routine in a grid of points where every measurement are conducted at the non measured point with longest distance from the volume centre while monitoring the error. The routine is complete when the accuracy is sufficient.

If one of the robot relating transforms is known, it enables quick recalibration of the other using the optical camera. However, this procedure introduces several practical issues. The passive markers on the physical tools or reference frames needs to be in the cameras view area, which effectively decreases the degrees of freedom in operational space. This is also a very relevant issue for general image-guided

interventions, as the clinicians have to keep the optically tracked tool visible to the camera in order to update its actual position. Additionally, quick recalibration would cause a direct propagation of errors from both the optical tracker and robot sensors into the new calibration matrix. A full calibration routine will minimize this error, and is thus preferred if applicable in the given scenario.

After calibration another obvious issue related to the static transformation requirement becomes relevant. Several of the potential applications involves contact between the tool extending the manipulator, for instance an ultrasound probe, and a surface of investigation. Consequently this results in an exerted force on the probe equal in magnitude and opposite in direction of the surface contact force. The probe may displace from the calibration with respect to the end-effector and thus induce an error in the system. In this project the probe adapter, as can be seen in figure 4.14 was poor, and regular use required frequent recalibration to avoid increasing error over time. Acquiring a new and more suitable probe adapter, for instance similar to the one used by Mathiassen *et al.* [29], is believed to reduce this problem.

Comparing the calibration results to related work is difficult as several choose to simulate the routine with Gaussian noise instead of conducting physical experiments. In addition, there is no common way of presenting the estimated errors. For a robot with six revolute joints, Dornaika and Horoud [44], presents two methods with relative translation error slightly higher than both methods presented here. Their closed-form method shows better results for the orientational error for a very small data set however. The Kronecker methods presented by Li *et al.* [46] and Shah[45] were sources of inspiration deriving the methods used in this project. According to the experimental results presented by Shah, the translational error is far from the threshold value of 1 mm, while the orientation error is quite fluctuating. The superior result herein might be correlated with better sensor system and less noise in experimental setup, also the number of measurements are significantly higher in our case.

The methods presented were so called closed-form solutions, and iterative calibration methods were not investigated during this project. Literature suggests that these methods are computationally demanding compared to closed-form solutions as they contain complex routines for optimization [42], but in return very accurate. A common issue with iterative methods is that they often not provide unique solutions, or require qualified guess on initial values. Compared to running a multiple measurement closed-form calibration routine, the latter is believed to be more tedious. A suggestion for future investigation, involves using the closed-form methods to initialize the calibration routine and after estimating a good guess for initial values, an iterative method could be used for further optimization. It is not known if this will have any effect on accuracy nor time consumption.

5.2 Image processing and analysis

One of the most important aspects of visual servoing using ultrasound is the image processing and analysis. In this thesis the latter is more a proof of concept for integration with robot manipulation than a novel contribution to the field. A pipeline with the goal of semi-autonomous detection and tracking of structures with circular properties was presented in section 3.4. The different steps included depends on the application. As example the cropping and Kalman filtering is used for tracking detected structures, but is omitted conducting a search for structures.

Cropping is introduced for several reasons. First, the processing speed could be significantly improved by analysing a sub-region of the image instead of the entirety. By estimating where objects of interest are most likely to reside, it is possible reduce the analysing region to avoid dedicating time to irrelevant areas of the image. Table 4.2 shows the effect on time consumption running the image analysis on images with different size. The largest images correspond to full sectors of typical linear B-mode probes. Apparently analysing images with this size causes real-time issues even if the frame rate is as low as 10. The smallest images correspond to typical sizes of cropped images tracking structures with diameter close to 10mm. Here we can cope with a frame rate up to approximately 25. This is acceptable, but would definitely benefit improving the processing speed.

The image analysis pipeline consists of several preprocessing steps, such as thresholding, smoothing and edge detection. To detect the most probable circle a Hough transform was applied to the edges of the original image. The image analysis in this project was kept at a trivial level, since the main objective was to implement an extension point for visual servoing, not to make novel contributions to the topic of ultrasound image analysis. However, the author believes that this is the main challenge in further development of visual servoing using ultrasound. Robust techniques are vital, and throughout the work several unsought situations occurred as a result of divergence in the image analysis. An example was related to the cropping, where the external library ITK was used to define the region of image extraction. Occasionally, the returned image was flipped, causing the image analysis pipeline to return faulty centre locations. At worst this could cause undesired motion of the probe, possibly losing the tracked structure in the image sector. Another concern is the speed of analysis, which already is on the verge of not being sufficient. This is believed to be a bigger problem in a clinical setting, with more complexity and non-trivial medical images. A possible change to increase the speed of analysis would be utilizing the graphical processing unit (GPU) instead of the central processing unit (CPU). The GPU can solve parallel problems faster than the CPU, while being more energy efficient and affordable [52].

At the end of the image pipeline, a Kalman filter is applied to produce an estimate of the new circle center location, based on both the current single measurements and the previous states with model consideration. A standard motion model with

constant acceleration was chosen, but it probably would be beneficial to inspect and identify properties of the tracked object for practical implementations. For a lab setup this model was sufficiently complex. Several parameters are used in the Kalman filter, and some are very relevant for the results. The covariance of the process noise \mathbf{Q} influences the weighting of measurements in the state update. A higher value makes the state update more affected by measurements and vice versa. Also the noise covariance \mathbf{R} needs to be initialized. Both parameters are tuned for the specific application, and depending on the quality of the tracking algorithm up to filtering. Applying the Kalman filter has definitely improved tracking, and introduced a more robust entirety. A bad measure of the circle center from the Hough transform, is effectively corrected with the Kalman filter.

