Navigation with Local Sensors in Surgical Robotics

Von der Universität Bayreuth
zur Erlangung des Grades eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

vorgelegt von

Philipp Stolka

aus Tychy/Polen

1. Gutachter:

2. Gutachter:

Tag der Einreichung:

Tag des Kolloquiums:

Prefix

- About 800, Jabir ibn Hayyan, later known as Geber, was educated reading translations from Greek and based his chemical system "*on two substances: sulphur, which...is hot and dry, and mercury, which is cold and wet. Since each contains all four elements, any other material can be formed by the proper combination of these two, and since we cannot know substance but only form, our search must aim at the most desired product, gold*". This is the most perfect, most virtuous product since, as Aristotle said, all things, even base metals, struggle upward.

- "To do two things at once is to do neither." – *Publilius Syrus, Roman slave, first century B.C.*

- "The past only exists insofar as it is present in the records of today. And what those records are is determined by what questions we ask. There is no other history than that." – *John A. Wheeler, 1982*

Abstract

Using robots in medicine and especially in surgery requires an adequate representation of and reaction to a changing environment. This is usually achieved by modeling the environment at different representation levels throughout the process, ranging from complex 3D imaging modalities which reflect the environment geometry to finding appropriate low-level control parameters for actual motion through environment regions. In this work, a common framework for different types of navigational problems in surgical robotics is proposed, and validated by the introduction of navigation cycles on novel local sensors.

Currently industrial (and surgical) robotic systems employ almost exclusively static global maps – if any – for navigation and planning purposes. Additional information – intra-process, spatial, current, and persistent sensor data – is useful to cope with uncertainty, measurement errors, and incompleteness of data. Between global pre-operative navigation and control, this work introduces the concept of intra-operative navigation on local sensor data into surgical robotics. This includes the creation and maintenance (both concurrent as well as independent) of local environment maps for navigation purposes. This intermediate level of sensory feedback and processing allows to react to changes in the environment, based on persistent but incremental mapping. Furthermore, local sensors permit intra-operative sampling of additional information which may be unattainable before process execution, or available only with reduced precision.

This work proposes to augment robot world models by introducing such local sensors (in particular, force and sound as well as ultrasonic sensors, all of which provide data from an estimated local ε-environment) and to build precise maps from local sensors, which serve as input for several introduced navigation algorithms. This map-building is improved by precise data localisation and precise data insertion. The general idea of nested control loops is illustrated on the basis of a specific surgical application – robot-based milling at the lateral skull base.

Zusammenfassung

Die Nutzung von Robotern in der Medizin und insbesondere in der Chirurgie erfordert eine angemessene Darstellung der Umgebung sowie eine entsprechende Reaktion auf sich darin ändernde Eigenschaften. Üblicherweise wird dies erreicht durch Umweltmodellierung auf verschiedenen Repräsentationsebenen innerhalb des Prozesses, von komplexen 3D-Bildgebungsverfahren, welche die Umweltgeometrie abbilden, bis hin zu Regelungsparametern niedriger Ebenen für die tatsächliche Bewegung durch die Umgebung. In dieser Arbeit wird ein umfassender Rahmen für verschiedene Arten von Navigationsproblemen vorgestellt und anhand der Einführung von Navigationszyklus basiered auf neuartigen lokalen Sensoren validiert.

Heutzutage verwenden viele industrielle und medizinische Robotersysteme allenfalls statische, globale Karten zur Navigation und Planung. Zusätzliche Informationen – prozessbezogene, räumliche, aktuelle und persistente Sensordaten – sind hilfreich im Umgang mit unsicheren, ungenauen oder unvollständigen Daten. Diese Arbeit stellt das Konzept der intra-operativen Navigation auf lokalen Sensordaten vor, welches sich zwischen globaler prä-operativer Navigation und Regelung einordnet. Dies beinhaltet die Erstellung und Aktualisierung von lokalen Umgebungskarten für Navigationsaufgaben (sowohl mitlaufend wie auch unabhängig). Diese Zwischenstufe sensorischer Rückkopplung und Verarbeitung erlaubt die Reaktion auf Umweltveränderungen basierend auf mitlaufender und persistenter Kartenerstellung. Weiterhin erlauben lokale Sensoren die intra-operative Aufnahme von Information, die vor Prozessausführung nicht oder nur mit verminderter Genauigkeit verfügbar wäre.

Die vorliegende Arbeit schlägt die Erweiterung von Roboter-Weltmodellen mit Hilfe solcher lokaler Sensoren vor, insbesondere mit Kraft-, Schall- und Ultraschallsensorik und deren Daten

aus einer zuvor bestimmten lokalen ε-Umgebung. Damit werden präzise Karten von lokalen Sensoren erstellt, die als Eingabe für verschiedene vorgestellte Navigationsalgorithmen dienen. Diese Kartierung wird optimiert durch genaue Datenlokalisierung und -eintragung. Die allgemeine Idee verschachtelter Navigationszyklen wird anhand einer konkreten chirurgischen Anwendung illustriert: robotergestütztem Fräsen an der lateralen Schädelbasis.

Acknowledgments

This work is a result of the project „Robot-based navigation for milling at the lateral skull base (RONAF)“ of the special research cluster „Medical navigation and robotics“ (SPP 1124) funded by the Deutsche Forschungsgemeinschaft (DFG) over the years 2002 through 2008, performed at the

> University of Bayreuth, Chair for Applied Computing Science III (Robotics and Embedded Systems) in Bayreuth, Germany under Prof. Dr. Dominik Henrich (formerly at AG Embedded Systems and Robotics/RESY, Technical University of Kaiserslautern, Germany),

in cooperation with the

> Universitäts-HNO-Klinik (Abt. HNO-Heilkunde) in Heidelberg, Germany under Prof. Dr. Dr. h.c. P. K. Plinkert with Dr. med. Philippe A. Federspil (formerly at Universitäts-HNO-Klinik Homburg, University of Saarland, Homburg/Saar, Germany).

Ultrasound-related device development and support was received from the

> Fraunhofer Institute for Bio-Medical Technology (FhG IBMT) in St. Ingbert, Germany, in particular Dipl.-Ing. (FH) Steffen Tretbar.

Further information about the project can be found at

> `http://www.ai3.uni-bayreuth.de/projects/ronaf/`

Aus Europa

At this place, I want to express my gratitude to the following:

Dominik "DCD" Henrich: For being my long-time academic advisor and mentor with lots (and lots, and then some) of patience.

Philippe A. "Featherplay" Federspil: For clinical advice and for making ultrasound a priority.

Steffen Tretbar: For faith in our experimental setups and for ultrasound.

Friedrich M. Wahl: For being so interested in this work as to agree to become a reviewer.

Michel "M-Dabbeliu" Waringo: For robot taming and long-time support, collaboration, and friendship.

Thorsten "Techno" Gecks: For widening horizons.

AI3/RESY: For being an outstanding lab peer group.

Emad "LB" M. Boctor & Russell H. Taylor For the awesome opportunity and trust.

N.A.: For the final caring push and constant belief.

The family: For being family, and for insisting.

MM,XX,AC,FW : For being here, there, for pushing, and support (deserved&undeserved).

Erklärung

Hiermit versichere ich, Philipp J. Stolka, dass ich die von mir vorgelegte Dissertation „Navigation with Local Sensors in Surgical Robotics“ selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Declaration

Hereby I, Philipp J. Stolka, declare that this present thesis "Navigation with Local Sensors in Surgical Robotics" was independently authored by me and that no sources and tools other than the listed ones have been used.

May 25 2011, Baltimore MD, USA — Philipp J. Stolka

Date, City

Contents

Chapter 1

Introduction

This chapter serves as an introduction to the presented work and structures the considered field. First, a motivation for investigating robot-assisted sensor-based surgical interventions at the lateral skull base will be given (Section 1.1). Then, a goal can be formulated that serves as a guide for research (Section 1.2). Together, they help to outline both the task at hand (Section 1.3) and the limits of the presented work (Section 1.5). Finally, an overview guides the reader through the presented work (Section 1.6).

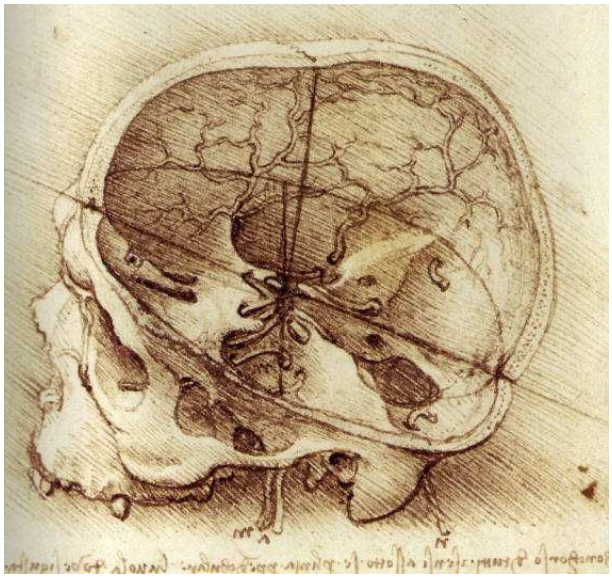


Figure 1.1: Drawing by Leonardo da Vinci of an opened human skull.

1.1 Motivation

The motivation for the research described in this thesis, and pursued in the surgical robotics field in general, derives from the convergence of two areas of interest, which shall be introduced in the following: Surgery and its changing methods (Section 1.1.1), and navigation in robotic applications (Section 1.1.2).

1.1.1 Evolution of Surgery

Surgery has a long history. The invasive therapy of head ailments by means of directly treating the physiological cause can be traced back to prehistoric times, where archaeological evidence even shows skulls with cleanly defined bone cuts after brain surgeries which – proved by their

healed boundaries – were survived by the patients (cf. modern examples in Figure 1.2). This practice was widespread in time and geographic location [brain-surgery.com], including neolithic findings of 7000 B.C. in Europe, Egyptian scrolls of 3000 B.C. referring to therapies based on skull trepanations, Pre-Incan tools of 2000 B.C. made from bronze and obsidian which were used for medical brain surgery as well as for spiritual reasons, Greek texts about both the causes and the surgical therapy of neurological illnesses and traumatic head injuries authored by Hippocrates (460–370 B.C.), Roman surgeons like Aulus Cornelius Celsus in the first century A.D. who therapeutically operated and described depressed skull fractures and symptoms, Chinese physician Hua Tuo (of the Three Kingdoms Era, approx. 221–280 A.D.) who is believed to have been knowledgeable about brain pathologies, Asian surgeons from historic Turkey, the Byzantine Empire, and the Islamic world, both in the Middle East and Spain, and finally Christian surgeons, who in the Middle Ages were mostly to be found in the clerical world of monastic background, but who also guarded medical knowledge, leading a development which culminated in Leonardo da Vinci's intricate anatomical understanding, epitomized in his famous sketches of the human body (Figure 1.1).

Figure 1.2: A trepanation tool from an early Northern American Colonial settlement (left), skull with non-healed trepanation marks (center), closed craniotomy site with coarse but healed cut borders (right) (author's photographs from the Smithsonian's National Museum of Natural History).

Today, robots are widely used in general surgical settings. They assist in passive tasks like camera holding as well as in more active situations like servoing of fixtures, endoscopic guidance, and actual tissue interaction [Taylor03][Speich04]. Such interaction includes both open interventions with direct access to the operation site (*situs*), and deeper-lying, occluded interaction through small incisions (*minimally invasive surgery/MIS*). Recent developments aim to render visible external incisions unnecessary by entering the body through natural orifices (*natural orifice translumenal endoscopic surgery/NOTES*). While the first real applications of surgical robotic systems have been soft tissue interventions (the first robotic tissue removal was a transurethral prostatectomy with PROBOT in 1991), orthopedic operations have been long common as well (well-known examples include ROBODOC, CASPAR, or Acrobot, Figure 1.3). Machining bone structures is easier compared with soft tissue interventions, since the bone can be either locked in place by fixtures or, more recently, tracked in real-time, and treated as a rigid body. Nevertheless, contrary to conventional machining, the subject of interest in surgical applications is a human patient, whose anatomy is highly individual and thus necessitates careful planning based on pre-operative data.

The respective strengths and weaknesses of humans or robots in surgical settings are listed in Figure 1.1. It is evident that both complement each other to a large degree. In the light of these properties the present work aims at combining the strong sides of both in a robotic system for surgical assistance.

Altogether, the advantages that can be expected from the use of surgical robots can be summarized in three main points [SpringerRobotics], page 1201:

- Robots have the potential to improve surgeons' *technical capability* and to enable previously impossible interventions.

*Figure 1.3: Orthopedic robot systems: Autonomous systems ROBODOC (IBM / Integrated Surgical Systems Inc., now part of Curexo Technology Corporation/*www.robodoc.com*) and CASPAR (former Orto-Maquet GmbH), cooperative system Acrobot/Sculptor (Acrobot Co., now part of Stanmore Implants Worldwide, Ltd.) (from left to right).*

- Their use promotes *surgical safety*, both through motion precision (exact positioning, "no-fly zones") and greater sensor and decision support in integrated surgical systems.
- The use of robot- and computer-assisted surgical systems promotes *execution consistency and documentation* through exact navigation capabilities and post-operative assessment data.

1.1.2 Navigation in Robotic Surgery

Many robotic applications rely on navigation to perform their tasks. With preprogrammed motions evoking the impression of autonomous activity, and with reasoning providing a certain level of insight, a robotic system embedded into an actual environment can only execute complex tasks when it has knowledge of its surroundings. This knowledge can stem from sensor information which is stored in maps to allow the system to access previously acquired knowledge for reasoning purposes. In particular, robotic applications with initially unknown or even changing environmental properties require a precise and up-to-date representation of the environment in order to fulfill specific navigation tasks like e.g. safe path planning. This representation has to encompass several orders of abstraction (occupancy, tissue type, interaction behaviour, simulation parameters, ...), precision (planning in free space vs. avoidance of sensitive tissue regions) and timeliness (difference between static and dynamic data). Thus, data sampling occurs at different instants of time during process planning and execution.

Currently, stationary robots – including both industrial and medical robotic systems – and to some extent mobile robots almost exclusively employ static maps for navigation purposes. These maps can be built from rangefinders like laser or ultrasound imaging (US), from cameras or camera arrays, or from computer tomography (CT) or magnetic resonance tomography (MRT) for medical applications, and from many other sources.

One can clearly differentiate two kinds of situations: this map can be available before process execution starts (*pre-operatively*), or it has to be built, completed or maintained during execution (*intra-operatively*). Furthermore, two conceptually different types of sensors can be used: *global sensors* which collect data from a large area and return it with associated position information (the data is embedded into a coordinate system, e.g. in radar imaging), and *local sensors* that are more restrained in their range – they collect only data from the immediate neighbourhood of the sensor interaction location.

	strengths	*weaknesses*
humans	• excellent judgment ("common sense") • able to integrate and act on multiple sources of information • versatile and able to improvise	• prone to fatigue and inattention • limited manipulation ability and dexterity outside natural scale • limited geometric accuracy
robots	• excellent geometric accuracy • untiring and stable • able to integrate and act on multiple sources of numerical and sensor data	• poor judgment • hard to adapt to new situations • limited haptic sensing • limited ability to integrate and interpret complex information

Table 1.1: Comparison of relative human and robot strengths and weaknesses in surgical settings (adapted excerpt from [SpringerRobotics], page 1202)

These two differentiations are also reflected in the two main navigation concepts for current surgical robot systems (also cf. Figure 2.2[1]). One is global spatial data sampled during a planning phase before process execution, which is then used statically for global navigation. The other is local data sampled during the process, which is fed back and used in open or closed loop controllers of the process in a non-spatial context. While facilitating planning, the former usually lacks resolution, segmentability, or timeliness, while the latter only persists during the instant of sampling and is discarded immediately after entering into the control cycles.

However, there exist applications, especially with autonomous robots, for which an additional information type – intraoperative, spatial, current and persistent sensor data – may be useful to cope with uncertainty, measurement errors, and incompleteness of data.

1.2 Problem and Goals

Within the context of robotic surgery, current deficiencies and the aims of the present work will be discussed next – in short, the central problem will be outlined.

1.2.1 Shortcomings of Current Systems

One main obstacle to the widespread use of robots in the operating room is the limited interaction capability of many early-generation systems. With development driven by an engineering perspective, their usability, let alone unobtrusive integration into established surgical work-

[1]This table does not aim to give an exhausting listing of medical imaging devices, only those that are regularly used because of their intra-operative availability. For a deeper discussion of these and other medical imaging modalities – including X-ray, functional magnetic resonance imaging (fMRI), positron emission tomography (PET), single photon emission computed tomography (SPECT), X-ray fluoroscopy, C-arm CT (CACT), and various optical imaging methods see [Yaniv06].

flows, was only addressed as an afterthought. Not being the active part of autonomous robotic interventions any more proved difficult to accept for the surgeons facing this new approach. Although several approaches to "hands-on robotics" in the operating room (*OR*) were explored, a general aversion to yielding control and, ultimately, responsibility to the machine was prevalent, due to a great extent to their *pre-programmed, semi-autonomous motions*.

Another aspect impeding the early acceptance of robotic surgical assistance was the preeminence of special-purpose solutions, whose exceedingly high cost, steep learning curve, and limited applicability seriously undermined their standing even in the research community for some time in the early 2000's. Clearly, this type of autonomous robots adapted from an industrial computer-aided manufacturing (*CAM*) scenario was not the optimal way to bring the robots' capabilities into the OR.