Qualitatively the result from image analysis is satisfying. Figure 4.8 shows an ultrasound image with a proposed circle completely enclosing the reflecting object. The size and location is used to create a sphere in the image plane of the probe in the 3D navigation window. This gives a very good intuition about the location of the object in space with relation to the robot. The location ground truth is not normally known for practical examples like this, but an attempt was made by setting up an experiment according to figure 3.12. A calibration arm with a single sphere marker at the foot was placed in a water tank. The mounted plate with markers on the arm was tracked optically, with an manually estimated calibration from the plates origin to the single marker in the water. The location of the single marker was used as ground truth in the image analysis. A plot of the resulting deviation between location proposed by image analysis and the optically tracked calibration arm is given in figure 4.6. On average the deviation lies slightly below 1 mm which is fair. Despite this, the calibration arm was questioned as ground truth due to the manual calibration between tracked frame and single sphere. Apparently this is a very fragile setup, and since the deviations are quite consistent, this might be due to a systematic bias from the initial calibration.

Estimation of structure size is useful information in many contexts, for instance for diagnostics where abnormalities in the tissue is quantified. Visualization of properly scaled structures in the navigation window requires a proper estimation of the size. Here, it can also be used as a control parameter in visual servoing. To estimate the most probable diameter of a circular structure, the Hough transformation is run with a range of possible diameters while monitoring the the number of intersections. The most probable circle location and diameter correspond to the maximum number of intersections. A result from an experiment validating this calculation is given in figure 4.7, where the diameter of a circle measured in several positions is presented. Noticeably the trend is slightly higher than the true value. This is expected as the borders of the structure is smeared out in the ultrasound image due to reflections. With more information on this effect, there would be possible to compensate for this by scaling the proposed size with a determined factor. However, the result is quite

good as the deviation always is smaller than 0.5mm, and further improvement was not found necessary in this project.

A clinical environment is very complex, and would eventually require extended robot vision, both for internal navigation using ultrasound as is the case here, but also external overview of the entire scene. It is necessary to make the robot more feasible and collaborative in interaction with humans. Establishing a two way communication between the ultrasound scanner and the workstation would result in more flexibility. As of now, communication is routed from the scanner to the workstation, but arguably it would be beneficial not to modify scanner settings such as image depth and frequency outside the framework. Image analysis and its applications, could potentially take a leap towards autonomy if this functionality was implemented, and feasibility would definitely increase as the user only need to interact with one workstation.

5.3 Integration of robot manipulation

Relationships between coordinate frames enables the integrated robot manipulation, which is a generic concept involving sequential motion, as well as single moves. The methods of motion is partially developed earlier, but integration with tools and patient scenes was enabled by calibration. Consequently this laid the foundation for new functionality related to image guided interventions, but potentially powered by the robot manipulator. Exploited methods involve preplanned motion such as scans physically defined using a surgical pointer or from the user interface interacting with patient registered preoperative data. Additionally, creating a extension point for visual servoing using ultrasound was possible, since the imaging sector of the ultrasound probe now was spatially related to the robot base through known transformations.

The motivation of implementing functionality for visual servoing using ultrasound is to enable autonomous tracking of moving structures in the body. Several clinical procedures struggle with issues related to this problem, and it is both hard and tedious to perform as a human. In this project, visual servoing was used for centering a imaging ultrasound probe over a tracked structure, and keeping the structure approximately fixed in the center of the image even while moving. A common restriction tracking structures this way was the ultrasound frame rate, as it limits the maximum velocity of the structure in the image plane. Here we assumed that the motion of trackable structures is less than the approximated maximum velocity of the diaphragm, i.e 100mm s^{-1} . Standard ultrasound scanners satisfies this criteria, but obviously the workload of image processing and analysis increases with frame rate. For tracking purposes, estimating the velocity of the structure between frames enables the possibility of servoing the probe as compensation, effectively increasing the maximum velocity of the tracked structure. Anyhow, motion of the circular

structure perpendicular to the image plane was found to be the biggest problem while tracking. Here, the the values of radius and number of circle intersections in the Hough transform was used as control parameters. Sadly, the experimental setup did not include the necessary materials to quantitatively investigate the tracking of moving targets. Centering the probe over a stationary structure after discrete offsets is the closest scenario, which is also investigated during a verification experiment. Figure 4.9 shows the resulting distance between the probes axial axis and the sphere center after running the probe centering routine. The optical tracker and internal robot sensors shows similar results, which indicates that the calibration was good and that introducing the robot manipulator did not induce larger errors to the existing system. On average, the distance from axial axis of the centered probe and to the sphere is lower using the image analysis. The most probable reason for this is again the manual calibration of the arm used in the experiment, together with the fact that motion is based on image analysis and not the optically tracked position of the sphere. All in all, the results were adequate.

As can be understood, visual servoing requires a very robust image analysis with interpretation of structure location in space. The main problem was related to perpendicular motion of the structure, however, as explained above this was not investigated quantitatively due to experimental restrictions. Improving the method implemented in this project is undoubtedly needed for practical use. Here, image processing was restricted to digital intensity based methods, and data components from signal processing such as RF or envelope data, was not available. Speckle tissue tracking, which is proposed as a possible solution to the problem with out-of-plane motion [14] is not implementable as it requires one of the mentioned data types. Hopefully these data types can be included and utilized in later versions of CustusX. Solving the problem could also be done using special probes [18] if available. However, if the standard 2D B-Mode probe and intensity based methods are to be used, the author believes that the method based on Shearlet transformations [16] should be investigated. Either way, the image analysis needs to be very robust for visual servoing in a clinical context.

Related to autonomous visual servoing is also the robots ability to mimic motion of a wireless pointer. A tracked surgical pointer can be used to plan the motion of the robot arm, for example defining a path for ultrasound scanning. Figure 4.10 shows results from an experiment where an operator moves a surgical tool in the operational space while optically tracking its position. The recorded positions are then used as a path of motion for the robot manipulator holding a ultrasound probe. Qualitatively, by inspection the paths are coinciding both tracking the probe with a optical tracker and utilizing the robot sensor system. In figure 4.11 the minimum distance from every point on the sampled path to the moved probe is given. The results using the internal robot sensors are significantly better than the optical tracker. The main reason is that the robot is moving the tool with respect to its

known position, which is tracked using the robot sensor system. Another interesting behaviour is prevalent in optical measurements. Clearly they are varying through the volume of motion. The error seem correlated with the distance from the camera, and possibly areas of lesser visibility. Noticeably, they are worryingly high at the end of the path. These results suggest the possibility of substituting the optical tracking system with the internal robot sensors when applicable. The internal robot sensor is superior in this context, as it is not dependant of visibility.

As a curiosity for further development, a preplanned ultrasound scan over a phantom was performed in the end. A 3D reconstruction of the acquisition was conducted with results as shown in both figure 4.12 and figure 4.13. In addition, the image analysing pipeline was running during the procedure to search for possible circular structures in the investigated volume. Seven abnormalities are found inspecting the reconstruction. Both the total image intensities and normalized number of intersection of the Hough transform along the scan lines suggested seven to eight regions with higher intensity than normal. From the CT data, seven tumor models was visible, making the observations from both modalities coincide.

3D reconstructions are performed utilizing the optically tracked ultrasound probe location at acquisition. This can make the procedure problematic as it requires visual line of sight between probe and tracker during the routine. Areas where the probe is poorly visible is fragile, inducing possible errors in the reconstruction. As the previous results suggested, using the internal robot sensors could provide a decent substitute of the optical transform between patient reference and tool \mathbf{T}_t^{pr} . This is also used for the 3D reconstruction, so utilizing the robot sensors instead of the optical during acquisition would remove the issue with visibility, and possibly improve the reconstruction quality.

5.4 Robot framework

Seamless integration into an existing platform for image-guided interventions, such as CustusX, with several extensibility points is believed necessary to establish an accepted standard in the research community. As mentioned, an extension named LightWeightRobotIGT is under development in 3D slicer by Tauscher *et al.* [22] using OpenIGTLink as a network communication protocol. Compared the robot framework developed in this project has several similarities, such as visualization, state updates and simple control. However, LightWeightRobotIGT lacks functionality beyond this point. Except for a visual example, there is no information about the methods for relating the robot manipulator to tools used in image-guided interventions, nor any proven integration with ultrasound. Conclusively, the framework implemented in this project seem more mature than LightWeightRobotIGT from a functionality perspective. To the best of the authors knowledge, the implemented framework is one of the most generic platforms for robot manipulation in image-

guided interventions. The functions are not considered novel, but an integration of this order is an original contribution to the field. As of now, there is no existing platform providing all of the presented functionality and extension points. The integration with CustusX is of course essential, as the platform provides several methods directly applicable for use with robot arms. CustusX is apparently well suited for integrated robot control, and will hopefully serve as a game changer in the future.

Developing a software for collaborative robot control in image-guided interventions is undoubtedly beneficial in the coming years. The amount of industrial collaborative robots used in health care is believed to increase immensely. Their flexibility and low weight, together with being programmable allows them to do specific and user-independent tasks with possible implications on the clinical work-flow. The manipulator used in this project, the UR5, would have a small footprint in a clinical environment, and has through verification experiments proven to operate in a precise manner. As mentioned earlier, a standard (IEC/NP 80601-2-77) for medical robots in surgery is under development. After standard commissioning, a large scale production of industrial manipulators satisfying the inscribed criteria is likely initiated by the large commercial firms. It is unknown whether the UR5 manipulator will need improvements to fulfill the standard but the developed concepts could easily be refined for use with other manipulators.

5.5 Future work

The previous sections revealed the frameworks features, and also included some suggestions for specific improvements. The following is restricted to present the work feasible extending the framework within a reasonable perspective. It is divided into two parts, technical extensions and clinical applications. Several suggestions was given to improve the framework in the previous discussion, so the technical focus in the following will mainly involve materialistic extensions. Based on the frameworks current features and performance, the author presents some future possible directions for clinical applications.

5.5.1 Technical extensions

An issue with this setup, and other image-guided interventions, is using optical trackers with passive markers. It is very fragile to patient movement if registered to preoperative data, and other inconsistencies in the environment such as movement of the robot base. A potential solution would eventually be to introduce 3D cameras or similar technologies, as is the case in the article by Shademan *et al.* [30]. A 3D camera can contribute to better adaption to the surroundings, not being restricted to only track passive markers. A promising 3D camera that is worth looking into

is developed in the research environment in Trondheim, namely the Shapecrafter 3D¹, which by the developers are claimed to be a prototype of the most accurate real-time 3D camera in the world. The camera outputs a 3D point cloud which is supported in CustusX. The depth resolution is proportional with the image size, and for a 100cm width image it supposedly has sub millimeter accuracy. The camera is made commercially available later this year. The author has not made a reviewed search on 3D camera technology, but believes the specifications described can provide more possibilities extending the framework. All types of cameras has a limited view area, but compared to the optical tracking system used in this project, the 3D camera does not require passive markers. However, it requires some sort of patient registration in order to relate the current scene to preoperative data. This requires further investigation, and sophisticated development of registration algorithms. In the context of robot manipulation it would arguably be much better to use a 3D camera as these can provide surface information in real-time, making it easier to manipulate robots with respect to the whole environment.