Instead, the current second generation of robotic assistants aims to combine the robots' precision and endurance with the surgeons' superior planning and sensing ability. While surgeons are limited in their positioning precision when it comes to small-scale or long-duration interventions, robot-based systems are still hard pressed in replacing the human visual, tactile, and acoustic senses, not to mention the commonsense necessary to deal with unforeseen navigational and strategic decisional scenarios. Thus, *tele-operated systems* like Intuitive Surgical's daVinci system [Guthart00] are increasingly common tools, allowing high dexterity in previously inaccessible or highly constrained regions through minimally-invasive surgery (*MIS*) while leaving complete control with the surgeon. Another approach is the use of *cooperative robots* – which are controlled by the surgeon, also called "hands-on compliant control" – through some sort of force-following scheme, and enforce limits (geometrical, force, or others) by virtue of motorized or braked mechanics at the same time.

Still, these solutions suffer in different degrees from the limited sensing abilities of the robotic system (Table 1.2). Their ability to position the tools, probes etc. precisely in the intervention area or region of interest (*ROI*) does not offset their blatant lack of concurrent, intra-operative sensing to control and modify the currently executed and future planned actions. These systems are mostly "blind" to sudden or gradual changes within the ROI and have to be supervised by the surgeon, so it is possible to revert to a conventional, manual procedure at any time.

1.2.2 Aims of the Present Work

Therefore, to leverage the robots' superior precision while improving their acuity to their environment, it is mandatory to implement sensory capabilities going beyond the state of the art. The latter currently consists of essentially navigating on static snapshots of the environment, while occasionally updating them in an intrusive, asynchronous fashion, and "flying by instruments" in between. The final goal in sensor-based surgical robotics should thus be a sensing, "omniscient" or at least "panoptic" surgeon-robot, fully and immediately aware of the current developments in the ROI, which should be able to react and re-plan in real-time according to the application's goal and safety requirements. Supplementing the global, static sensors like CT or MRI as planning input, the present work argues that the (missing) final building block for intelligent surgical robot systems might be the use of a second, under-represented class of sensors – eventually introducing *navigation on maps from local sensors*.

These sensors can serve as a concurrent or on-demand source of information about the ROI, later to be assembled concurrently or during an independent step into current maps. However, additional sensors alone only raise the system's awareness of potential risks. For an autonomous system to react properly, or for a cooperative system to provide for proper risk-avoiding guidance to the user, these sensors' current data needs to be associated with geometric locations inside the ROI, and future trajectories infringing on risk regions should be prohibited or re-planned. In doing so, future systems can provide human surgeons with tools and robots far superior to

	CT, MRI	*US*	*Laser Scans*	*Localizers*
spatial resolution (+ image dimension)	high (3D)	medium (2.5D/3D)	high (2.5D)	high
reliability	high	noisy imaging	easily disturbed	depends on technology
discernible features	bone + soft tissue (CT); soft tissue + bone (MRI)	soft tissue (+ bone)	surface features (+ tissue)	no environment information
suitability for planning	directly usable	map reconstruction necessary	only for surface features	only minimally suitable
suitability for intra-operative navigation	difficult (CT); slightly less difficult (MRI)	good (for registration, tracking)	good (for registration)	good (for registration)
latency	non-real-time	high (intermittent)	high (intermittent)	low
temporal resolution	n/a	low...high	low (depends on strategy)	high
sampling cost	high	relatively high	relatively high	low
patient stress	high (CT: radiation; MRI: time, noise)	low	low	none

*Table 1.2: Comparison table of some predominantly used CAS/RAS sensors available for intra-operative use (computed tomography/*CT*, magnetic resonance imaging/*MRI*, ultrasound/*US*, and localizers)*

today's instruments, enabling them to perform interventions in previously inaccessible regions or with less risk of damaging vital structures, through their inherent ability to gather, structure, and process risk information so dangerous actions are reduced or eliminated.

Current systems have to go to considerable lengths to ensure proper registration between patient, tools, and imaging data. In case of patient or tissue shifts (motion of or in the ROI because of inadequate fixation or soft tissue manipulation), this registration may be lost and would need to be retrieved carefully, if this is detected at all. Although registration loss is not addressed by the introduction of local sensors, its unwanted results can be mitigated by controlling tool behavior or position on-line with local sensor information. Furthermore, future research can lead to the introduction of "intelligent tools" – hand-held tools outfitted with local sensors – to support conventional interventions as well.

The goal of the presented work is thus to establish this new class of so-called local sensors in surgical applications, and to equip a robotic surgical system with them in order to build and use maps from local sensors. The benefits are twofold: First, the system will be safer and more precise than comparable ones without local sensor support. Second, the system will allow to drop the requirement for pre-operative imaging altogether in some cases.

1.3 Task and Contributions

Overall, the present work tackles two problems. The first task will be to implement and investigate a complete, new so-called "local navigation" cycle based on local sensors in a surgical robotic system. It describes how this kind of local information can be used together with the other navigation and control modes in a consistent manner, i.e. how it is integrated into a common handling strategy. The second task will be the use of these new local sensors in a "global navigation" cycle.

Achieving these tasks can be split into the following distinct steps. For each, the *scope of the sub-task* is described first, then followed by an outline of the *scientific contribution* and the *challenges* to be handled, and finally the *benefits* brought about by the results of each subtask.

1. *Definition of local sensors and evaluation criteria:*

   - First, it is important to define the range of sensor types to be covered in the following research. The new class of "local sensors" encompasses a wide range of hugely disparate sensor modalities, which can be classified according to several different criteria (tool-dependence, localizability, and dimension). This first step also includes specification of the evaluation criteria that local sensors must satisfy to be considered a useful addition to the current array of options.
   - There exists no clear definition or classification of local sensors so far; therefore the first goal of the presented work is the formal introduction of this term. Based on existing sensors in the application field, relevant evaluation criteria will be deduced to be applied to new local sensors. Since there is only little previous work on the use of what is here termed "local sensor data in CAS/RAS", the decision what is a "useful sensor" must be evaluated on a case-by-case basis for each investigated sensor.
   - Given that many – sometimes conflicting – interpretations of the term "local sensors" can be brought forward, it is imperative to clearly derive a definition for the object of this work, including the differentiation between sub-classes of local sensors. This will make it easier to discuss the scope of the present work.

2. *Development of local sensors and classification methods:*

   - Based on the investigated surgical application chosen for the demonstrator, appropriate local sensors need to be developed. This involves choosing physical sensors capable of capturing meaningful data from the ROI, and investigating classification procedures to extract states of the intervention which are relevant to the system. In particular, this work's focus will be on force-torque and audio sensing, with Bayes and Nearest-Neighbor classification performing the detection of previously identified states. A-Mode ultrasound will be used as another (tool-independent) sensor modality.
   - Again, since there is little prior work concerned with the detection of relevant intervention states in CAS/RAS, it is necessary to define a range of relevant states for the investigated application first, and then evaluate existing physical sensors and subsequent classification methods. One contribution of this work is the first use of local sensors in the investigated application.
   - Going beyond the current use of online sensor data for pure control purposes, the proposed local sensors will provide contact state information which may be available to the human surgeon in conventional (manual) procedures, but is currently going unused in autonomous robotic interventions. This offers another dimension of safety by constraining operation actions which is currently not available.

3. *Development and testing of (re-)registration procedures*:

   - Although local sensors (more precisely, their data) themselves do not need to be registered with the patient or prior image data, an initial positioning of the robot (if one is used) with respect to the patient is necessary before intervention and concurrent map-building can take place. Thus, registration procedures to define the ROI within the robot and localizer workspaces need to be developed. These will include robot force-following as well as pointer-based demonstration procedures, with availability depending on the specific system configuration.
   - Current registration methods are mostly aimed at one of two scenarios: Either registering one imaging modality to another, or retrieving the patient's current position with respect to the prior imaging data. In the case of navigation on local sensor data, however, a prior map need not exist at the time of intervention start. Therefore, in the absence of pre-planned intervention regions, new methods to interactively define the ROI intra-operatively must be devised. As dictated by the application needs, several different image-less registration methods will be investigated.
   - The proposed markerless and/or ultrasound-based registration methods greatly reduce patient stress and simplify the pre-operative workload necessary to perform image-based and robot-assisted surgery compared with current standard procedures.

4. *Build maps using local sensors:*

   - Once the local sensors start passing through the ROI, their incoming data can be synchronized and combined with localizer information to be entered into a persistent "map from local sensors" which is built and updated in real-time. Since local sensors exhibit a variety of shapes of their sensing components (either through dedicated sensor areas or as a by-product of tool interaction), their data can be localized with varying precision, depending on additional knowledge. This will result in a variety of mapping functions between spatially and temporally ordered sets of data samples. Particularly interesting is the *destructive mapping* aspect of certain sensing principles, i.e. that the mapped region may irreversibly change features after being explored (which is application- or tool-dependent, e.g. through milling). Occupancy mapping is one viable option to store information geometrically.
   - Since there are no previous works on concurrently-built maps from local sensors, synchronization and localization are important topics covered by this work. Especially the blur-inducing convolution of sampled information with the sensor shape proves a challenge to cope with by implicitly de-convolving the information based on tool shape and direction data to extract the maximum amount of information. This is closely related to deciding on proper map discretization. Furthermore, local sensors yield only information from a limited local sensor range, so the destructive nature of some mapping processes (through milling) will make gathering information difficult, because most of the described environment will vanish immediately. For other processes (ultrasound sensing), geometric reconstruction of the complete neighbourhood from the single discrete sensing results is important and non-trivial.
   - These maps from local sensors will allow both humans and autonomous systems to consider environment data which has been collected concurrently and in a minimally intrusive fashion during the process. Thus they provide a valuable basis for cross-checking the intervention progress with previous plans.

5. *Procedures for navigation on local sensor data:*

   - While such a map from local sensors can be useful for a post-operative comparison with reference pre-operative planning images, in order to serve as support during the intervention itself it needs to be integrated into on-line navigation procedures which control and modify the robot motions. Existing planning procedures include pre- and intra-operative navigation cycles on global data, which then can be supported by the proposed maps from local sensors serving as planning data in an intra-operative local navigation cycle.
   - For the specific application considered in this work, the pre-operative planning component of the proposed system computes complete milling trajectories for the robot. The intra-operative modification or "navigation on local sensor data" will then adapt these trajectories to increase safety during their execution (the system performs only simple adaptation because complete re-planning would complicate the safety checking). This will include locally restricted re-planning procedures, which need to account for path integrity and patient safety. Depending on the specific local sensor (tool-mounted for concurrent sampling or tool-independent for independent sampling, cf. Section 4.1), this re-planning may be performed once or concurrently. A classification of these navigation strategies includes:
     - R_{once}: modification of the trajectory occurs *only once* at the beginning of the intervention in an independent step,
     - $R_{concurrent}$: *concurrent evasion*, i.e. with trajectory modifications occurring continuously after each detection of dangerous regions,
     - $R_{iteration-wise}$: *iteration-wise evasion*, i.e. synchronized at several discrete points in time after every completion of one milling sinking iteration.

     To avoid complications with real-time requirements, only R_{once} and $R_{iteration-wise}$ will be examined in this work.
   - Formerly "blind" autonomous robotic systems which relied exclusively on pre-operative data will now be able to modify their intervention trajectories depending on their current view of the region of interest. This will increase safety by incorporating the most current planning information throughout the whole intervention.

Altogether, the proposed work will result in a surgical robotic system whose navigational capabilities can be adapted to the current intervention in a modular fashion. As described in the following section, this system will be able to perform milling interventions at the lateral skull base. Using the proposed approach, it will be possible to perform less invasive, safer, and/or more precise interventions than current procedures allow. In particular, it will become possible to perform the investigated surgical interventions without computed tomography support.

1.4 Investigated Application

For the problems described in the preceding section, it is important to validate them in a realistic operational setting. Therefore, the application serving as a testing ground for the demonstrator system is described in this section. This comprises both medical (Section 1.4.1) as well as technical (Section 1.4.2) considerations.

1.4.1 Medical Application

There exist certain pathologies of the middle and inner ear which result in loss of hearing (deafness) and cannot be treated successfully with conventional hearing aids. In these cases, the patients can receive surgical therapies based on implantable hearing aids or cochlear implants. The former type mechanically stimulates the ossicles which then communicate the vibration further to the inner ear and the tiny hair populating the cochlea, while the latter directly transmits electrical impulses into the cochlea via an electrode-studded wire to excite auditory nerves directly. Depending on the size of the implantable component, both may require the removal of bone volumes from the skull calotte and the calvarial bone or *mastoid* (Figure 1.4).

Figure 1.4: Approximate location of the amplifier/transceiver component of implantable hearing aids or cochlear implants (left; from placement guide for Vibrant Soundbridge from Med-El GmbH); postauricular skin flap opening exposing the mastoidectomy location (right; from "Atlas Of Neurotology And Skull Base Surgery", R. Jackler, 1st ed.).

The scope of application investigated in this work is cochlear implantation, in particular the automated milling of cavities in the skull bone for subdermal implantation of hearing aids. This is to be performed by the medical robotic system RONAF ("Robot-based navigation for milling at the lateral skull base") [Henrich02][Federspil03]. One step of the process is the removal of bone material from the thin calotte in the shape of the flat amplifier/transceiver components. As of 2010, three US Food and Drug Administration (FDA)-approved multichannel cochlear implant systems (i.e. comprising multiple electrodes on the cochlea-inserted component) are routinely used in the United States, including the Nucleus 5 cochlear implant system (Cochlear Corporation), the Clarion 90K (Advanced Bionics Corporation), and the Combi 40+ (MED-EL Corporation) (Figure 1.5, [Megerian10]).

*Figure 1.5: Subdermal amplifier components of cochlear implants (*CI*) and implantable hearing aids (types from left to right: Combi40+ by Med-El (CI), Clarion by Advanced Bionics (CI), TICA by Implex, Vibrant Soundbridge by MED-EL/Symphonix).*

A major difficulty in these interventions is the necessary precision together with the safety requirements. Interventions of this kind demand extreme precision (with tolerances well below one millimeter) in spite of the high force required to remove larger quantities of bone. For congenital hearing loss cases, implantation at the age of 12 months is recommended. For these cases in particular, bone thickness is minimal, but CT imaging is counterindicated; and in

general, radiation doses should be minimized or avoided even in adult patients. Nevertheless, high-resolution CT represents the main modality for both revealing counterindications and abnormalities requiring modification of the standard insertion procedures, as well as for procedure planning itself. In some cases, this may be complemented or substituted by T2-weighted fast spin echo MRI for better discrimination between bone and nerve tissues and fluids; however, availability and cost may prohibit its use.

The initial operation phase (of a conventional, manual operation) consists of preparation of the intervention site (opening a postauricular skin flap) and definition of the milling site by means of mechanical templates (alternatively, when using computer-assisted navigation systems, this site may have been defined pre-operatively based on planning data) (cf. Figure 1.6). After this, a prolonged period of bone milling follows, in which the surgeon needs to ream out comparatively large quantities of hard bone for the bulk of the implant bed cavity (also *implant well*), which is a both physically and psychologically exhausting task. Then, this cavity needs to be finished in detail with high precision to ensure an optimal fit of the implant with the surrounding bone. This may include the creation of dural islands (regions where the dura is exposed), especially in children with low skull bone thickness. Also, it may be indicated to create a freely-moving bony island sitting on the dura. During this phase of the (manual) procedure, one problem is the possibility of breaking through the calotte into the skull interior with subsequent rupture of the sensitive meningeal (*dura mater*) enclosing the brain. This can result in complications like infections or longer reconvalescence. A third phase comprises the reaming-out of the mastoid process (*mastoidectomy*) to allow access to the cochlea (for CI) or the ossicles (for implantable hearing aids). Both milling phases may be swapped as well, and include several minor milling and preparation steps not detailed here. Milling is routinely carried out with larger cutting burrs ($3mm$, $6mm$ diameter) for major volume removal and smaller diamond burrs ($1mm$, $2mm$) for facial nerve skeletonization, mobilization of the dural bone island, cochleostomy, and general finishing.

Risks for both cochlear implantation and mastoidectomy are similar and include postoperative infection, facial paralysis, cerebrospinal fluid (CSF) leakage, and meningitis [Megerian10]. Within the scope of this work, it is worth pointing out that the latter three are related to the precision of imaging, planning, monitoring, and execution of the milling procedures.

1.4.2 Technical Constraints

The procedure described above is to be performed with a surgical robot system supported by a variety of sensors, tools, and control computers (Figure 1.7). In particular, the main tool for ablative treatment will be a robot-mounted surgical miller (for specifications and technical details cf. Section A.2).

In the chosen demonstrator application, the milling procedure will be performed through autonomous robot operation. The robot end-effector will be a tool-holder mounted on a 6D force-torque sensor, with the miller instrument accompanied by an A-mode ultrasound probe and an optical tracking system marker, each mounted at a 90-degree angle to either side of the miller.