Today, most of the renowned robotic surgical systems lack the ability of force feedback, including multiple editions of the mentioned da Vinci system [9]. Introducing this through pressure sensors or similar technology would give the robot system better tactility [69] and in combination with vision this is believed to have several application areas, for instance automatic ultrasound scans. Ultrasound scans are often performed by free hand, and without going into deep detail, involuntary motion can potentially induce error [70]. The acquisition is affected by acoustic contact and variable velocity in the motion. Using a real-time force feedback system together with a robot arm could in theory be used to retain best possible acoustic contact on surfaces of interest and steady velocity to decrease that undesirable variations and thus increase the image quality. As mentioned earlier, Mathiassen *et al.* has implemented a force feedback system and transferred the tactility into a spatial navigator [29] in which the user interacts. This gives the user the ability to move the manipulator, including a sense of touch. Implementing a system like this would undoubtedly be helpful in a clinical scenario, as it would potentially allow easier transitions between user interaction and autonomous control. Future development of the feedback system towards higher degree of autonomy would introduce extended functionality and control for visual servoing using ultrasound.

Extending the framework for clinical applications is desirable, but also demanding. If the goal is to go beyond the research step, the medical software need to satisfy several elaborated standards (ISO 13485, ISO 14971, IEC 62304). The goal of these standards is to ensure that the software is designed for use in a good and safe manner. The IEC 62304 [71] specifies the life cycle requirements of the medical software, and if possible in the future software development this should be used as a template. As of now, CustusX is not approved for clinical use.

¹<http://www.shapecrafter.no/>

5.5.2 Clinical application

A wide range of clinical applications can make good use of robotic arms, especially the simple, repetitive and tedious procedures. Expansion possibilities include automatic ultrasound scanning, robot-guided needle insertions and biopsy, as well as a selection of therapeutic ultrasound and radio-surgical procedures.

A featured procedure that possibly would benefit from a system with implemented methods for visual servoing based on ultrasound is treatment of kidney stones. Kidney stones, or urinary calculi, is hard structures of crystals, which may be formed in the kidney or ureter, a condition known as urolithiasis [72]. The incidence is increasing among the general population, partly because of obesity [73, 74]. Currently Extracorporeal Shock Wave Lithotripsy (ESWL) is standard of care depending on stone size and density. This is a non-invasive procedure performed by focusing acoustic shock waves towards kidney stones localised using either X-ray or ultrasound. The goal is to fraction the stones such that they pass spontaneously through the ureter [75, 76]. Despite still being the favorable procedure by being non-invasive, it has several drawbacks. The main issue is that kidney stones move during the procedure, both due to physiological effects, but also because of the exerted pressure from shock waves. The shock waves are emitted with a constant frequency of approximately 0.5 Hz - 2 Hz for a period of several minutes, before the operator again uses X-ray to check the status of both stone fractionation and localization. Not updating this information frequently enough causes more than 50% of the wave energy to miss the target, causing damage to surrounding tissue and longer treatment times. By introducing real-time visual servoing using ultrasound the ESWL method could be improved significantly by synchronising precise stone localisation with the emitted shock waves. The potential result is an increased number of treated patients, shorter treatment- and convalesce times, as well as less use of X-ray and anaesthesia.

Anyhow, predicting the future is difficult. In light of the global upheaval in robot-based interventions and use of robotics for treatment of patients, research in this field is necessary to meet the necessary treatments of tomorrow. The demographic change will continue inducing more pressure on the clinical workload, but introducing sophisticated robot systems to certain procedures can possibly help attenuate the process. However, if this is to happen, technological competence must be combined with clinical expertise. Only together can these challenges be solved.

Chapter 6

Conclusion

In this project a robot framework integrated with the navigation system CustusX is developed. The framework is written in C++ utilizing several external libraries. Implemented functionality includes automatic calibration routines for spatially relating the robot arm and tools used for image-guided interventions. From verification experiments the positional calibration error is determined to be in the sub-millimeter range, making it viable for most clinical applications. In addition, the calibration creates spatial relationships enabling possibilities for integrated robot manipulation. This includes user planned robot motion of medical instruments, such as ultrasound probes. Motion planning can be done in the user interface interacting with patient registered preoperative data, or in the physical scene with a wireless pointer. Additional functionality for visual servoing using ultrasound is developed, which enables autonomous robot motion interpreting image analysis.

Despite lacking functional novelty, the framework has several promising features and extension points. Together with CustusX the system provides original integration of robot control with existing implementations renowned in the research community of image-guided intervention. Overall, the framework looks promising. All predetermined development goals are considered fulfilled with satisfactory results, and several directions for further work of both technical and clinical relevance are possible and suggested. Technically this involves implementation of 3D camera and haptic technology, to improve the visual servoing. Another goal should be targeted development towards ISO approvals. Using the framework and the manipulator as a basis, several clinical applications are believed implementable, especially standardized and repetitive tasks. Featured expansion possibilities include automatic ultrasound scanning, robot-guided needle insertions and biopsy, as well as a selection of therapeutic ultrasound and radio-surgical procedures. These expansions involve large patient groups, and with the forecast on demographic change further development should be emphasized. The performance of the framework and manipulator, including the economical aspects, is believed to be clinically feasible.

Bibliography

- [1] Wolfgang Lutz, Warren Sanderson, and Sergei Scherbov. The coming acceleration of global population ageing. *Nature*, 451(7179):716–719, February 2008.
- [2] Kristian Roksvaag. *Arbeidsmarkedet for helse- og sosialpersonell fram mot år 2035 : dokumentasjon av beregninger med HELSEMOD 2012*, volume 2012/14 of *Rapporter (Statistisk sentralbyrå : online)*. Statistisk sentralbyrå, 2012.
- [3] Ryan A. Beasley. Medical Robots: Current Systems and Research Directions. *Journal of Robotics*, 2012:14, 2012.
- [4] Matthew Kroh and Sricharan Chalikonda. *Essentials of Robotic Surgery*. Springer International Publishing: Cham, 2015.
- [5] Arthur Visser. Rise of the Industrial Robots. *Connector and Cable Assembly Supplier*, January 2014.
- [6] Andreas Østvik. *Robotic Control in image-guided intervention*. Department of Physics, NTNU, 2015.
- [7] H. M. Shao, J. Y. Chen, T. K. Truong, I. S. Reed, and Y. S. Kwoh. A New CT-Aided Robotic Stereotaxis System. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 668–672, November 1985.
- [8] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati. A robot with improved absolute positioning accuracy for CT guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35(2):153–160, February 1988.
- [9] Intuitive Surgical. The da vinci surgical system. *Intuitive Surgical Inc., Sunnyvale, CA, available at: <http://www.intuitivesurgical.com>*, 2013.
- [10] A. M. Priester, S. Natarajan, and M. O. Culjat. Robotic ultrasound systems in medicine. *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 60(3):507–523, March 2013.
- [11] A. Krupa and F. Chaumette. Control of an ultrasound probe by adaptive visual servoing. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2681–2686, Aug 2005.
- [12] A. Krupa and F. Chaumette. Guidance of an ultrasound probe by visual servoing. *Advanced Robotics*, 20(11):1203–1218, November 2006.
- [13] Alexandre Krupa, David Folio, Cyril Novales, Pierre Vieyres, and Tao Li. Robotized Tele-Echography: an Assisting Visibility Tool to Support Expert Diagnostic. *IEEE Systems Journal*, pages 1–10, April 2014. in press.

- [14] A. Krupa, G. Fichtinger, and G.D. Hager. Real-time motion stabilization with b-mode ultrasound using image speckle information and visual servoing. *The International Journal of Robotics Research, IJRR*, 28(10):1334–1354, 2009.
- [15] Thomas L. Szabo. *Diagnostic Ultrasound Imaging: Inside Out, second edition*. Elsevier, 2014.
- [16] Lesley-Ann Duflot, Alexandre Krupa, Brahim Tamadazte, and Nicolas Andreff. Toward Ultrasound-based Visual Servoing using Shearlet Coefficients. In *IEEE Int. Conf. on Robotics and Automation, ICRA'16*, Stockholm, Sweden, May 2016.
- [17] C. Nadeau and A. Krupa. Intensity-based ultrasound visual servoing: modeling and validation with 2d and 3d probes. *IEEE. Trans. on Robotics*, 29(4):1003–1015, August 2013.
- [18] C. Nadeau and A. Krupa. Improving ultrasound intensity-based visual servoing: tracking and positioning tasks with 2d and bi-plane probes. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'11*, pages 2837–2842, San Francisco, USA, September 2011.
- [19] Andriy Fedorov, Reinhard Beichel, Jayashree Kalpathy-Cramer, Julien Finet, Jean-CristopheC. Fillion-Robin, Sonia Pujol, Christian Bauer, Dominique Jennings, FionaM Fennesy, Milan Sonka, John Buatti, StephenR Aylward, JamesV Miller, Steve Pieper, and Ron Kikinis. 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, 30(9):1323–41, 11 2012.
- [20] Tian Xia, Clint Baird, George Jallo, Kathryn Hayes, Nobuyuki Nakajima, Nobuhiko Hata, and Peter Kazanzides. An integrated system for planning, navigation and robotic assistance for skull base surgery. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 4(4):321–330, 2008.
- [21] Junichi Tokuda, GregoryS Fischer, Xenophon Papademetris, Ziv Yaniv, Luis Ibanez, Patrick Cheng, Haiying Liu, Jack Blevins, Jumpei Arata, AlexandraJ Golby, Tina Kapur, Steve Pieper, EveretteC Burdette, Gabor Fichtinger, ClareM Tempny, and Nobuhiko Hata. Openigtlink: An open network protocol for image-guided therapy environment. *Int J Med Robot*, 5(4):423–434, 12 2009.
- [22] Sebastian Tauscher, Junichi Tokuda, Günter Schreiber, Thomas Neff, Nobuhiko Hata, and Tobias Ortmaier. OpenIGTLink interface for state control and visualisation of a robot for image-guided therapy systems. *International Journal of Computer Assisted Radiology and Surgery*, 10(3):285–292, 2015.
- [23] ISO 10218-1:2011 - Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots.
- [24] ISO 13849-1:2015 - Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design.
- [25] IEC/NP 80601-2-77 - Medical electrical equipment – Part 2-77: Particular requirements for the basic safety and essential performance of medical robots for surgery.
- [26] Fereshteh Aalamifar, Rishabh Khurana, Alexis Cheng, Russell H. Taylor, Iulian Iordachita, and Emad M. Boctor. Enabling technologies for robot assisted ultrasound tomography: system setup and calibration. *Proc. SPIE*, 9040:90401X–90401X–9, 2014.