The milling intervention can – but need not – be planned on pre-operatively or intra-operatively acquired imaging data like computed tomography (*CT*) or 3D ultrasound (*3D-US*). Depending on planning data, time, and safety considerations, different approaches to milling path computation can be chosen – either pre-computed, generic paths specific to the chosen implant volume (and several other parameters, like miller head diameter, inter-path distances, depth feed etc.), or paths computed pre-operatively depending additionally on the intervention area geometry (derived from pre-operative planning data) and its intersection with the generic implant volume. However, all these paths are subject to several parametrizations based on

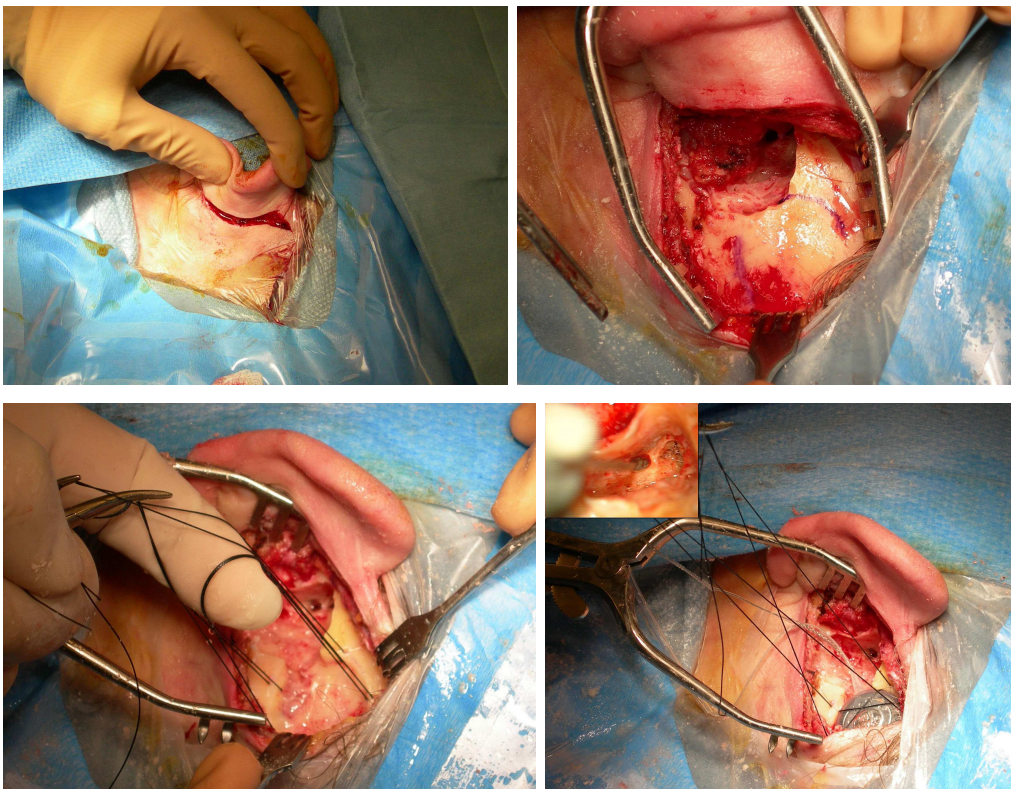


Figure 1.6: Intraoperative views during the manual implantation of a cochlear implant: postauricular incision (top left), mastoidectomy result (top right), implant bed with thin remaining skull layer and introduction of tie-down sutures (bottom left), and fixation of the implant receiver in its bed (bottom right) (images taken from [Megerian10]).

earlier research results:

- The trajectories are computed according to the "*concentric paths*" approach (as opposed to parallel or potential field paths, cf. Figure 1.8) and incorporate the "*milling with sinking iterations*" strategy (as opposed to "layered milling"); resulting in (almost) any-time-capable paths, i.e. such that can be aborted at any of several discrete points of time [Waringo03c]. This results in high operational-*vs*-empty-drive motion time ratios and guarantees a usable cavity at a "sufficiently high" number of intermediate instants of time (for a more in-depth definition, cf. Section 6.2.2).

- The *orientation (2 degrees of freedom (DOF): yaw, pitch)* of the miller head is fixed (Figure 1.9). While later research work by the author's group has shown a positive influence of orientation optimization on resulting forces and on the quality of the milled cavity (personal communication M. Waringo), a standard 6-DOF robot arm then covers a considerably larger portion of the robot configuration space, resulting in a large amount of inaccessible path points during path execution, and thus leading to a high rate of path rejections and/or abortions.

- The milling paths are *statically specialized* with respect to their roll orientation angle (Figure 1.10). This fixes the one remaining, unconstrained degree of freedom (*DOF*) of the five DOF milling path (three translational DOF and two rotational DOF are already defined) by setting a constant roll value for all path points [Stolka07a] depending on a static analysis of the actual robot-patient setup in the operating room. Although this strategy is not optimal with respect to minimization of joint (angular) path length, angular velocities etc., it allows simple testing for admissible or non-admissible paths.

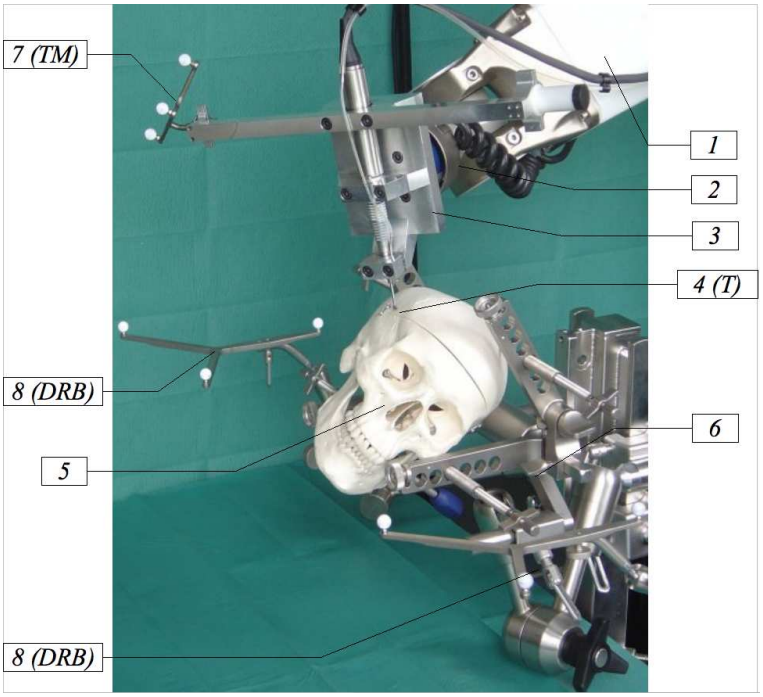


Figure 1.7: Setup of the investigated RONAF surgical robotic system (1: robot arm; 2: F/T sensor; 3: tool holder; 4: miller head; 5: skull phantom; 6: fixation; 7: tool navigation marker; 8: base reference markers).

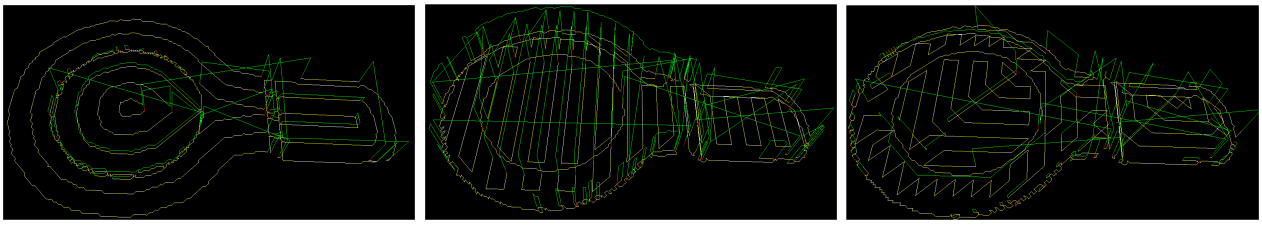


Figure 1.8: Milling paths for the Soundbridge implantable hearing aid (for one sinking iteration; left to right: concentric, parallel, and potential-field based paths).

1.5 Delimitation of Work

The devised demonstrator is comprised of a multitude of components, some of which do not directly contribute to the goals set forth in Section 1.2. In particular, the following procedures or components are not part of the present work:

- *Physical sensor development:* Only existing physical sensors will be considered as bases for the investigated (logical) local sensors[2]. Initially, investigating the field of research with already proven sensors of known characteristics seems to be more promising. Future work can expand on this topic.

- *Tool development:* Similarly, only existing surgical tools will be used in the envisioned robotic system. A robot-based tool holder will adapt the manual intervention tools to the robot system.

- *Trajectory planning algorithms:* These have been investigated thoroughly in [Waringo03a],

[2]The A-mode ultrasound sensor used as one of the local sensors of the presented robot-based system has been specifically developed by the Fraunhofer Institute for Biomedical Technology (IBMT) partner of the RONAF project, which is the basis of this presented work. However, this sensor's specification and development are not within the scope of this thesis.

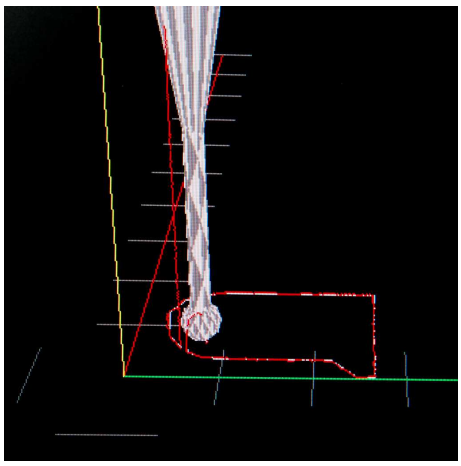


Figure 1.9: Orientation (yaw, pitch) of the miller head, fixed perpendicularly relative to the milling layers.

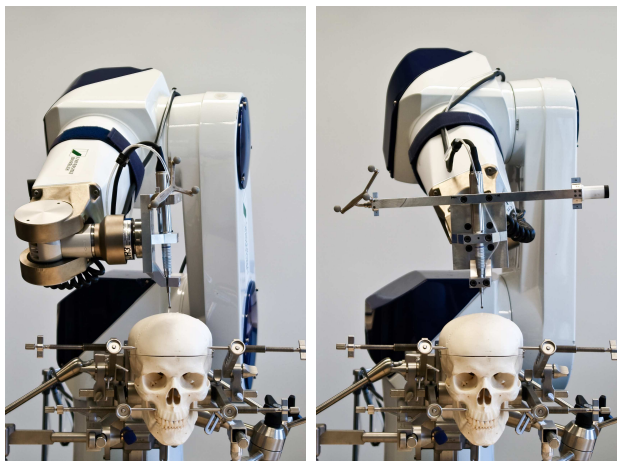


Figure 1.10: Robot poses for different roll angle specializations (left: 90°, right: 0°).

[Waringo06] and others. In the present work, only existing sinking-iteration path planning algorithms will be used.

1.6 Overview

Altogether, this work is thus concerned with the investigation and demonstration of navigation cycles based on novel local sensors. The two main thrusts along which these cycles are to be validated are (1) intra-operative one-time planning and registration (based on 3D ultrasound maps), and (2) real-time map building based on tool-mounted local sensors for concurrent milling path adaptation.

In Chapter 2 "State of the Art", current developments in medical sensing, computer-assisted intervention and navigation, and surgical robotics are discussed. Based on this, Chapter 3 "Proposed Approach" develops the angle of attack to be followed in this work, defines a vocabulary that structures navigation in surgery, and outlines the position of the proposed work within these frameworks. Then, Chapter 4 "Local Sensors" introduces the new class of local sensors and investigates several of them. In Chapter 5 "Maps from Local Sensors" their data is combined with localizer information to build maps. These serve as input for navigation algorithms in Chapter 6 "Navigation on Maps from Local Sensors", with overall experimental outcomes and system information also given there. Finally, Chapter 7 "Conclusions" summarizes the findings with a discussion of the achieved results and proposes directions for future work (Figure 1.11).

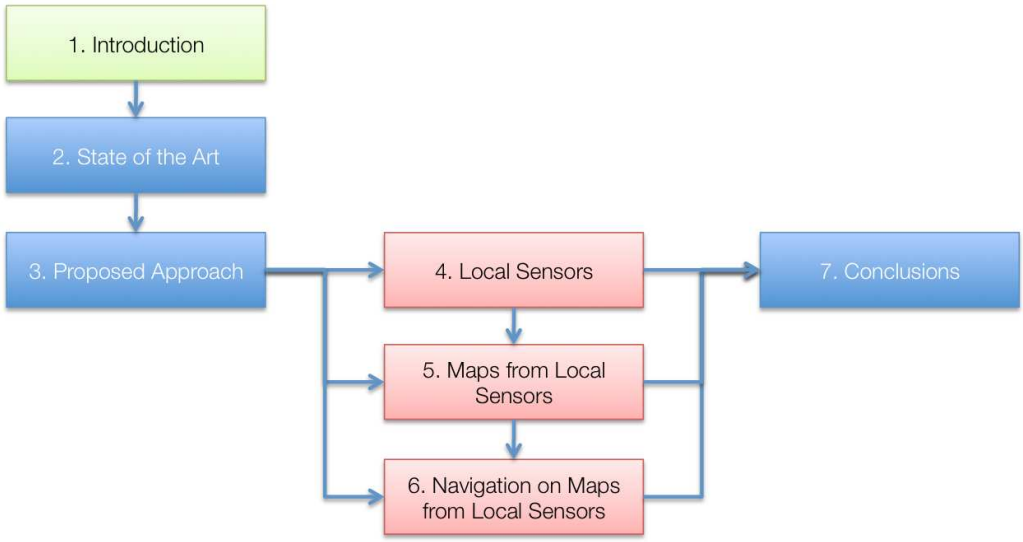


Figure 1.11: Outline of the presented work (main technical contributions in red/center).

Chapter 2

State of the Art

The field of surgical assistant systems and surgical robotics has been developing at an explosive pace since its inception in the late 1980s. As stated in [Taylor03], in the late 1990s it would still have been possible to give a reasonably complete and concise overview of all robotic systems in interventional medicine. A more current review article lists almost 160 different systems, spanning a vast range of such systems from many different interventional and medical sub-fields while concentrating only on systems under development in 2001–2005 [Pott05]. As of mid-2009, around 25 companies populate the market for robotic medical systems, with Intuitive Surgical's installed base of 1,676 *da Vinci* laparoscopic robots (and its successors, the *da Vinci S* and the *da Vinci Si*) [Intuitive] at the forefront of commercial success (Mid-2011). In spite of the relatively young research area, it is thus impossible to completely describe the environment in which the presented work is set. Instead, a more focused approach will be taken here.

In this chapter, the main goals of the presented work will be motivated by looking at the state of the art in surgical robotics and several directly related subfields. First, a general overview of intelligent tools and systems (including robotic systems) in surgery will be given, classifying past and current systems into different groups and outlining their properties. Then, a succession of more specialized sections will delve into navigation, sensors, and particularly local sensors in surgical robotics, giving specific examples.

2.1 Classification of Intelligent Tools and Systems in Surgery

The broad field of intelligent tools and systems in surgery comprises a multitude of vastly different approaches. Systematic categorizations of these systems can be inspired by their respective means of actuation, by their use as tools, or by their capabilities. Each of these will be briefly highlighted in the following.

Classification by Actuation

One classification approach which structures the field of intelligent tools and systems in surgery [Davies00] differentiates between Robot-Assisted Surgery (*RAS*) vs. Computer-Assisted Surgery (*CAS*). The former category encompasses systems based on a surgical robot, which in turn can be defined as "a powered computer-controlled manipulator with artificial sensing that can be reprogrammed to move and position tools to carry out a range of surgical tasks". Naturally, the robot itself depends on an array of other subsystems which perform imaging, planning, and sensing. The robot's main advantages can be summarized as accurate positioning (even under low/no visibility, e.g. in minimally invasive interventions, and over long periods of time) and assistance for repetitive motions (e.g. milling, knot-tying etc.).

Computer-Assisted Surgery

In contrast, the CAS category omits the robot from the system, entrusting the human surgeon with guiding and holding the surgical tools himself. The tools are enhanced with a certain amount of embedded intelligence, which need not necessarily reside in the tools themselves (like in intelligent power tools), but may be located in the CAS system as a whole (e.g. as localization and warning capabilities). Usually, CAS systems ("image-guided surgery"/*IGS*, also simply called "navigation systems") support planning by providing sophisticated segmentation and access planning methods based on pre-operative imaging data, and provide intra-operative guidance of the hand-held tool w.r.t. allowed or forbidden volumes or paths.

One example of this systems class would be a specialized IGS system like the commercial BrainLAB VectorVision platform, allowing tracking of surgical tools relative to pre-operative imaging data.

Robot-Assisted Surgery

Within the RAS category, systems can in turn be classified as powered robots used as passive tool holders, as active robots, as synergistic systems, or as master-slave (telemanipulator) systems.

The first sub-category (*powered robots as tool holders*) is mostly comprised of stereotactic frame enhancements or replacements, i.e. systems which serve as positioning devices in neurosurgical settings. In general, motions of the robots' end-effectors are performed autonomously by the robot, however, the degrees of freedom for the actual patient interaction are under complete manual control of the surgeon. The first applications of robots in surgical settings were in this group – Kwoh (cf. [Davies00]) used a Puma 560 robot to position a drill guide next to the patient's head to assist the surgeon in neurosurgery. Based on a plan with a pre-operative CT, on the day of operation the patient was registered with the robot, which then locked into position in a power-off mode after properly attaining a pose which allowed the surgeon to insert drills and biopsy needles at the correct target locations. A similar functionality is achieved with systems like AESOP (Intuitive Surgical Inc.) for guidance of endoscopic cameras, although they manipulate the camera pose while in contact with the patient.

Active robots (the second sub-category from above) constitute a group of systems drawing some controversy in recent years. Here, powered robots actively interact with the patient, potentially allowing more complex or more precise motions than what can be achieved with passive systems as described above. Safety concerns are far more important for this class of systems because of their autonomy during direct interaction and their handling of surgical tools. Therefore, many approaches develop specially-designed robot kinematics for the particular task under consideration. While laparoscopic camera robots can be subsumed in this class as well, they typically do not execute motions or perform tissue interaction autonomously, which greatly reduces their (actual or perceived) potential for danger. However, the bulk of active robots are autonomous robots as well. Because of the improved positioning capability (its accuracy) and a robotic system's endurance, the main areas where active surgical robots are found are orthopedic interventions.