- [27] Fereshteh Aalamifar, Rishabh Khurana, Alexis Cheng, Xiaoyu Guo, Iulian Iordachita, and Emad M. Boctor. Enabling technologies for robot assisted ultrasound tomography. *The International Journal of Medical Robotics and Computer Assisted Surgery*, pages n/a–n/a, January 2016.
- [28] Lars Eirik Bø, Erlend Fagertun Hofstad, Frank Lindseth, and Toril A N Hernes. Versatile robotic probe calibration for position tracking in ultrasound imaging. *Physics in Medicine and Biology*, 60(9):3499, 2015.
- [29] Kim Mathiassen, Jørgen Enger Fjellin, Kyrre Glette, Per Kristian Hol, and Ole Jakob Elle. An Ultrasound Robotic System Using the Commercial Robot UR5. *Frontiers in Robotics and AI*, 3(1), 2016.
- [30] Azad Shademan, Ryan S. Decker, Justin D. Opfermann, Simon Leonard, Axel Krieger, and Peter C. W. Kim. Supervised autonomous robotic soft tissue surgery. *Science Translational Medicine*, 8(337):337ra64–337ra64, 2016.
- [31] Peter Corke. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, volume 73 of *Springer Tracts in Advanced Robotics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
- [32] Leonhard Euler. Formulae generales pro translatione quacunque corporum rigidorum. *Novi Acad. Sci. Petrop*, 20:189–207, 1775.
- [33] Richard M. Murray. *A mathematical introduction to robotic manipulation*. CRC Press, 1994.
- [34] G Zehfuss. Über eine gewisse determinante. *Zeitschrift für Mathematik und Physik*, 3:298–301, 1858.
- [35] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5):403–420, 1970.
- [36] Kirk Baker. Singular value decomposition tutorial. *The Ohio State University*, 24, 2005.
- [37] Bruno Siciliano. *Robotics : Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing. Springer London, 2009.
- [38] Mark W. Spong. *Robot modeling and control*. Wiley, 2006.
- [39] Jacques Denavit and Richard Scheunemann Hartenberg. A kinematic notation for lower-pair mechanisms based on matrices. *Trans ASME J. Appl. Mech*, 23:215–221, 1955.
- [40] Universal Robots A/S. UR5 Technical specifications. Item no. 110105. http://www.universal-robots.com/media/50588/ur5_en.pdf. Last accessed: 2016-06-06.
- [41] Roger Y. Tsai and Reimar K. Lenz. Real time versatile robotics hand/eye calibration using 3D machine vision. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation (Cat. No.88CH2555-1)*, pages 554–561, 1988.
- [42] Mili Shah, Roger D. Eastman, and Tsai Hong. An Overview of Robot-sensor Calibration Methods for Evaluation of Perception Systems. In *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, PerMIS '12, pages 15–20, New York, NY, USA, 2012. ACM.

- [43] Z. S. Hanqi Zhuang, R. Roth, and R. Sudhakar. Simultaneous Calibration Of Robot/world And Eye/hand Transformations. In *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on*, volume 2, pages 1063–1070, 1992.
- [44] F. Dornaika and R. Horaud. Simultaneous robot-world and hand-eye calibration. *Robotics and Automation, IEEE Transactions on*, 14(4):617–622, 1998.
- [45] Mili Shah. Solving the Robot-World/Hand-Eye Calibration Problem Using the Kronecker Product. *Journal of Mechanisms and Robotics*, 5(3):031007, 2013.
- [46] Aiguo Li, Lin Wang, and Defeng Wu. Simultaneous robot-world and hand-eye calibration using dual-quaternions and kronecker product. *International Journal of the Physical Sciences*, 5(10):1530–1536, 2010.
- [47] Gene H Golub and Charles F Van Loan. Matrix computations. johns hopkins studies in the mathematical sciences, 1996.
- [48] Bjørn A. J. Angelsen. *Ultrasound imaging : waves, signals, and signal processing : Vol. 1 : Basic principles, wave generation, propagation, and beamforming in homogeneous tissue*. Emantec AS, 2000.
- [49] M. A. Flower and Steve Webb. *Webb's physics of medical imaging*. Series in medical physics and biomedical engineering. CRC Press, 2nd ed. edition, 2012.
- [50] Gabor Herman. *Fundamentals of Computerized Tomography: Image Reconstruction from Projections*. Advances in Pattern Recognition. Springer London, London, 2009.
- [51] T. Acharya and A.K. Ray. *Image Processing: Principles and Applications*. Wiley, 2005.
- [52] Erik Smistad, Thomas L. Falch, Mohammadmehdi Bozorgi, Anne C. Elster, and Frank Lindseth. Medical image segmentation on GPUs – A comprehensive review. *Medical Image Analysis*, 20(1):1–18, February 2015.
- [53] Rafael C. Gonzalez. *Digital image processing*. Pearson/Prentice Hall, 3rd ed. edition, 2008.
- [54] Mehmet Sezgin and Bulent Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–168, 2004.
- [55] Linda G. Shapiro. *Computer vision*. Prentice Hall, 2001.
- [56] Mark A. Haidekker. The Hough Transform, 2010.
- [57] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960.
- [58] Christian Askeland, Ole Vegard Solberg, Janne Beate Lervik Bakeng, Ingerid Reinertsen, Geir Arne Tangen, Erlend Fagertun Hofstad, Daniel Høyer Iversen, Cecilie Våpenstad, Tor-mod Selbekk, Thomas Langø, Toril A. Nagelhus Hernes, Håkon Olav Leira, Geirmund Unsgård, and Frank Lindseth. CustusX: an open-source research platform for image-guided therapy. *International Journal of Computer Assisted Radiology and Surgery*, pages 1–15, September 2015.

- [59] Marco Nolden, Sascha Zelzer, Alexander Seitel, Diana Wald, Michael Müller, AlfredM. Franz, Daniel Maleike, Markus Fangerau, Matthias Baumhauer, Lena Maier-Hein, KlausH. Maier-Hein, Hans-Peter Meinzer, and Ivo Wolf. The medical imaging interaction toolkit: challenges and advances. *International Journal of Computer Assisted Radiology and Surgery*, 8(4):607–620, 2013.
- [60] OSGI Alliance. OSGi Service Platform Core Specification, Release 4, Version 4.2. (2009). <http://www.osgi.org/Download/Release4V42>. Last accessed: 2015-11-27.
- [61] Jasmin Blanchette. *C++ GUI Programming with Qt 4, Second Edition*. Prentice Hall, 2008.
- [62] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. <http://eigen.tuxfamily.org>, 2010.
- [63] Inc Kitware. *The VTK user’s guide*. Kitware, 11th ed. edition, 2010.
- [64] Hans J. Johnson, M. McCormick, L. Ibáñez, and The Insight Software Consortium. *The ITK Software Guide*. Kitware, Inc., third edition, 2013. *In press*.
- [65] SRI International. Artificial Intelligence Center and G.J. Agin. *Real time control of a robot with a mobile camera*. Technical note. SRI International, 1979.
- [66] J. Tavares and R.N. Jorge. *Computational Vision and Medical Image Processing: Recent Trends*. Computational Methods in Applied Sciences. Springer Netherlands, 2010.
- [67] J. Ehrhardt and C. Lorenz. *4D Modeling and Estimation of Respiratory Motion for Radiation Therapy*. Biological and Medical Physics, Biomedical Engineering. Springer Berlin Heidelberg, 2013.
- [68] Ole Vegard Solberg, Frank Lindseth, Lars Eirik Bø, Sebastien Muller, Janne Beate Lervik Bakeng, Geir Arne Tangen, and Toril A. Nagelhus Hernes. 3d ultrasound reconstruction algorithms from analog and digital data. *Ultrasonics*, 51(4):405 – 419, 2011.
- [69] C.R. Wagner and R.D. Howe. Force Feedback Benefit Depends on Experience in Multiple Degree of Freedom Robotic Surgery Task. *IEEE Transactions on Robotics*, 23(6):1235–1240, December 2007.
- [70] Pai-Chi Li, Cheng-Yen Li, and Wen-Chun Yeh. Tissue motion and elevational speckle decorrelation in freehand 3d ultrasound. *Ultrasonic imaging*, 24(1):1–12, 2002.
- [71] IEC 62304:2006 Medical device software – Software life cycle processes.
- [72] Orson W Moe. Kidney stones: pathophysiology and medical management. *The Lancet*, 367(9507):333–344, February 2006.
- [73] Nicole L. Miller. Modern bariatric surgery and nephrolithiasis—are we on the verge of a new epidemic? *The Journal of Urology*, 179(2):403 – 404, 2008.
- [74] Brian R. Matlaga, Andrew D. Shore, Thomas Magnuson, Jeanne M. Clark, Roger Johns, and Martin A. Makary. Effect of gastric bypass surgery on kidney stone disease. *The Journal of Urology*, 181(6):2573 – 2577, 2009.
- [75] CH. Chaussy, Walter Brendel, and E. Schmiedt. Originally published as volume 2, issue 8207 extracorporeally induced destruction of kidney stones by shock waves. *The Lancet*, 316(8207):1265 – 1268, 1980.

- [76] C. Chaussy, E. Schmiedt, D. Jocham, W. Brendel, B. Forssmann, and V. Walther. First clinical experience with extracorporeally induced destruction of kidney stones by shock waves. *The Journal of Urology*, 127(3):417–420, March 1982.

Appendix A

Digital appendix

Following the links below gives access to the code implemented during the project. The branch contains all the necessary code for running the experiments presented and the functionality of the framework. It will not necessarily be up to date with the current development branch in the software CustusX, as it is dynamic. However, it is possible to download and build the software without being synchronised with CustusX.

Development branch - robot framework code:

<https://github.com/androst/CustusX/tree/feature/robot/source/plugins/org.custusx.robot.ur5>

A video used in a presentation earlier this year gives a proof of concept. In addition, raw data from measurements can be found following the link below:

<https://goo.gl/SwN1FP>

<https://drive.google.com/open?id=0B4j2FpydMNfQOFdEeGxzbHZTVW8>

Please do not hesitate to ask if there is any problems or questions. I can be reached on my student mail address androst@stud.ntnu.no.

Appendix B

Definitions

B.1 The skew-symmetric matrix

Skew-symmetric matrices is square matrices satisfying $\mathbf{S}^T + \mathbf{S} = 0$, i.e its transpose is equal to its negative matrix. Considering the vector $\mathbf{k} = [k_x \ k_y \ k_z]^T$ it holds for $\mathbf{S} \in \mathbb{R}^{3 \times 3}$ that

$$\mathbf{S}(\mathbf{k}) = \begin{bmatrix} 0 & -k_z & k_y \\ k_z & 0 & -k_x \\ -k_y & k_x & 1 \end{bmatrix}.$$

B.2 Basic transformation matrices

A common trigometric abbreviation for the trigometric functions sine and cosine can be written as

$$\cos \beta = c_\beta \quad \text{and} \quad \sin \beta = s_\beta.$$

The set of the basic homogeneous rotation transformations are given by

$$\mathbf{R}_x(\beta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\beta & -s_\beta & 0 \\ 0 & s_\beta & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_y(\beta) = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{R}_z(\beta) = \begin{bmatrix} c_\beta & -s_\beta & 0 & 0 \\ s_\beta & c_\beta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The set of basic homogeneous translations are given as

$$\mathbf{T}_x(\beta) = \begin{bmatrix} 1 & 0 & 0 & \beta \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_y(\beta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & \beta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{T}_z(\beta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \beta \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Appendix C

UR5 Specifications

C.1 Robot state messages

A complete list of the robot state messages are as of september 2015 found in a excel file following the link to the manufacturers new support site <http://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/remote-control-via-tcpip-16496/>

C.2 Homogenous transformation matrix

The following is the derived homogeneous transformation matrix between the base and end-effector of the robot manipulator. Where d_i is the offset along the z_{i-1} -axis and a_i is the length of the common normal as defined by the Denavit-Hartenberg convention.

$$\mathbf{T}_e^b = \mathbf{H}_6^0 = \begin{bmatrix} \mathbf{R}_6^0 & \mathbf{p}_{06}^0 \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In addition to the length offsets a abbreviation for the trigonometric functions is used, i.e $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$ and $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$, where θ is the joint configuration.