Prime examples are the ROBODOC system for hip endoprosthesis interventions developed by Taylor et al. in the early 1990s at the IBM Thomas J. Watson Research Center [NIST06], which reached the stage of clinical application (albeit under testing conditions) and was later marketed by Integrated Surgical Systems, Ltd.), or the similar CASPAR system (by the former OrtoMaquet GmbH) for hip and knee prosthetic interventions. Although the scope of CASPAR was eventually extended to anterior cruciate ligament (*ACL*) reconstructions, both systems concentrated on milling interventions in orthopedic applications, in particular on large bones (femur and hip) that can be easily clamped and fixated in position. This renders the investigated

applications similar to conventional, industrial CAD/CAM (*computer-assisted design/manufacturing*) approaches in the sense that they can be pre-planned in detail pre-operatively and then executed autonomously (which amounts to "blind execution" to a large extent). This similarity prompted other groups [Taylor03] to adopt the term "*surgical CAD/CAM*" for this group of systems. Nevertheless, active robots have a foothold in soft-tissue interventions as well. A very early example is the Probot prostatectomy robot by Davies et al. [Davies00] of the late 1980s, based on specially-designed "safety frame" kinematics allowing only the absolutely necessary degrees of freedom for the robot holding and guiding tools for prostatectomies, an intervention which – when executed manually – requires a high degree of spatial imagination on the side of the surgeon. The particular intervention under investigation, although working in a soft-tissue environment, did not require intra-operative imaging and relied exclusively on plans based on ultrasound images taken at the start of the procedure. Another example would be the Remote Center of Motion (*RCM*) robot of the Johns Hopkins University, later developed into the LARS robot, which allows to perform kidney punctures and biopsies with a minimal, specialized robot allowing motorized motions and rotations around a point at the skin puncture position, due to its special parallel-linkage kinematic setup. The Imperial College of London with Davies et al. used a similar setup for the neurosurgery MINERVA system.

The third sub-category of robot-assisted systems are *synergistic systems,* which combine the sensory and cognitive abilities of the surgeon with the positioning accuracy of a robot. In this approach, the surgeon directly controls a robot by guiding it manually through the intervention – also aptly named "hands-on compliant control". One such system is the Acrobot ("active constraint robot") by Davies et al. In a pre-operatively defined region, the robot can be moved around freely by a force-following scheme. Approaching the edges of this safe region, the robot slows down by increasing its impedance relative to the exerted forces. At the edge itself, it comes to a standstill, thus providing a natural safeguard against transgression into dangerous regions. This system has proved valuable in knee prosthesis operations. Troccaz et al. have developed another synergistic system called PADyC (Passive Arm, Dynamic Control) [Troccaz96], which also limits the system's motion into unsafe regions, but does not use motorized components, relying on the surgeon to guide the passive arm within the defined areas, blocking further motion by means of high-bandwidth brakes (i.e. an apparatus based on freewheels combined with electric motors, which allows fine-grained constraining of the joint velocities). This whole approach using defined safety regions to prevent dangerous operations is also known as "virtual fixtures" [Park01].

Finally, in *master-slave (telemanipulator) systems* the operator is in total control of the end-effectors' motions. Using a joystick or similar master device, he guides the robot arms (the slaves) directly, most efficiently with visual feedback. Force feedback has been an active research topic for these systems in recent years as well. Difficulties arise when the distance between master and slave becomes large, resulting in true telemanipulation systems. At the time of writing, one commercially successful device is the three- or four-armed da Vinci by Intuitive Surgical Inc., providing 3D vision for bimanual minimally invasive operations through an endoscope held by the third arm. For such master-slave systems, force feedback (or more generally, sensory substitution, i.e. replacing one type of feedback with another, such as substituting force by visual indicators) is an active area of research, hampered by the potentially long distance between the components. In 2001, the so-called "Operation Lindbergh" (Figure 2.1, [Marescaux02]) was the first trans-Atlantic telemanipulated or "remote surgery" – a laparoscopic cholecystectomy – performed from New York on a patient located in Strasbourg, France, using a Zeus (Computer Motion Inc.) system similar to the da Vinci.

Robot-based systems present inherent safety concerns, but also inherently high accuracy in a well-designed system. Nevertheless, safety requirements and conclusions can be considered to be substantially similar for the CAS and RAS approaches, both of which are means to navigate

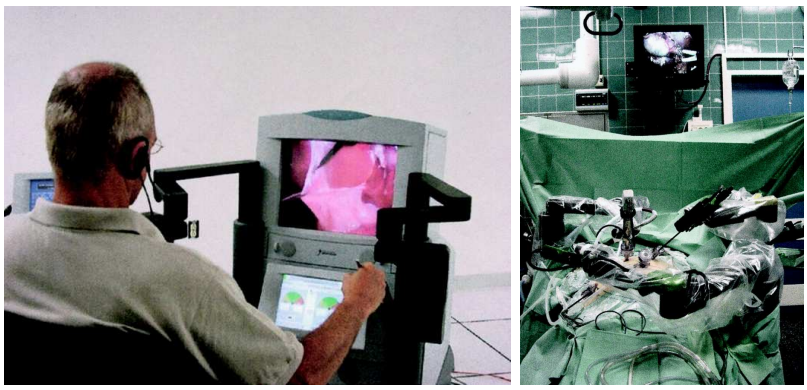


Figure 2.1: Setup of the remote surgery "Operation Lindbergh" laparoscopic cholecystectomy (operator and surgeon's robotic console in New York/left, Zeus robotic arms at the remote site in Strasbourg/right) (taken from [Marescaux02]).

intraoperatively based on image data.

Classification as Tools

Another classification schema is presented in [Dario03], where intelligent surgical systems are viewed as tools which support the surgeon.

Surgical interventions evolve together with the tools they use. Conventionally, the surgeon directly interacts with the patient in the region of interest using conventional, "dumb" tools. This constrains him to either the open surgery approach or minimally invasive surgery (*MIS*) with endoscopic tools. Both, however, offer little support in the generation of motions or the interpretation of data generated by the intervention. The surgeon sees the patient directly, and the feedback he receives is either unaided or even hampered by the constraints of the endoscopic approach (limited visual and force feedback, motion inversion through the fulcrum point of the trocar, limited dexterity etc.).

Endowing the tools of the surgeons with more intelligence, the simplest group of components are *handheld tools* comprising scientific instruments to measure tissue properties or surgical performance (e.g. force- or electrical-impedance-based), *surgical mechatronic tools* which improve conventional tools like graspers or endoscopes (by providing sensing and display functionality, or actuation used for manipulation and error reduction), and directly driven mechanisms which slow down or constrain the surgeon's motions (which can be both active and passive), but provide no additional navigation assistance.

Further along the line of tool evolution, the next category are *teleoperated surgical tools for MIS.* Initially, their development was driven by the idea of the tele-present surgeon who can provide medical assistance to wounded individuals, e.g. in the battlefield. Later, the usefulness of systems offering sensing and actuation at previously inaccessible locations became apparent in more civilian circumstances, too. First, they increase the dexterity of the minimally invasive surgeon operating in constrained spaces; and second, they offer improved ergonomics compared to conventional tools, through both vision enhancement or decoupled kinematics (e.g. the pedal-operated clutching solution to unlock and re-arrange the master and slave arms of the da Vinci system relative to each other). While distancing the surgeon from the intervention physically, control over the tools' motions lies still with the human operator. This category is populated by systems like the afore-mentioned da Vinci or Zeus.

Finally, surgical tools in the widest sense also include *autonomous surgical robots.* They execute specific intervention steps automatically to improve precision, navigation capabilites, and reaction to external stimuli. Their applications include orthopedics, MIS, and soft tissue operations. Although their first use was in a soft tissue environment, orthopedics is the

field where autonomous robots are most readily deployed, because under favorable conditions their advantage of high precision can be realized easily (e.g. working with clamped, fixated bone structures, resembling an industrial machining task). This usually requires very precise matching (*registration*) of pre-operative planning data with the intra-operative actual situation. Systems based on adapted industrial robots like the previously mentioned ROBODOC (Integrated Surgical Systems Inc.) and CASPAR (OrtoMaquet GmbH; both being orthopedic milling robots) reduce the surgeon to mere supervision of the autonomously executed tasks. However, "autonomy" can also extend only to macro-like execution of short, preprogrammed motions like e.g. knot-tying. An interesting intermediate situation arises when the human operator controls one part of a robotic system, while the control component determines the motion of other parts of the same system, e.g. in bi-manual knot tying where the cognitive load can be reduced by having only one robot arm controlled manually, while the second robot arm conforms to spatial and temporal constraints required to complete the task successfully [Kapoor08].

2.2 Related Problems and Applications

In the following, the fields touched upon by the discussed application – related to surgical and navigational aspects of handheld and robotic systems – will be discussed.

2.2.1 Bone Milling Interventions

Physical bone interaction during orthopedic milling interventions can be done with a multitude of different tools, such as chisels, rasps, saws, or rotatory drilling and milling tools. Skull base surgery regularly uses the latter group of tools, with drill bits in the shapes of drilling tips, diamond milling heads, or various bladed cutting burrs. While cutting burrs are able to remove large quantities of bone material within a short time, their blades pose a risk to critical structures. Therefore, in the vicinity of nerves or the dura mater, fine removal is performed with diamond-coated spherical bits, with less bone removal capability, but higher tolerances with respect to critical region transgressions. Other specialized tools include craniotomy saws and trepanation devices, both of which serve to cut the skull calotte while preserving dura integrity, at the expense of limited flexibility in application. In any case, continuous irrigation cooling in the milling region is paramount [Fuchsberger86]. Without cooling, thermal necrosis of the bone and adjacent structures like blood vessels, nerves, dermal structures etc. sets in rapidly due to temperature spikes above $100°C$.

2.2.2 Surgical Navigation Systems

Surgical navigation systems (like infrared optical trackers, magnetic or ultrasound trackers) are useful for tracking the absolute position and orientation (*pose*) of objects (instruments or the patient) within the operating theatre. This ability is employed e.g. for conventional, computer-assisted interventions, where the surgeon performs the action manually while having an enhanced sense of position and orientation of his instruments relative to interesting structures, allowing for more precise or even previously impossible interventions. This strategy obviously requires the collection of preoperative data to compare it with the current instrument pose. Conventionally, this is achieved by acquiring a 3D image and registering it to the operating theatre (e.g. CT or MRI, but also 3D ultrasound or elasticity imaging [Keil09], [Stolka10a]) with some sort of subsequent real-time guidance or video overlay.

For a known and restricted application area, an alternative may be image-less navigation, based on generic anatomical models (atlases) which are registered with and adapted to the

patient's individual features by sampling appropriate sets of surface points (while mostly applicable only to rigid bodies as in orthopaedic settings, this can also be performed on semi-rigid organs such as kidneys using ultrasound surface detection [Seo09]). With either approach, the instrument is being tracked intraoperatively by means of a localizer like optical tracking systems (OTS), electromagnetic tracking systems, or mechanical localizer arms, and co-displayed with the anatomy.

Navigational issues like region avoiding or path planning are usually up to the surgeon, unless an autonomous robotic system is integrated in the loop. However, in an approach called *Navigated Control* [Hein02] position information of a handheld milling tool is used in skull base surgery to modulate instrument power, thus avoiding transgression into critical regions predefined in pre-operative planning data. This can be used e.g. to support the handheld jigless creation of cochlea implant beds in the skull [Strauss05] (improving accuracy fourfold down to $0.2mm$, although without registration to pre-operative data in that work, longer milling duration with Navigated Control, and unclear milling depth handling), functional endoscopic sinus surgery (*FESS*) [Koulechov06] where inflamed tissue from the sinus is "shaved" away with a tracked surgical shaver tool (with dental-splint-based registration under optical tracking, and voxel-based tool projection into the intervention region map; maximum safety region excess $1.8mm$), or spinal surgery drilling [Trantakis09] (with only rare cases where excess milling was $1.0mm$ or more).

2.2.3 Registration

In robot-assisted surgical systems, registration is often related to determining the spatial relationship between pre-operative planning data and the intra-operatively acquired reference set. However, this registration often cannot be assumed to be fixed, or static. It is difficult to completely immobilize the patient, and nearly impossible to do so when dealing with soft tissue. Motion of the patient (*patient shift*) and motion of soft tissue like skin, the brain, or other inner organs (*tissue shift*) are inevitable results.

A straightforward landmark-based approach, although undesirable due to implantation trauma, is to attach bone-fixed markers (*fiducials*) in the intervention area to make them clearly visible in imaging data as well as physically approachable with the robot, as early commercial systems like Robodoc [Taylor94] demonstrated for static (initial) registration.

However, often this is not an option, especially for endoscopic surgery. Furthermore, simple fiducials do not allow dynamic tracking. As non-invasive tracking methods are generally preferred nowadays, preference might given to surface-mounted optical markers for tracking skin motion (such as the Claron MicronTracker [ClaronInc]), clamped markers for head motion (e.g. the VBH dentally-attached head holder [Bale97]), or servoing based on visual clues from video streams [Denis03]. It is also possible to establish dynamic (persistent) registration with optical or electromagnetical tracking markers. All of these result in dynamic registration relationships and aim at the reduction of errors between expected and actual positions of the intervention area.

Alternatives for static registration include point-like anatomical landmarks, defined with pointing devices, or shape-based methods of registering geometrical features from image data, like surfaces, crests, or sets of points. Those methods, possibly combined with atlas databases of deformable organ models, may even make preoperative imaging unnecessary (e.g. in the BrainLab VectorVision knee TKR/VectorVision CT-free, where a handheld optically-tracked pointer intra-operatively samples surface features that adapt an atlas model). However, anatomic atlases are more common for femurs or hip bones, less so for the skull.

Another class of approaches uses an intermediate imaging modality such as tracked ultrasound for registration (cf. below in Section 2.3).

2.2.4 Navigation in Robot-Based Systems

In principle the avoidance of critical regions can be guaranteed when supported by some kind of mechanical apparatus guiding the tool. This may be a simple jig or template (as conventionally used for implant milling procedures, either with a negative template e.g. for hearing aids or a positive template e.g. for femoral hip endoprosthesis components), but also a passive or active robot system.

First however, it is important to differentiate between the aforementioned categories of "active/autonomous robots/systems" on the one hand and "teleoperated" as well as "compliant/synergistic systems" on the other hand. Where the first group needs to have some facility to plan a motion for the end-effector to perform a certain set of actions (e.g. path planning for milling, scanning, etc.), the second group can offload this task to the human operator and rely on his hand-eye coordination to perform motions.

Path planning can have many forms: It can involve the motion of a tool-equipped robot for orthopedic bone milling/cutting [Waringo03c], [Taylor94]/[Kazanzides95][1]; the positioning of a mirror to direct an osteotomy laser towards target points and its concomitant "packing" or "stacking problem" of laser-dot coverage [Kahrs09]; or trajectory planning for "active cannula" needles consisting of multiple pre-bent sheaths that allow flexible targeting during needle insertion under 3D ultrasound guidance [Boctor10]. While the latter two examples are "static" in the sense of their solutions being dependent only on their respective entry points and the target geometries, general path planning for robotic systems also includes a notion of optimality w.r.t. reachability (especially for minimally invasive interventions [Sugita08b]), trajectory energy minization (in particular for on-line trajectory computations [Kroeger10]), tool interaction parameters or patient anatomy (e.g. for collision avoidance in robotic sinus surgery [Li04], [Bumm05]).

Navigation with passive, compliant, or teleoperated systems very often reduces to first the definition of "no-fly zones", and second their intra-operative observation. A classic example of a passive robot is the PaDyC system limiting the hands-on robot motion to predefined regions by braking with clutching freewheels [Troccaz96], whereas the more general concept of *virtual fixtures* would allow both hands-on and teleoperated systems. A current example demonstrates the usefulness of such fixtures in the compliant cooperative milling in skull base surgery using a NeuroMate robot with force-torque sensing human input [Xia08].

2.2.5 Positioning

An important consideration when using any kind of mechanical structure in the OR environment is proper placement. Kinematic constraints both within a robot and between the robot and the patient (where optimal access may be difficult to achieve as well) make it difficult to define a universally applicable base position or access position for any given intervention. Instead, the following aspects have to be considered [Engel03]:

1. Feasibility of the robot path
2. No proximity to singularities
3. No collisions of robot and patient
4. No collisions of robot arm segments

[1] Of course, path planning is closely related to NC milling planning, which has been around for much longer than orthopedic surgical robotics, and correspondingly contains a large amount of prior art; cf. [Dragomatz97] for an exhausting review of about 220 publications up to 1996.

5. Consideration of surgical access path

Each of these may make plan execution impossible and cause plan abortion at any time. To locate a robot's optimal or at least acceptable position relative to the patient, the criteria above need to be considered under variation of the degrees of freedom of the mobile platform on which the robot is mounted (the *base*) as well as the degrees of freedom of the operating table or the patient. While for teleoperated systems such as the daVinci it is feasible to set an approximate position for the arms only based on experience, this becomes less attractive for autonomous systems that may require major reconfiguration procedures when running into limitations.

Optimal robot placement can be computed on the whole solution space, by iteratively subdividing the continuous parameter space based on interval analysis [Maillet04]. This allows integration of various optimization parameters such as reachability and dexterity, and generates valid positions or subspaces as a result.

Concentrating more on a pre-defined path for skull milling applications, another approach (the RobaCKa project, [Engel03]) iterates over the path points, ordering them according to a criticality metric, and performs a brute-force search on the relevant kinematic parameters (like position of the robot and the OR table) to find the best configuration. The surgical team is guided towards this configuration during OR setup with the help of optical tracking of robot and table.

However, it has to be noted that OR environments contain more objects and obstacles than just the robot and its dexterity – usually there is a patient, accessibility constraints (such as trocar port placement), or other objects such as multiple robot arms. Optimality criteria then include reachability, joint limits and singularities, collisions, manipulability and accuracy, dexterity, and robustness, some of which are necessary conditions, others can be fulfilled partially. One solution based on genetic algorithms for initial solution finding based on discontinuous constraints and gradient-based optimization for fine-tuning, and with plan transfers into the OR using localizers, robots, or projections is presented in [Konietschke08]. However, a distributed implementation of these algorithms on a computing cluster takes about one hour for completion.

2.2.6 Deformation

Tool deflection by deformation can have noticeable effects on serial manipulators. For simple setups, this can be approximated reasonably well by axis-independent linear models (for an overview cf. [NewRiverKinematics]; [Denis03] presents another similar, medical system).