$$r_{11} = c_6(s_1 s_5 + c_1 c_5 c_{234}) - c_1 s_6 s_{234}$$

$$r_{12} = -s_6(s_1 s_5 + c_1 c_5 c_{234}) - c_1 c_6 s_{234}$$

$$r_{13} = c_5 s_1 - c_1 c_{234} s_5$$

$$r_{21} = -c_6(c_1 s_5 - c_5 c_{234} s_1) - s_1 s_6 s_{234}$$

$$r_{22} = s_6(c_1s_5 - c_5c_{234}s_1) - c_6s_1s_{234}$$

$$r_{23} = -c_1c_5 - c_{234}s_1s_5$$

$$r_{31} = c_{234}s_6 + c_5c_6s_{234}$$

$$r_{32} = c_6c_{234} - c_5s_6s_{234}$$

$$r_{33} = -s_5s_{234}$$

$$p_x = d_4s_1 + d_6(c_5s_1 - c_1c_{234}s_5) + a_2c_1c_2 + c_1d_5s_{234} - a_3c_1s_2s_3 + a_3c_1c_2c_3$$

$$p_y = a_2c_2s_1 - d_6(c_1c_5 + c_{234}s_1s_5) - c_1d_4 + d_5s_1s_{234} - a_3s_1s_2s_3 + a_3c_2c_3s_1$$

$$p_z = d_1 + a_2s_2 + a_3s_{23} - d_5(c_4c_{23} - s_4s_{23}) - d_6s_5(c_4s_{23} + c_{23}s_4)$$

C.3 The Jacobian

The following is the derived Jacobian matrix of the UR5 manipulator. Where d_i is the offset along the z_{i-1} -axis and a_i is the length of the common normal as defined by the Denavit-Hartenberg convention.

$$\mathbf{J} = [\mathbf{J}_1 \quad \mathbf{J}_2 \quad \mathbf{J}_3 \quad \mathbf{J}_4 \quad \mathbf{J}_5 \quad \mathbf{J}_6] = \begin{bmatrix} j_{11} & j_{12} & j_{13} & j_{14} & j_{15} & 0 \\ j_{21} & j_{22} & j_{23} & j_{24} & j_{25} & 0 \\ 0 & j_{32} & j_{33} & j_{34} & j_{35} & 0 \\ 0 & j_{42} & j_{43} & j_{44} & j_{45} & j_{46} \\ 0 & j_{52} & j_{53} & j_{54} & j_{55} & j_{56} \\ 1 & 0 & 0 & 0 & j_{65} & j_{66} \end{bmatrix}$$

In addition to the length offsets a abbreviation for the trigonometric functions is used, i.e $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$ and $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$, where θ is the joint configuration.

$$j_{11} = d_6(c_1c_5 + c_{234}s_1s_5) + d_4c_1 - a_2c_2s_1 - d_5s_{234}s_1 - a_3c_2c_3s_1 + a_3s_1s_2s_3$$

$$j_{12} = -c_1(d_5(s_{23}s_4 - c_{23}c_4) + a_3s_{23} + a_2s_2 - d_6s_5(c_{23}s_4 + s_{23}c_4))$$

$$j_{13} = c_1(d_5c_{234} - a_3s_{23} + d_6s_{234}s_5)$$

$$\dot{j}_{14} = c_1(d_5 c_{234} + d_6 s_{234} s_5)$$

$$\dot{j}_{15} = d_6 c_1 c_2 c_5 s_3 s_4 - d_6 s_1 s_5 + d_6 c_1 c_3 c_5 s_2 s_4 + d_6 c_1 c_4 c_5 s_2 s_3 - d_6 c_1 c_2 c_3 c_4 c_5$$

$$\dot{j}_{21} = d_4 s_1 + d_6(c_5 s_1 - c_1 c_{234} s_5) + a_2 c_1 c_2 + c_1 d_5 s_{234} - a_3 c_1 s_2 s_3 + a_3 c_1 c_2 c_3$$

$$\dot{j}_{22} = -s_1(a_2 s_2 + a_3 s_{23} - d_5(c_4 c_{23} - s_4 s_{23}) - d_6 s_5(c_4 s_{23} + c_{23} s_4))$$

$$\dot{j}_{23} = s_1(c_{234} d_5 - a_3 s_{23} + d_6 s_5 s_{234})$$

$$\dot{j}_{24} = s_1(c_{234} d_5 + d_6 s_5 s_{234})$$

$$\dot{j}_{25} = c_1 d_6 s_5 - c_2 c_3 c_4 c_5 d_6 s_1 + c_2 c_5 d_6 s_1 s_3 s_4 + c_3 c_5 d_6 s_1 s_2 s_4 + c_4 c_5 d_6 s_1 s_2 s_3$$

$$\dot{j}_{32} = a_2 c_2 + a_3 c_{23} + d_5 s_{234} + \frac{d_6}{2}(c_5 s_{234} - c_{234} s_5 - s_{2345})$$

$$\dot{j}_{33} = a_3 c_{23} + d_5 s_{234} + \frac{d_6}{2}(c_5 s_{234} - c_{234} s_5 - s_{2345})$$

$$\dot{j}_{34} = d_5 s_{234} + \frac{d_6}{2}(c_5 s_{234} - c_{234} s_5 - s_{2345})$$

$$\dot{j}_{35} = -\frac{d_6}{2}(s_{2345} + c_5 s_{234} - c_{234} s_5)$$

$$\dot{j}_{42} = s_1$$

$$\dot{j}_{43} = s_1$$

$$\dot{j}_{44} = s_1$$

$$\dot{j}_{45} = c_1 s_{234}$$

$$\dot{j}_{46} = c_5 s_1 - c_1 c_{234} s_5$$

$$\dot{j}_{52} = -c_1$$

$$\dot{j}_{53} = -c_1$$

$$\dot{j}_{54} = -c_1$$

$$\dot{j}_{55} = s_1 s_{234}$$

$$\dot{j}_{56} = -c_1 c_5 - c_{234} s_1 s_5$$

$$\dot{j}_{65} = -c_{234}$$

$$\dot{j}_{66} = -s_5 s_{234}$$