Deformation of the robot has been discussed for only a few surgical robotic systems, including e.g. the sensor-aided serial robot system in [Engel02], where deformations of up to $1.5mm$ are reported for milling operations. However, this effect is interesting for high-precision autonomous robots like e.g. the CRIGOS hexapod orthopaedic system, for long, teleoperated arms like the ZEUS endoscopic system, and has been discussed for industrial systems, e.g. the nuclear inspection 4DOF (degrees-of-freedom) manipulator Cobra, where extensive modelling based on deflection matrices for individual links is performed, and accuracy could be improved from $1.5in$ to $0.09in$ under $30lbs$ load [NewRiverKinematics]. In another medical deformation modelling work [Drouet02], a patient positioner's accuracy was improved from $5...8mm$ to $0.4mm$ through representation of geometric and elastic errors for six axes and calibration with a laser tracking device. Freehand robot-less navigation with force-based deflection compensation is discussed e.g. in [delaFuente04]. Overall, however, deformation modelling is more important for autonomous systems, less so for cooperative, handheld, or endoscopic systems, as all of them are under direct visual control of the operator.

Deformations of tissues and other surgical environments are usually represented by finite element models and mainly used for training and simulation. They are of little interest here

because of their generally lower requirements for exact representation of actual behaviour.

2.2.7 Image-Guided Therapy, Control, and Communications Toolkits

For image-guided medical robotics, once the initial imaging is performed, the next step involves the computation of derived imaging modalities, followed by segmentation, region definition, robot or tool path planning, displaying the results, validation by the user, tracking, image guidance, and a variety of other process steps depending on the particular application under consideration. The development of such systems will naturally draw on existing drivers, components and interfaces, but will probably introduce large-complexity issues due to the amount of interacting parts. Real-time and image processing toolkits aim to reduce the time and development investment related to recurring modeling and interfacing tasks.

A considerable amount of such toolkits, frameworks, and component collections is populating the field of real-time and/or image guidance research and medical robotics system development. Apart from low-level hardware vendor APIs, several open-source projects have stepped up to this task.

One toolkit aiming at modular abstraction in image-guided therapy (IGT) in the widest sense deals with the problem at the networking level: OpenIGTLink [Tokuda09] defines a protocol for data transmission between communication parties and provides a reference implementation for TCP/IP sockets. Considered data types include images, positions, status messages etc. It should be considered not so much an IGT toolkit but rather an important enabling technology, as many other approaches are building on the interoperability that OpenIGTLink provides.

The IGSTK (Image-Guided Surgery Toolkit) [IGSTK] recognizes the overhead involved in re-inventing classes and functionality related image guided surgery and provides extensive support for aspects like hardware-independent tracking, volume representations and manipulations, file I/O (e.g. DICOM), networking (by means of OpenIGTLink), registration (on ITK), calibration, GUI building (on VTK and Qt), and many others. It does not, however, include support for robotics-related issues or real-time ultrasound imaging or any other non-tracking sensing modality.

The ITK (Insight Toolkit) [ITK] is very widely used and concentrates on image analysis, filtering, registration, and segmentation tasks, but has only very limited focus on real-time aspects, and none on sensor, ultrasound, or robot interfacing at all.

One development that has a similar goal as IGSTK are the CISST libraries [CISST] in conjunction with the SAW Surgical Assistant Workstation [SAW], incorporating support for robotics, algebra and operating system abstractions, vision, and intra-operative imaging. However, they should be viewed more as a technology carrier for IGT and minimally-invasive surgery where e.g. US imaging is a useful modality for imaging, i.e. they rely on external support to receive planning data.

Another interesting project is the Robot Operating System (ROS) [ROS], a real-time infrastructure toolkit providing architectural components for – in particular robotic – systems, the most important here being inter-process message passing in a publish/subscribe approach. Messages can be published in "topics" and are automatically distributed to subscribing clients, and parameter servers support e.g. central initialization handling.

Also aimed at the development of real-time robotic systems, the OROCOS (Open Robot Control Software) project [OROCOS] targets lower-level control aspects, such as component communication, component distribution, kinematics and dynamics, and data filtering.

Overall, it becomes clear that although many aspects of image-guided therapy and robotic system design are dealt with by one or the other existing toolkit, none actually solves the problem of providing a simple-to-use, hardware-independent research environment that allows

simple implementation of new IGT-related algorithms and the application-agnostic, flexible setup of robotic research systems. Furthermore, most of the described projects were just begun in the early to mid-2000's, so they were not available at the time of planning for the project presented in this work.

2.2.8 Ear-Nose-Throat and Head/Neck/Neurosurgery

Several surgical robotic systems already exist in the field of skull base, ear/nose/throat (*ENT*), or head/neck/neurosurgery. All of these systems are active and either autonomous or interactive ("hands-on", "synergistic"), based on serial or parallel hexapod kinematics. As a rough overview of fields tackled by medical robotics in the cranial area, the following list provides some orientation (adapted from [Widmann07], which also contains an exhaustive literature overview in image-guided surgery (*IGS*) and medical robotics in the cranial area):

- hole-drilling/trepanations with automatic stop after bone penetration,
- implant bed milling or bone fixtures drilling,
- bone surface shaping in plastic surgery,
- bone sawing in osteotomies/craniotomies
- bending of osteosynthesis plates and correct intra-operative placement,
- guidance for brachytherapy.

The RobaCKa system [Eggers05] is technically the closest relative to the system discussed in the present work, being derived from a CASPAR surgical robot as well, with the human surgeon supervising an autonomous robotic craniotomy. Straight cuts into the skull bone are performed by the robot using a surgical craniotome under optical tracking. Initial registration with planning CT data is performed with a pin-based approach with both optical and robot force-following steps. It incorporates various safety features that were independently developed in the presented work as well, but does not include significant intra-operative sensing, nor does it perform the whole bone-cutting procedure on its own, instead relying on the surgeon to perform the lower half-bone-width cut manually for safety reasons. The main benefit of the robot in this approach is the high geometric accuracy of the plan-to-OR transfer. In later work [Raczkowsky06] [Klenzner09a], an accuracy of $0.1mm$ that was not achievable with the RobaCKa system could be reached with a hexapod setup. Later work of the same group is aiming to replace the robotic milling steps with laser ablation [Kahrs10] with optical control using image processing [Klenzner09b], achieving micrometer precision ablation monitoring (cf. Section 2.3).

For milling in the sinus sphenoidalis area, another serial robot system "A91" (based on the 6-DoF Mitsubishi RV-1a) makes use of pre-operative CT planning data and intra-operative optical tracking [Bumm05]. An important technical feature is the endoscopic approach with an integrated milling/irrigation/suction tool and camera. Different control mechanisms were implemented, all incorporating collision control between the endoscopic tool and the narrow approach area through the nose; with resulting deviations of up to $1mm$, mostly due to the drilling process itself. A precursor system by the same group, using the same planning procedures and based on a URS Evolution 1 hexapod system is described in [Nimsky04], but was abandoned for a variety of reasons, including hardware availability, cumbersome setup, and achievable precision.

The CRANIO system based on a hexapod kinematic carries a surgical drill and carries out trepanations and craniectomy cuts with optical registration and tracking with a mean

precision of $0.24mm$ in synthetic phantoms [Engelhardt06] [Popovic03], but without significant intra-operative sensor support. An early medical hexapod system ("OTTO" based on a MSS/jojumarie SurgiScope robot, [Lueth98]) allows either autonomous robotic interventions based on pre-operative CT, or the robot's use in an "Intelligent-Medical-Instrument-Support" mode under surgeon teleoperation guidance. Application areas are rigid catheter insertion and bone fixtures for ear prostheses.

In a recent work, a robotic system for autonomous milling at the skull base with optical tracking [Baron10] on top of a KUKA KR3 robot was able to achieve $0.5mm$ accuracy in laboratory experiments, using both optical tracking and robot encoders to generate redundant position information.

In what can be described as a guidance approach, the MicroAssistant [Lenze06] positions a surgical instrument, e.g. for skull bone milling, with two DoF, leaving the axial motion under surgeon control. With a small cartesian-stage-type kinematic, trajectory errors on the order of $< 0.4mm$ could be achieved. In an earlier work [Li04], the underlying idea of of force-following under virtual fixtures was explored with the 7-DoF JHU Steady Hand Robot [Taylor99], resulting in errors of about $0.75mm$. For milling applications at the skull base, this was implemented on a NeuroMate robotic arm with StealthStation optical navigation support in [Xia08], resulting in typical overcuts of $1 - 2mm$.

Another small system, aimed at trajectory planning and execution for minimally invasive hearing aid implantation, in particular the linear drilling access to the inner ear, is described in [Caversaccio09], but explicitly mentions precision, stiffness, and segmentation of critical anatomical structures as existing obstacles to clinical use.

The only robotic system using intra-operative sensing data for contact state estimation and real-time process control aims at cochleostomies, one step in the implantation process for hearing aids [Brett07]. It bases this estimation on force-torque measurements, allowing to stop the milling feed in vertical direction just before breakthrough into the cochlea occurs.

Overall, it can be stated that most existing systems have concentrated on kinematics, control, and planning issues, with sensing constrained to pre-operative CT and/or human guidance and supervision, with only few systems incorporating advanced intra-operative sensing.

2.3 Sensors for Surgical Robotics

In CAS/RAS, robotic systems widely and almost exclusively employ global sensors for navigation (Figure 2.2). This includes global imaging sensors (CT, MRI, X-ray, and C-Arm-based derivative modalities such as cone-beam CT (*CBCT*), and to a lesser extent PET, SPECT, etc.) for preoperative planning and confirmation purposes, as well as position sensors (encoders, trackers) for intraoperative use. Such systems thus form a subset of image-guided surgery (*IGS*) applications, and comprise the vast majority of surgical robotic approaches.

Local Surface Sensing

The above-mentioned global imaging sensors all carry certain drawbacks – CT and X-ray expose the patient or the surgeon to ionizing radiation, MRI requires the OR equipment to be non-ferromagnetic, they have a large space footprint, require image registration, have limited resolution, are costly etc.

In this light, some research groups introduced intraoperative sensing capabilities that sidestep these problems, where the systems exploit data sampled with relatively cheap and small sensors that incur only negligible additional overhead or which can be closely integrated into the surgical system itself. Such sensors can be tentatively called *local sensors* here, with a precise definition following later in Section 4.1.1.

Closest to the imaging paradigm, one such system samples surface information during a bone milling procedure with a sweep of line laser scans which are assembled into a 2.5D surface representation of the region of intervention [Malthan04]. As the planned bone removal volume is known, the surface distances between the desired and the actual volumes can be computed and later be used for simple and fast progress monitoring and possibly further intervention planning (surface-based registration based on laser scanning, with partially defective surfaces due to specular reflections from blood or irrigation fluid, had a reported accuracy of $0.5mm$ [Malthan03]). Fluorescence measurement for tissue discrimination (bone, soft tissue) is another possibility explored there under early development [Woessner04].

Contact Sensing

In milling applications, the contact between miller and material provides a wealth of auditory, visual, haptic, tactile, and olfactory clues for state estimation and process control. However, this data receives comparatively little attention. One area of research is use of force samples e.g. for classification of surgeon skill based on force measurements at endoscopic grippers [Rosen01], where Hidden Markov Models prove useful for analyzing the low-bandwidth flow of data.

Another recent approach in medical sensing is the high-frequency analysis of intra-operative vibration and sound [Boesnach04], where microphones record audio samples during vertebra drilling which are classified as different bone contact states through neural networks, support vector machines, or Hidden Markov Models. To avoid damage to the spinal cord and adjacent blood vessels, information about areas of different bone densities and transitions between them that is independent of pre-operative planning data is provided by classification of data from both room and boundary layer microphones (at $96kHz$ sampling rate) affixed to the vertebrae under intervention. All three classifiers performed well enough (78...89% classification rates) and were real-time capable.

In a similar vein, laser osteotomy (laser bone ablation) generates sound emissions amenable to tissue classification methods as well. (For an exhaustive survey of laser ablation techniques and its history cf. [Kahrs09].) Without haptic feedback, it is even more important to subject process noise to analysis in order to determine ablation crater depth and tissue type. With short-pulse lasers, thermal necrosis can be minimized, and when sampling the explosive acoustic emissions from the tissue vaporization with e.g. piezoelectric (structure-borne noise) and piezo-ceramic (room noise) transducers [Raetzer05], it is possible to derive depth from time delay and tissue type (bone or soft tissue) from the wider bandwidth and higher energies at high-frequency spectral components ($100...200kHz$). In [Kahrs09], laser-based cochleostomies are supervised through camera-vision-based control with region segmentation to detect boundary layers in the cochlea wall.

The use of 3D force and sound samples as input for contact state classification and mapping in milling applications is described later in the present work (Section 4.3, [Stolka05]). Force/torque sensors can also operate in control cycles. Prominent examples are the control of feed rate in milling applications based on force measurements ([Engel02], [Federspil03], or the ROBODOC system) or tool deflection estimation (also [Engel02]). To prevent excessive forces during bone machining, a predictive scheme based on knowledge about the locations of hard tissues can avoid high-speed collisions with the milling tool, as e.g. in [Sugita09].

Another sensor (developed for the presented system) is a highly precise A-mode ultrasound scanner for thickness measurements [Federspil03c], which can be used for bone volume reconstructions in conjunction with tracking.

Situated somewhat between contact-based force sensing and contact-free audio sampling, a vibro-tactile approach allows tissue type classificiation under contact. Vibration sensors can be placed on handheld tools and serve to support the surgeon with real-time environment

Applications in CAS/RAS		**Global Navigation**	**Local Navigation / Control**
		• complete map	• incomplete / no map
Global Sensors wide range static / „single-shot" *in widespread use*	• Computed Tomography • Magnetic Resonance Imaging • Laser surface scanning • Optical tracking system (OTS) • Robot (encoders)	• Path planning, optimization on image data [Waringo03] • *CASPAR* • *ROBODOC* • *Acrobot* • *CRIGOS* • *RobaCKa* • ...	• Path correction (e.g. under *patient shift* [Ranftl05])
Local Sensors data from immediate neighbourhood dynamic *currently under research*	• F/T sensors • Audio • Electromyo-graphy • Ultrasonography • Laser fluorescence • (Video?)	• Map creation/ update [Stolka05] • Path replanning [Waringo04]	• Controlled milling [P-law control: Federspil03], [window control: Engel02] • Deformation estimation [Engel02], [Stolka06] • Contact classification (audio / vibration) [milling: Boesnach04, Stolka05], [laser: Peters05?] • Map creation (stochastically complete) [no references!]

Figure 2.2: State of the art in sensor usage (according to sensor type and navigational use) in the CAS/RAS fields. The use of local sensors for navigation (bold red frame) is still rare and not formalized.

information based on resonance excitation during milling interventions in the head/neck area, where bone and soft tissues are very closely located together [Plinkert98].

Ultrasound

The application of ultrasound sensors (*ultrasonography*) in medicine is very common. This includes initial diagnostic use (predominantly for soft-tissue imaging, e.g. pre-natal examinations, cardiologic and vascular diagnostics, and others; there is some space for ultrasonography even in orthopedic contexts) and interventional application (for non-invasive real-time monitoring and guidance, e.g. needle tracking for brachytherapy, prostate interventions and biopsies etc.).

However, the use of ultrasound has been traditionally limited to manual intervention planning, monitoring, and guidance purposes due to its low resolution and strong imaging artifacts. Where clinical ultrasound systems offer B-mode, Doppler, and sometimes elasticity imaging (EI) as standard imaging modalities, advanced ultrasound research – involving e.g. synthetic aperture imaging, adaptive beamforming, photoacoustic imaging, thermal imaging, high-performance elasticity imaging, but also high-precision distance or thickness measurements etc. – requires access to the underlying ultrasound radio-frequency (RF) data. Although some manufacturers' systems allow the sampling and storage of RF data through sometimes proprietary, sometimes open interfaces (e.g. Ultrasonix [Wilson06], Siemens [Brunke07], Hitachi [Shamdasani08], ZONARE [Mo08], or Philips/Agilent [Fabian01]), the resulting systems are relatively closed, and data is often saved into files in an asynchronous fashion, making access only offline and often requiring "monkey jobs" such as pressing buttons and scrolling during acquisition.

A few approaches are building on top of those machine-specific application programming interfaces (APIs) or toolkits to allow for IGT or ultrasound research development. SynchroGrab [Boisvert08] for example encapsulates the low-level interfacing with ultrasound (US) machines and tracking devices to provide tracked 3D US functionality. The resulting volume data is handed over via OpenIGTLink to compatible clients, e.g. 3D Slicer [Slicer3D] for visualization and intervention guidance. As SynchroGrab calls on VTK to provide a video data source, it can fairly easily be extended to support other machines than the ones it currently interfaces with (by Ultrasonix Inc., Burnaby, Canada), even using frame grabbing in case no open low-

level access is available. Although the developers correctly realized the problems monolithic vendor-proprietary solutions pose, SynchroGrab itself again provides one functionality, although admittedly with the ability to extend and build on the source code. Another approach is pursued by [Stolka10c], where algorithmic components interact with each other by means of network connections, allowing real-time access to machine-independent US data and its use in very flexible system setups, e.g. [Stolka10a].

In spite of ultrasound's drawbacks, its non-invasive nature makes it a strong candidate technology for registration, as e.g. with optically-tracked US B-mode probes to register the patient with a prior CT scan in orthopedic (spinal surgery) applications, e.g. [Winter02]. There, bone surfaces are extracted from a CT volume in a preprocessing step (which is clearly direction-dependent when related to US) and registered with 3D ultrasound in a surface-volume approach. This ultrasound/CT registration naturally also works in skull surgery, e.g. in [Maurer99] (and later [Amstutz03]). There an optically tracked US probe collects surface points on the skull and registers these to surfaces extracted from pre-operative CT with a mean target registration error (TRE) down to $1.0mm$, benefitting from the invariability of the bone surface under different conditions during imaging and operation. An example demonstrating the applicability of this approach in robot-based settings was also given by the CRANIO project [Popovic05]. Actual 3D ultrasound scanning for planning as well as registration can be found in only a small number of (orthopedics) projects, e.g. in the IR-navigated B-mode manual scanning of the shoulder/elbow area with concurrent 3D volume reconstruction in [Jan04].

Expensive pre-processing may not always be possible due to time constraints (such as in conjunction with intraoperative imaging). To generate more comparable volume representations of the region of interest, it may be necessary to switch from B-mode US to advanced modalities such as elasticity imaging (EI). In one such work within the context of laparoscopic navigation for partial nephrectomies, 3D-EI of the kidney is registered with CT that includes pre-operative planning information using volume-volume methods [Keil09], improving registration quality as compared with 3D-B-mode.

B-mode US can also be used not just for static registration, but also for real-time tracking, e.g. for the closed-loop control of a robot-based high-intensity focused ultrasound (HIFU) ablation device for kidney stone destruction, where the kidney surface is tracked with orthogonally arranged linear B-mode probes and surface-to-model registration [Seo09].

Recent developments towards high-precision US systems enable their use at least for situations where the thickness of tissues with highly different densities needs to be measured [Federspil05], [Federspil06], [Tretbar04b], with ultrasound-based tissue boundary detection on the skull calotte, as described later in Section 4.3.3.

Beyond the applications discussed above, there exists a plethora of emerging ultrasound technologies for monitoring and therapy (e.g. elastography, shear wave imaging, new matrix-array probes, thermometry, HIFU), all of which bear some relation to the application at hand, but shall not be investigated further in the present work where the focus is less on imaging but more on navigation on imaging sensors.

2.4 Conclusion

The key fields still presenting problems or technical barriers related to the use of medical robots and, more generally, computer-integrated surgical systems can be summarized as follows [Taylor03] [Dario03]:

1. Imaging, Modelling, Analysis

2. Intraoperative Registration

3. Haptic Feedback
4. Systems Integration and Modularization
5. Human Interfaces
6. Initial System Cost

In general, optical tracking offers a good compromise of precision and speed for both static and dynamic registration in many applications such as orthopedic, neurosurgery, and ear-nose-throat (ENT) interventions (all of these benefit from a rigid anatomy and more or less direct accessibility which allows line-of-sight tracking and good marker fixation). However, beyond tracking and several ultrasound and radiological imaging methods (with all their respective drawbacks), current robot-based surgical systems have no means of dynamically updating their navigation data to improve precision, react to changing environmental conditions, or build maps from the ground up.

Intra-operative map-building is considered only insofar as tracked US or laser scans are used for reference purposes or registration control. No approaches seem to exist which aim at persistent use of and navigation based on local sensor data in orthopedic applications. However, both the classification of sound or vibration data for the determination of milling state in handheld applications and, unrelatedly, the incremental building of a tactile map of an intervention area have been shown to be feasible. Thus, the use of local sensors and local-sensor-based navigation (as defined in [Henrich04] and to be extended later in the present work) promises a benefit yet unexplored in current systems.

Also, regarding the achievable overall accuracy of milling operations, more information is necessary about first the influence of registration based on externally tracked pointers, and second about milling deformation effects, to determine the precision of robot-assisted surgical milling processes. In the presented bone milling application with the RONAF system [Henrich02], tool forces of up to $30N$ suggest the need for deformation modelling.

Returning to the above-mentioned list of technical barriers, the present work will touch upon points $1-5$ in some way or other: by proposing map-building procedures based on local sensors, by solving the intra-operative registration problem by rendering it unnecessary, by providing force feedback to the robot process control, by integrating all system components into a modular system easily adaptable to different interventional workflows, and by providing an ergonomic graphical user interface that guides the user through the whole procedure while intercepting errors or omissions.

Chapter 3

Proposed Approach

In this chapter, the basic approach of this work will be outlined. The main overall contribution is the introduction of local sensors into global and local navigation cycles in RAS. These and related terms will be put into context in this chapter.

A definition of the underlying concepts in Section 3.1 clarifies the meaning of both existing as well as of newly introduced terms. Then, general navigation principles which can be identified in most or all CAS/RAS setups – and thus also in the presented system – are presented in Section 3.2 (cf. [Henrich04]). With this background, the basic work hypotheses to be investigated can then be discussed in detail (Section 3.3).

Finally, one particular instantiation of the proposed approach will be described – the system presented in this work. From a process view of the medical intervention as detailed before, all intervention process phases will be explained (Section 3.4). This covers the full spectrum of user-system interaction, i.e. from registration and imaging up to adaptive path execution. Furthermore, the presented approach allows several different system configurations, all showing different aspects of the investigated scientific contribution. These system configuration options are described in the last section of this chapter as well.[1]

3.1 Definitions

This chapter recapitulates definitions for the basic concepts used in the presented work, especially in the context of computer-assisted surgery (*CAS*) and robot-assisted surgery (*RAS*). The differences between new definitions arising within the scope of this work and existing ones are especially highlighted.

3.1.1 Sensor Classes

Sensors are entities of a system (consisting of hard- and software) which provide data from a suitably defined environment. Sufficiently reliable and accurate sensors can thus be used as information sources for map-building and control algorithms. In the context of this work, we define sensors as system components able to provide a feature vector C of the physical external environment or of the internal system state.

Useful differentiation of sensor classes includes the distinction between *internal sensors* (which measure the internal state of the system and whose readings are independent of the

[1] This work implicitly shows a *development perspective* as well, which describes the complete process of investigation and system integration. Its phases range from the initial requirements evaluation all through to the final robot-assisted system. While not explicitly covered in a separate section, these aspects are contained in each of the main chapters. The system itself is systematically built in consecutive steps from local sensors, maps from these sensors, and navigation algorithms.

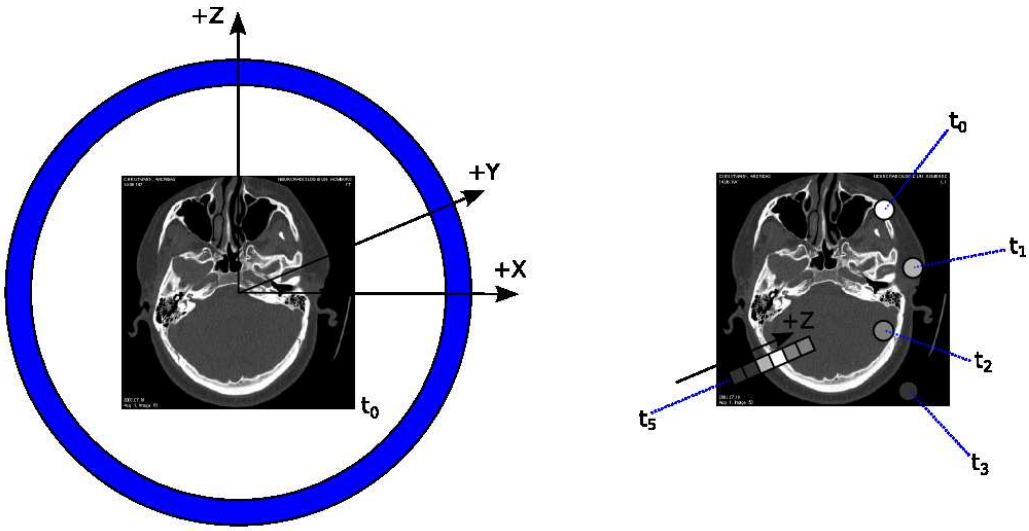


Figure 3.1: Schematic representation of conceptual differences between global (left, blue circle) and local (right, single points) sensors. Global sensors deliver a complete view of the ROI (in the center) in one shot at time t_0, while local sensors can only sense disjoint parts of the ROI at different times t_i.

environment) and *external sensors* (which are essentially independent of the system), and the distinction between *active sensors* (which emit energy of some form into the environment to be measured and sample the response) and *passive sensors* (which rely on the energy dissipated by the environment without additional stimulation).

The distinction between *physical sensors* (which are physical components providing actual readings of environment features based on a physical sensing principle) and *logical sensors* (which are integrated components consisting of underlying physical sensors and subsequent sensor data pre-processing and provide higher-level, interpreted environment information) will be important. As stated in the delimitation of this work (Section 1.5), the focus lies not on the development of new sensing principles and sensors, but on using existing physical sensors and extending them with sensor data pre-processing and classification procedures.

All of these sensors can provide analog (continuous) as well as digital (discrete) data, in terms of time, space, and quantization (within the limitations of the discussed electronic-digital systems).

The differences between global and local sensors will be elaborated in the following (cf. Figure 3.1). Given an environment, what is sought as a criterion to define "local sensors" is the environment's representation as returned by the sensor. The criterion will be based on the amount and qualities of returned sensor data.

3.1.1.1 Global Sensors

In the scope of this work, *global sensors* shall be entities providing data about the whole sensor environment or about the sensor position relative to this environment in one measurement, i.e. through a (from a systems perspective) atomic sampling step.

The data is embedded in a special associated coordinate system of the sensor. Furthermore, global sensors usually provide long-range data in the sense that the location of the sensor and the origin of the data are decoupled, i.e. not necessarily collocated – nevertheless, the transformation between data elements and sensor origin as well as the world-coordinate origin is generally well-known. The nature of global sensors sometimes limits them to providing static data, i.e. their space, time, or technical requirements do not allow data acquisition concurrent with the process.

Global sensors are widely popular, however, as most of the currently used sensor modalities

exhibit the properties described above. Examples from the CAS/RAS fields are (cf. Figure 2.2):

- *Computed Tomography* (CT, also Computed Axial Tomography / CAT)
- *Magnetic Resonance Imaging* (MRI, also called Magnetic Resonance Tomography / MRT, Nuclear Magnetic Resonance / NMR, and others)
- *Laser scans*
- *Optical cameras*
- *Optical Tracking Systems* (OTS, sometimes also called navigation systems)
- *Robot joint encoders*

3.1.1.2 Local Sensors

Local sensors shall be understood as such entities providing data about the immediate neighborhood of the sensor itself (the extent of the "neighborhood" needs to be defined appropriately for each sensor type) [Stolka05]. The data thus carries no information about its global location of origin, which may however be attached in a subsequent localization step and may be necessary for navigation purposes. Data from local sensors can be sampled unobtrusively in a continuous or concurrent fashion, therefore their data is of dynamic nature, at least in dynamic processes.

Local sensors are only recently being investigated in the context of CAS/RAS and include, but are not limited to, the following modalities:

- *Force/Torque sensors* (*F/T*, including vibration sensors)
- *Audio sensors* (for both air- and structure-borne noise)
- *Optical sensors* like *laser fluorescence sensors, optical coherence tomography* (OCT) etc.
- *Electrophysiological sensors* (including electromyography (*EMG*), electrocardiography (*ECG*), electroencephalography (*EEG*), and others)
- *Ultrasound* (*US*, also echography or (ultra-)sonography)

For a more detailed definition and specific properties of local sensors, see Section 4.1.

3.1.2 Mapping Process

Also known as charting, the mapping process involves a local sensor (moved by e.g. a robot, or more generally an agent) located at varying poses in the environment to be mapped, recording sensor data into a map. This environment is multidimensional and may be composed of multiple aspects – detectable by different sensors – which may or may not be relevant to navigation.

3.1.2.1 Map

A *map* is defined as a function or mapping from the environment (more precisely, its features) into a possibly lower-dimensional representation, accompanied by access functions used by navigation algorithms. Semantics of the map are application-dependent; however, a basic requirement is that the mapping between reality and map be "faithful", i.e. a certain application-dependent degree of similarity between real and represented features is required. This similarity may take any form as long as a bijection of represented features (at least restrained to the

already mapped ones) and environment features exists. This requirement may be relaxed insofar as a strict 1:1 mapping of map and environment features is not necessary, as long as application constraints (collision avoidance, memory requirements, range etc.) are satisfied.

The representation stores environment information in an associated coordinate system and provides access functions to retrieve this data, usually parametrized by location, but possibly by time or other dimensions. The mapping need be neither surjective (i.e. the resulting map need not be complete) nor total (i.e. parts of the environment may remain unmapped). In particular, a map M may store this environment information in an associated coordinate system, and provides data entry functions

$$b_M(C,\ t,\ p)\ =\ M' \tag{3.1}$$

(which return a modified map M') and query functions

$$q_M(p,\ t)\ =\ C(p,\ t) \tag{3.2}$$

with $p \in \mathbb{R}^n$ and $t \in \mathbb{R}$ being positional and temporal constraints on the map interaction, and C being a relevant feature vector.

Requirements as to the representational power of the map are formulated in terms of the application, which determines which environmental features should be represented (cf. Section 4.2.1) and how the map needs to be accessible (cf. Sections 5.2.4, 5.2.5, and 5.3.1). In case the map stores multiple aspects of the represented environment, sensor data fusion may take place in the map as well[2].

3.1.2.2 Localization

Furthermore, a robotic system needs to be able to *localize* itself or the sensor with respect to some coordinate system by determining a coordinate vector $p \in \mathbb{R}^n$ of its own location, which is performed by measuring or receiving environment features supporting that task.

For both writing into and reading from a map, localization of the sensors and/or the robot with respect to the environment is essential. Assuming a wrong sensor location during writing the map may prove disastrous for the map, which then becomes successively overwritten with misplaced data and thus fails to conform to the "faithfulness requirement". Similarly map-based navigation fails if it relies on map readings from estimated locations which do not correspond to the actual position. For CAS/RAS, localization is generally achieved by tracking tool positions by some kind of realtime localizer which has been previously registered with the environment and the robot or tool.

3.1.2.3 Registration

For any two sensor modalities, or a robot or tool and a sensor modality, or two robots or tools, to be interpreted or to work jointly together in one workspace, features defined in the respective coordinate systems need to be mapped onto each other. The determination of this transformation is called *registration*, and may be performed by different methods, like point- or feature-based mapping and subsequent rigid interpolation, or surface- or volume-based mapping with possibly non-rigid interpolation, or combinations as well.

Conceptually, registration allows to identify pairs of corresponding locations in two data sets. For surgical robotics applications, an intuitive way to provide this identification is to use the robot as a localizing device, pointing at distinct features clearly distinguishable in both the patient's anatomy and the existing data set to be registered with the robot, and computing

[2]This could mean storing e.g. both force/torque and audio sensor information in the map. This has not been explored in the present work, as emphasis was placed on the theoretically achievable accuracy of the proposed mapping approach.

the rigid 6-*DoF* (degrees-of-freedom) transformation between the two 3D point clouds (e.g. as in [Arun87]). Another option is to use imaging modalities or external tracking devices (like e.g. the NDI Polaris optical tracker) which determine the relative positions of both the patient and the robot together (co-registration).

In the RONAF system presented in this work, registration between milling path and the patient is equivalent to location planning for the implant bed. For generic implant bed milling paths, this has so far been performed by pointing at the origin and axes of the implant coordinates with the robot itself, using a force-following scheme (Hybrid N/P Control, [Stolka03]). Note that this scheme does not require a global map. Gathering a 3D ultrasound map directly with the robot is possible as well. Since the ultrasound sensor is rigidly attached to the robot, this map is implicitly registered and can be surface-matched with e.g. preoperative CT scans.

3.1.2.4 Exploration and Map Building

Exploration is covering the environment with possible sampling of sensor data. Coverage may be complete (as in sweeping algorithms), stochastically/asymptotically complete (as in random walks), or incomplete (as in directed traversals).

When exploration of the environment with a mobile agent or a sensor probe is combined with actual sampling of environment features and reliable localization, a map can be built concurrently. This *map-building* closes the loop of prerequisites for navigation – sensor data acquisition, localization, and entering the data into a map which can be read and interpreted later on as a faithful representation of an environment in which further motion planning and execution can be performed.

3.1.3 Planning and Navigation

Finally, if a map exists and the robot system is able to perform localization with respect to the map, it can not only reason with that information and use it for planning, but also execute motion plans and thus engage in navigation.

Planning on a map (in the sense of path planning) involves the generation of a *robot path* $\mathbf{P} = \{p_i\}, p_i \in \mathbb{R}^n$ as a "geometric representation of the desired robot motion" [Kroeger10], by taking into account constraints like start/end positions p_0 and p_{end}, robot kinematics, path requirements (e.g. full coverage, shortest/fastest/collision-free path, random walk, path abortion ability, vertical miller feed, etc.), and map entries $M(p)$ or $M(p,t)$ of its static or dynamic environment.

Execution of this path necessarily unfolds in time, resulting in motion between path points at certain speeds, and associated definition or generation of *trajectory* variables $p(t)$, $\dot{p}(t)$, and $\ddot{p}(t)$ for the motion of the end-effector and $q_d(t)$, $\dot{q_d}(t)$, and $\ddot{q_d}(t)$ for the motion of each robot joint d.

Depending on map availabilty, sensor range, and environment dynamics, it may be necessary to discard or ignore the map representation of far-away environment regions, instead focusing on those regions "near" (according to some suitable application-dependent criterion) the robot or its end-effector. Navigation on complete maps ("global navigation") and navigation on partial maps, local maps, or maps from local sensors ("local navigation") will be defined in the next section, and their difference will be relevant throughout this work.

3.2 Navigation Cycles in CAS/RAS

Surgical robot systems mainly rely on two sources of navigation information: *global data* sampled during a planning phase before process execution (preparation phase), which is then used

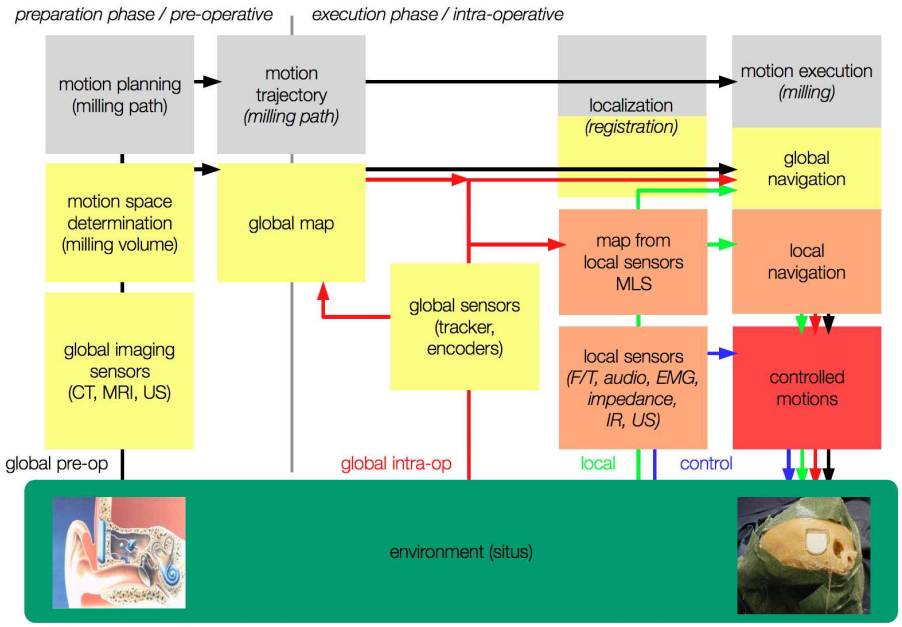


Figure 3.2: General navigation cycle architecture including four major navigation principles (black/-global pre-operative navigation, red/global intra-operative navigation, green/local navigation, blue/control) (modified from [Henrich04])

statically in a spatial context for global navigation, and *local data* sampled during the process (execution phase), which is fed back and used in a non-spatial context in open- or closed-loop process control.

The former may have insufficient spatial resolution, be difficult to segment, or expensive, while the latter only persists during the instant of sampling and is discarded immediately after entering into the control cycles. However, especially with autonomous robots, an additional class of information – intraoperative, spatial, current, and persistent sensor data – may be necessary to cope with uncertainty, measurement errors, and incompleteness of data. This local sensor data combined with position information makes up the basis for the construction of maps from local sensors, which form the main contribution of the presented work. This contribution is represented in a separate "local navigation" cycle, as described below.

The following sections describe the four idealized, general navigation principles in medical robotics which can be identified in a multitude of systems (Figure 3.2). This classification is based on [Henrich04], but was extended to accomodate the reality of various intra-operative uses of global and local sensors and maps.

3.2.1 Global Pre-Operative Navigation

Global navigation with a preoperative map (cycle "A" in Figure 3.2, black) requires preoperative imaging of the intervention region and is used mainly for planning. Locations and paths can be described within this map in a global fashion. This map needs to be registered with the environment before process execution. No strict temporal restrictions between data sampling and process execution are imposed. Global, preoperative sensors include computed tomography (*CT*; Figure 3.3), ultrasound (*US*), and magnetic resonance tomography (*MRT*).

The main advantage of global, pre-operative navigation is that it allows the planning of interventions ahead of time, not subject to the time and space pressures of the operating room, and thus allows to make guarantees about results and safety, which are verifiable by the surgeon beforehand. Also, it benefits from having a complete map at its disposal for planning. Its drawback, however, is the static nature of the planning data, which is of necessity stale at

the time of intervention.

In the presented system, this cycle contains – apart from global imaging – also motion space determination, i.e. the computation of the actual milling volume as the intersection of bone and implant volumes after implant position optimization [Waringo03c], and motion planning, i.e. computation of the milling path for a given intersection volume. Both data structures – the global map and the motion trajectory – are then transferred into the execution phase *en bloc*.

3.2.2 Global Intra-Operative Navigation

Global navigation based on an intraoperatively acquired map (cycle "B", red) is conceptually similar to the previous principle. One has knowledge of the complete environment either via a standard global map (from global sensors), a map from local sensors, or some non-imaging global sensors that provide positions. Acquisition may take place occasionally during the intervention, thus updating the environment representation. Furthermore, computations based on combinations of imaging data, motion trajectories, and additional intra-operative data are now possible. Here as well, it may be beneficial that the complete environment has been mapped.

External trackers (optical, electromagnetic, and others) and robot or mechanical localizer encoders provide current position data for tools, patient, or robot. This is useful to transform between image data and patient coordinate spaces, and to navigate relative to both. Appropriate registration is key to establish the geometrical transformations between the three registration entities "image data", "patient", and "tools/robot". These global positioning sensors can build a global map on their own (e.g. when sampling organ surfaces mechanically), can help to build maps from local sensors (e.g. tracked 3D ultrasound), and help to register motion trajectories to the patient space (e.g. through three-point registration).

Global intra-operative navigation can also encompass motion trajectory post-processing steps. A reachability analysis – i.e. checking the pre-computed trajectory against kinematic constraints and performing collision checking – can only be carried out after the whole path has been localized in the robot workspace through one of the mentioned registration procedures.

These intra-operatively acquired maps are much "closer" to the intervention than pre-operative data in terms of location and development of features, which is their main advantage. However, it may be cumbersome to acquire this data, or the data may be limited, depending on the desired modality.

In the presented system, three different groups of procedures fit in here: Navigation on intra-operatively acquired 3D ultrasound maps (Figure 3.4), intra-operative registration, and some trajectory post-processing steps.

3.2.3 Local Navigation

In local navigation (cycle "C", green), execution begins without prior knowledge, and an initially empty local map is continuously filled with information sampled during execution (Figure 3.2.4). This new information has two important properties: it is local, and it may provide more current and precise knowledge of the environment than global sensors could. However, for correct mapping it is extremely important to have precise and current location information. This needs to be provided by global positioning sensors. In essence, local navigation does not rely on the completeness of maps at the time of using them. Although in principle local navigation can be used with global maps, its main advantage is the timeliness of planning input, which is the reason why partial maps (from local sensors) are useful here.

Local navigation as presented later sidesteps the problems associated with explicit registration procedures for the establishment of registration between navigation data and actual situation (which global maps are subject to). However, the flip side of this advantage is that the

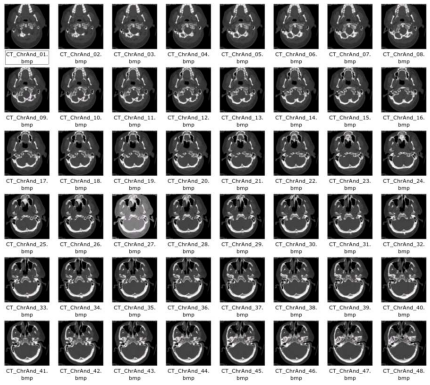


Figure 3.3: Global navigation on pre-operative data: Axial slices of a skull CT (data courtesy of Dr. med. Ph. Federspil, Heidelberg)

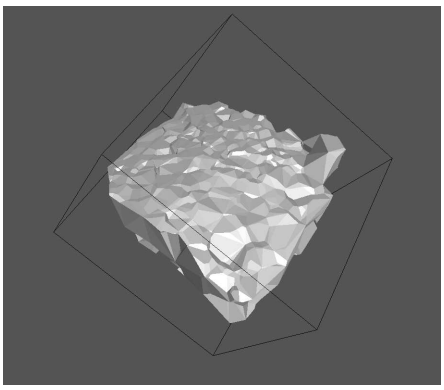


Figure 3.4: Global navigation on intra-operative data: Robot-based 3D ultrasound skull profile [Stolka07a]

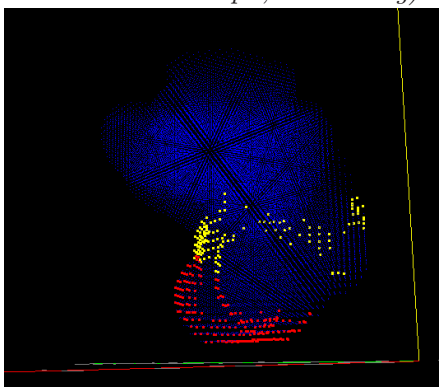


Figure 3.5: Local Navigation: Incomplete, concurrently-built map from local sensors [Stolka07b]

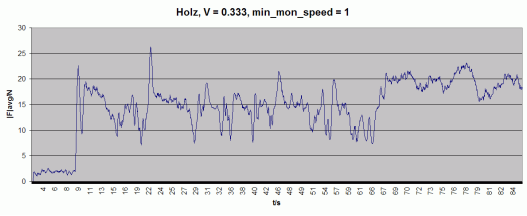


Figure 3.6: Control: Force-based milling speed control, target value $F_{target} = 15N$ while milling in wooden sample [Stolka01]

data underlying local navigation only describes a subset of the relevant environment (in terms of extent and possibly resolution), which limits the set of procedures which can be employed here.

Local sensors appropriate for surgical applications can be force/torque sensors (F/T), electromyography (EMG), electrical impedance, ultrasound, or infrared (IR) sensors.

3.2.4 Control

Control (cycle "D", blue) encompasses the sampling of data elements from the process – using the same intraoperative sensors as cycles B and C –, computing a reaction that is forwarded to an actuating element, and possibly providing a feedback path to the controller for closed-loop control. Control does not require spatial information; it serves as a reactive navigation principle without persistent map-building functionality.

An important positive aspect of control procedures is their inherent real-time property. The control input data immediately influences the system's behavior without necessitating lengthy planning procedures. However, this also implies that this data's value is very limited in time, namely only in the instant of sampling and control.

In the presented system, a control loop is realized e.g. through force-based speed control of the milling process (Figure 3.2.4).

3.3 Hypotheses

In this section, the main hypotheses of this work are laid out together with their accompanying experimental proof designs. Each hypothesis is analyzed in terms of its claim and scope, followed by an outline of the experiments designed to prove and support the claim, and a discussion of the expected gain of the experiments and their limitations.

At the most fundamental level, the contribution of this work can be found in the introduction and demonstration of local sensors and of the navigation cycle based upon these sensors. Therefore, the main hypothesis brought forth can be summarized as:

> *"Navigation on local sensors is a sound and complete set of procedures to enable surgical robotic systems to perform certain interventions."*

However, *soundness* (the system produces or infers only valid propositions) and *completeness* (all valid propositions can be inferred by the system) are very difficult properties to verify for an actual embedded system such as the presented one.[3] Thus, the proposed main hypothesis needs to be broken down into single statements which can be validated on their own. These statements are represented by the following (sub-)hypotheses.

It is worth noting that these hypotheses describe a general strategy for the use of local sensors, not just the special development of several specific, single sensors. Therefore, for other applications, new local sensors, mapping and navigation strategies can be developed and introduced in the same way as described below.

3.3.1 *H1:* Local Sensors Providing Surgically Relevant Information

One premise of building a surgical robotics system based on local sensors is developing such sensors which are applicable in surgical interventions (Figure 3.7). Given the "natural" or necessary behavior of the actuators or tools, and given the resulting system response, what is sought is a non-invasive way to derive system information from this response.

First, the relevant range of desired measurement types needs to be identified. In the present work, this includes mostly *contact states* and *distances*. Here, the concept of *contact* applies to the physical interaction between the tool and its environment, which are predominantly the miller tool and the patient intervention region. The main relevant states are *Tool-Air Contact* (C_0), *Tool-Bone Contact* (C_B), *Tool-Dura Contact* (C_D), and *Tool Resonance* (also known as "tool chatter"/C_R). *Distance* measurements are performed during thickness sampling of the skull bone. These are needed to create maps of sufficient precision to serve as input for planning algorithms later.

Then, appropriate sensors need to be developed (or more specifically, logical sensors constructed on top of physical ones). Finally, the sensors' performance needs to be characterized with respect to the investigated application. These important initial steps are covered by this hypothesis.

3.3.1.1 Claim

> *"Local sensors provide information relevant to the investigated application in surgical robotics."*

[3] For the described surgical robotic system: *Soundness* would be the system's ability to detect, describe (in terms of location and state), and react to only actually present environment features (*"everything the system sees is actually present"*); *completeness* would be the system's ability to guarantee the above-mentioned property under all circumstances (*"everything present can actually be seen"*).

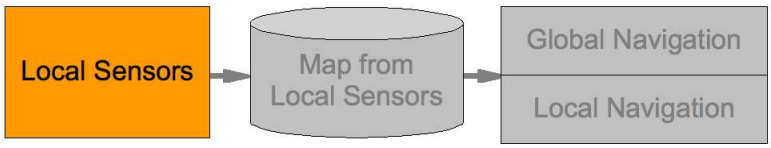


Figure 3.7: First phase of proposed local sensing-based navigation cycle: Local sensors converting raw intervention data into system state information.

Since this general claim is too broad to be derived from the definition of local sensors alone, this work concentrates on investigating several specific logical sensors, in particular some that are built on top of force/torque (F/T) sensors, air-borne sound (room) microphones, and ultrasound probes. Therefore, an appropriate specialized instantiation of the above claim for the investigated application area of this work – robot-assisted milling at the lateral skull base – can be formulated as:

> *"There exist local sensors, applicable for surgical milling interventions, which satisfy the evaluation criteria for sensors [as formulated below]."*

3.3.1.2 Experiment Design

Loosely based on the criteria used in Table 1.2, the selected local sensors will be investigated for their usefulness in surgical robotics. The following evaluation criteria are relevant in this context:

- *Discriminatory Power, Reliability*

  For local sensing during milling in the investigated application, it is important to differentiate between the relevant system states described before. Sensor data (both audio and force data) will be sampled from closely controlled situations (milling in the air, bone, and on the dura) for classifier training purposes. These classifiers will then be tested on a separate set of data to generate the confusion matrices (correct/incorrect classifications per class). For ultrasound, this criterion includes estimating the reliability of bone thickness measurements.

- *Precision*

  This criterion describes how well the sensor data can be localized in space around the sensor, i.e. what significance the position information for the data returned by the sensor carries[4]. This includes consideration of the spatial layout of active and passive regions on the tools, or the accuracy of ultrasound-deduced thickness values.

- *Timeliness*

  For online sampling, i.e. measurements taken during the execution of the primary task, this refers to sensor latency (the time duration between the occurrence of an environment feature and it being reported by the sensor). This also refers to the sensors' ability to provide data concurrently with the primary task (i.e. unobtrusively, without separate sampling phases, as opposed to "offline" or non-concurrent situations, where the primary task is interrupted or not active at all during the taking of measurements).

[4]Note that the local sensors' precision does not describe the number of significant digits of any absolute position information that may be attached to the sensor readings in later steps (by means of robots encoders or tracking devices), e.g. to build a spatial map. Instead, spatial precision of the local sensor refers to how much information is available to localize its data in its own local coordinate system.

- *Speed*

  The sensor data rate determines the later maximum achievable temporal resolution. This can simply be measured in the actual application setup and determines the spatial resolution by linking it to the robot motion speed.

- *Effort/Cost*

  As a higher-level criterion, this expresses the effect of the sensor application on the actual intervention (especially by slowing the process down). This is reflected in any sensor's synchronous (i.e. concurrent) or asynchronous application, which translates into online or offline sampling w.r.t. the primary task.

- *Burden/Stress/Exposure*

  Of equally high importance for the patient, and thus for the system design, is the burden, stress, or exposure to harmful influences it forces onto the patient and operator. Being more of a qualitative measure, these will be mentioned only in passing, although these criteria of course influence the initial choice of sensors to explore.

This validation encompasses a multitude of separate experiments which are described in more detail in Chapter 4.

3.3.1.3 Expected Gain

Based on experience with real-world interventions and experts' opinions, the intervention-specific relevant contact states and environment features are expected to be detectable and distinguishable with the proposed set of sensors. These can then serve as *direct input for control components* as well as for higher navigation cycles, enabling *real-time feedback* beyond the simple force-based speed control prevalent in state-of-the-art systems.

Furthermore, for future research, they can be expected to be mountable onto hand-held tools[5], thus extending their application range – beyond robot- or manipulator-based interventions – to manual interventions.

3.3.2 *H2:* Map-Building with Local Sensors

Goal-oriented trajectory planning (or navigation) requires not only spurious sensor readings, but persistent maps to store an environment representation (Figure 3.8). This map must be accompanied with data entry and query functions that allow fast and accurate mapping between real environment features and map entries. Furthermore, local sensors by definition provide only highly (spatially) focused data, so maps built from local sensor data need to be filled successively and/or concurrently with actual task execution, since localization of the sensor data can be provided only online. These maps from local sensors differ qualitatively from global maps from global sensors, as they cover only regions of the environment which have already been visited, but provide the most current information available at the time of visiting. Such a map can serve both as an intra-operative guide for safe trajectory replanning and as a post-operative reference for plan completion (although the latter is not investigated in this work).

[5]This applies to non-strain-based sensors (i.e. those not requiring a steady-state zero-frequency force signal component, as e.g. vibration and sound sensors in contrast to force/torque sensors, and also ultrasound).

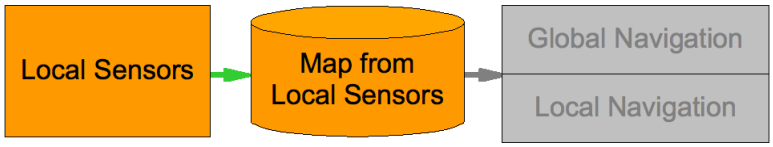


Figure 3.8: Second phase of proposed local sensing-based navigation cycle: Local sensor data being entered into a map from local sensors.

3.3.2.1 Claim

> *"Maps can be built concurrently with local sensors and additional localizers."*

This claim will be substantiated in three ways – theoretical analysis, practical considerations, and an experimental prototype.

The definition of "map" will be analyzed and its components – environment features, points and regions, coordinate systems, access functions, and achievable precision – discussed separately. Practical considerations involve estimations of the time and space complexity of the map-building process, based on specific application parameters, thus showing the feasibility of the task. Finally, a constructive proof will be given by showing that such maps can be built in an actual real-time application, and by comparing them with ground-truth knowledge of the environment.

Again, a more specific formulation of the above claim for the area of investigation is:

> *"Faithful and current maps of relevant features in the intervention area of surgical milling interventions can be built concurrently with certain local sensors and additional localizers."*

3.3.2.2 Experiment Design

With the local sensors investigated in this work falling into two categories (tool-based and not tool-based, cf. Section 4.1.2), two different kinds of maps will be investigated (concurrently-built and independently-built, cf. Section 5.1), which require different approaches to their validation.

Simulations will be used to determine good parametrizations of the concurrent map-building algorithms for tool-based sensors (force/torque and audio, both of which perform destructive sensing, thus irreversibly modifying the environment during the experiments) to determine best-case precision of the map-building algorithms. These will then be used in actual dummy and preparations (synthetic and physiological phantoms) experiments. For non-destructive local sensing (using ultrasound), validation can be performed in preparations experiments immediately, building 3D ultrasound maps of physiological (skull) phantoms.

Localization of the data is an important factor for the usefulness of the created maps. This is reflected in deformation experiments and different approaches to registration of pre- and intra-operatively generated data sets, all of which will be investigated with respect to their achievable precision.

3.3.2.3 Expected Gain

The resulting maps should be *characterized in terms of precision and correctness* of its entries, which is not only depending on the incoming correct sensor information, but also on correct integration of the new information into the existing map. Furthermore, it is important to determine the time and space *complexity of concurrent map-building* with local sensors to make sure it is useful in a real-time context.

Altogether, this will provide a foundation which allows the use of a new, complementary source of image data.

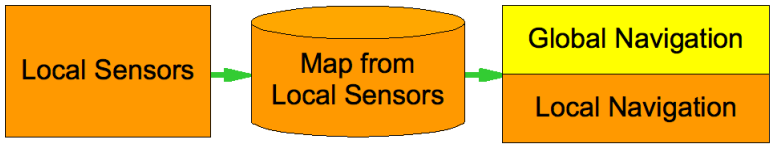


Figure 3.9: Third phase of proposed local sensing-based navigation cycle: Spatially localized sensor data being used to compute motion changes.

3.3.3 *H3:* Navigation on Maps from Local Sensors

Navigation consists of (self-)localization in a map, goal-directed motion planning and motion execution (Figure 3.9). This includes constraints on the planned motion – like obstacles, forbidden regions, and safety – and criteria like motion length and speed. While the problem of complete motion planning is actually outside of the scope of this work, navigation input consists of a map which should be a faithful representation of the environment to allow for relevant re-planning of existing paths.

3.3.3.1 Claim

> *"The current maps from local sensors can be used for navigation purposes."*

Here, the used map needs to be faithful and precise enough to allow motion planning that satisfies the constraints. Specifically, in the proposed system, localization is provided by global sensors (localizers like encoders and tracking systems), and safe motion execution can be assumed, since other measures – inside the robot controller – guarantee exact or "precise enough" path following. Therefore, this hypothesis mainly asserts existence of an environment map which allows safe motion planning, i.e. guaranteed avoidance of forbidden regions:

> *"The current maps, created according to the procedure laid out in the previous hypothesis, are valid input for the used path planning algorithms, and guarantee safe avoidance of forbidden regions under these algorithms during planning and execution with a robot-based system."*

3.3.3.2 Experiment Design

As the claim of this hypothesis focuses on the added benefit that can be gained from using maps from local sensors, a sensible experiment should be designed so as to highlight the effect of navigation-induced planning vs. no or only manual planning. Since initial path planning does not constitute a main objective of the presented work, it is reasonable to use a pre-planned milling path as a starting point, and to only apply planning modifications like position optimization. This will be shown in an experiment where first a pre-operative independent map from local sensors will be built using ultrasound, and then used to plan an optimal position for the milling path.

To show the effect of intra-operative re-planning, an experimental setup will be used which starts out with a given standard milling path execution on an anatomical preparation and then concurrently builds a map from local sensors. The system then changes that path autonomously upon encountering critical regions. To ensure it is possible to make exact statements about the precision and efficacy of the proposed approach, this experiment will be executed as a simulation.

3.3.3.3 Expected Gain

The overall advantages of performing navigation on maps from local sensors can be summarized as *increased safety* and a greater *variety of intervention approaches* due to a larger set of possible planning inputs. With the additional sensor data pathway opened up through local sensors, global data is updated, amended, or created during the intervention, which in turn results in increased safety when the navigation procedures account for the new knowledge about critical regions. In particular, *interventions without global preoperative data* become possible.

Overall, safe navigation on maps from local sensors will be a substantial step forward in extending the range of available intervention options in CAS/RAS.

3.4 System Configurations and Intervention Process Phases

With the general background of navigation cycles in CAS/RAS, the associated steps in each cycle, and the deficiencies in the state of the art, the hypotheses motivating this presented work have been derived in the preceding section. In a second step, these generic hypotheses will be related to a specific system. In this sense, the following should be understood as a description of the presented system.

First, a short overview of the possible configurations of the system and their necessary navigation modules is given in Section 3.4.1. This is meant as a guideline for the reader to orient himself along the successive development steps. Second, these still-abstract modules are described as process phases as they appear to the user during intervention execution (Section 3.4.2). Together, they exhaustively describe the user's choices and experience of the system.

3.4.1 System Configurations

The proposed system can be used in any of three different configurations (Figure 3.10).

Based on a conventional workflow with pre-operative global imaging (e.g. CT), the system can compute optimal intervention paths, register them in the intra-operative setting, and execute them safely, including emergency stop mechanisms, force-based speed control, and manual force-following registration. This was part of the author's earlier work [Stolka01][Stolka02].

Extending this workflow, the main contribution of the present work lies in two proposed extensions to this base system. One is a navigation solution with robot-based 3D ultrasound map-building, which for the first time allows the intra-operative sampling of planning data and its one-time use for CT-free creation of inherently safe paths. The other navigation solution proposes a system which can perform the investigated interventions without any pre-operative imaging, and instead substitutes complete maps with concurrently-built, possibly incomplete maps created during the intervention itself as planning input.

Both approaches are based on local sensors (one on A-mode ultrasound, the other on force/-torque and audio sensors), all of which incur only minimal additional process overhead and patient stress. They can also be merged, combining their respective benefits. However, this was not investigated.

3.4.2 Process Phases

In this section, the proposed system will be presented from a process perspective, seen through a user's experience using the system. The robot-assisted navigation and intervention process is split up in distinct phases, each implementing a single functional step of the complete intervention. These phases are reflected in the graphical user interface (GUI) leading the user through the process and structuring the data acquisition, processing, and execution. The development

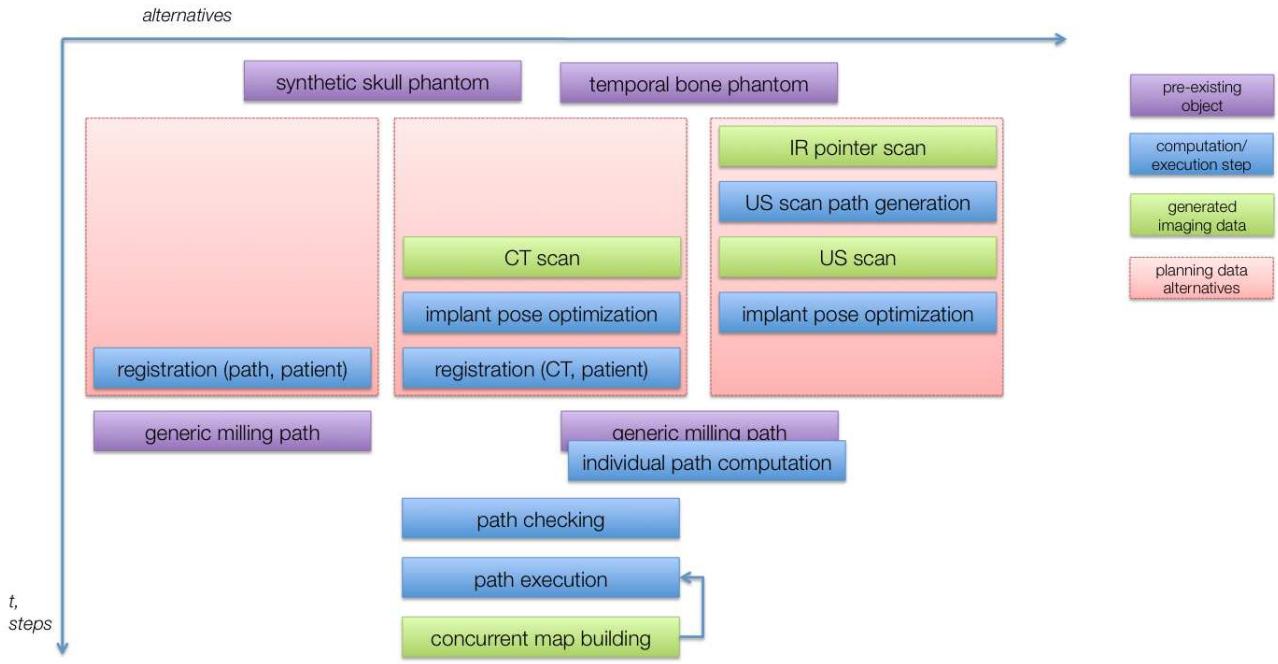


Figure 3.10: Three possible workflows (implemented and investigated) in the presented system: simple image-free intervention (left); CT-based intervention (center); and image-guided intervention with robot-based 3D-ultrasound (right). All alternatives allow concurrent map building, which enables iterative path re-planning during the intervention.

of a user-friendly GUI was part of the presented system [Peters03], but will not be discussed in depth.

3.4.2.1 Phase "Configuration"

In the first phase, the system is configured according to intervention needs (Figure 3.11). By choosing the desired intervention type ("implant bed milling", "mastoidectomy") and available navigation support ("CT imaging", "3D ultrasound", "tracking system support", "local sensors"), the modules representing the later phases are activated appropriately. For this work, the combinations including the "implant bed milling" intervention and non-CT navigation support options will be relevant.

Other available configuration options refer to the force-following motion control for registration ([Stolka03], Section 5.4.2) and force-based milling speed control ([Henrich02], Section 6.1.2.3.1).

3.4.2.2 Phase "Implant Selection"

In case of implant bed milling, the geometric model for the chosen implant is loaded next. This triangle surface model (in standard ASCII STL format [STL]) will later be used to perform position optimization and path planning. Available implant models include (cf. Figure 1.5):

- *MED-EL Combi 40+ (C40+):* cochlear implant
- *Advanced Bionics Clarion:* cochlear implant
- *Symphonix Vibrant Soundbridge:* partially implantable hearing aid
- *Implex TICA:* fully implantable hearing aid

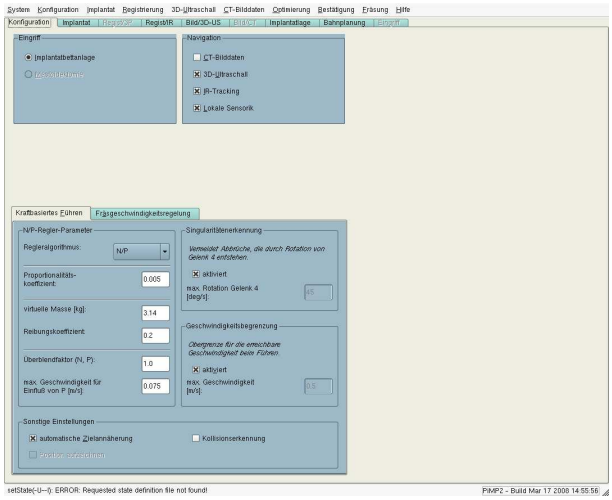


Figure 3.11: Screenshot of the process phase "Configuration" (showing available intervention types, imaging modalities, and parameters for force following control).

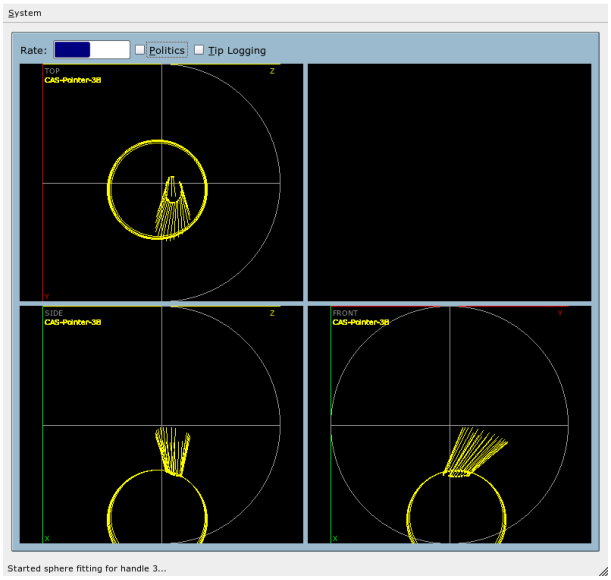


Figure 3.12: Screenshot of the OTS control window during process phase "Registration" (while acquiring a pivot center location for handheld pointer calibration).

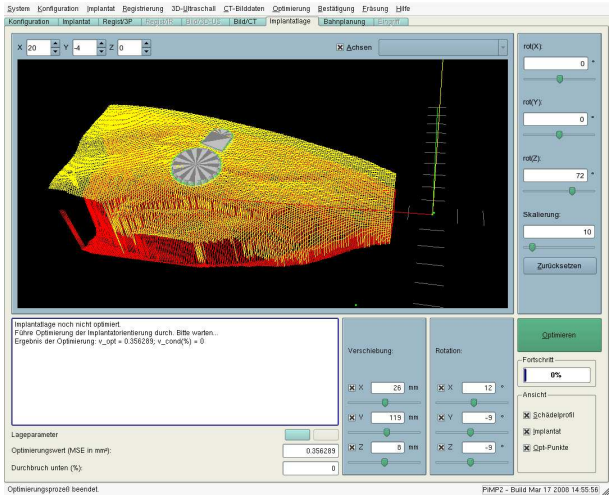


Figure 3.13: Screenshot of process phase "Implant Position Optimization" (with Soundbridge implant model and CT-based image data, after an automatic position optimization step).

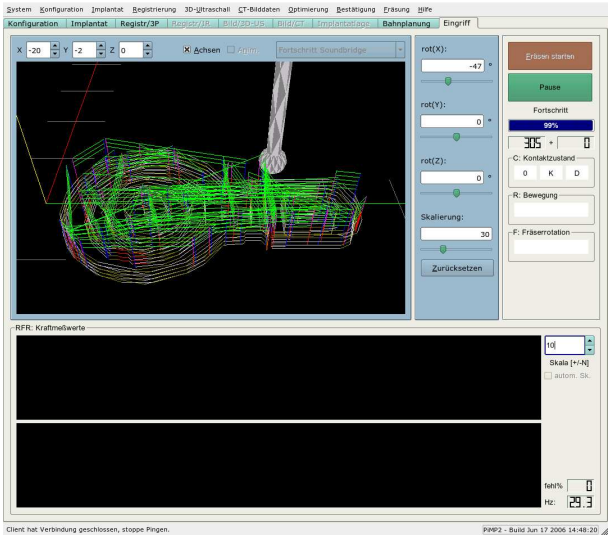


Figure 3.14: Screenshot of process phase "Path Execution", showing a miller tool being moved along a precomputed implant bed milling path.

3.4.2.3 Phase "Registration"

In the presented system, it is possible to position milling volumes (whether implant or general geometries) within the robot workspace without the use of pre-operative imaging data.

In case no optical tracking system support was selected, the position of the milling volume can be provided to the system by force following, i.e. manually guiding the robot to positions which in turn make up a local coordinate system giving the milling volume position [Stolka03]. Alternatively, the robot is first registered with the optical tracking system. Then the user can define the points giving the milling volume position manually with a hand-held tracker (Figure 3.12, [Stolka06a]). As a third alternative, the volume position can also be determined without direct demonstration, instead offloading the task of finding an optimal placement to the later step of implant position optimization [Waringo03b]. In either case, the position is then transformed into robot workspace coordinates.

All of the above-mentioned approaches were investigated in the present work.

3.4.2.4 Phase "Imaging"

Imaging data can be provided to the presented system either pre-operatively (in this case as CT data, [Waringo03c]) or acquired intra-operatively (as 3D ultrasound, [Stolka07a]). While the former can be read in as a file that was generated beforehand, the latter requires the user to provide some input data to generate the robot scan paths along which the ultrasound scan will be performed during the intervention.

This second use of binary 3D ultrasound maps of the skull bone was developed and investigated in the described work, and marks a major step towards a novel imaging modality for robot navigational use.

3.4.2.5 Phase "Implant Position Optimization"

Once imaging data is available, the milling volume needs to be positioned within that data. While it is possible in general to perform this step manually and on-screen, a dedicated semi-autonomous optimization procedure can achieve better results in less time. This was not a focus of the presented work, but investigated within the scope of the general project ([Waringo03b], Figure 3.13).

3.4.2.6 Phase "Pre-Operative Path Planning"

Depending on whether a generic reusable milling path can be pre-loaded (for implant geometries) or an individual one needs to be planned (for individual path-length-optimal milling paths, or for mastoidectomies of general shape), at some point of time the computation of that plan needs to take place. For the latter case, the planning step uses the intersection volume of bone (extracted from the imaging data) and milling volumes as input for the sweeping algorithms creating the paths. This was also investigated within the general project [Waringo03c].

3.4.2.7 Phase "Path Execution / Milling"

The final process phase is the execution of the planned paths (Figure 3.14). This may happen either "blindly", i.e. without current sensor data to update the environment representation, or with local sensor support to build and update a map from local sensors of the region of intervention [Stolka05]. In the latter case, the actual execution path may differ from the pre-planned one, depending on local sensor measurements and detected environment features, leading to path adaption or re-planning.

Further aspects include the specialization of under-specified 5D milling paths for optimal executability and force-based speed control. All of these aspects were investigated in the present work.

Chapter 4

Local Sensors

The first step in establishing a local navigation cycle is to define local sensors capable of providing relevant environment information from the ongoing process (Figure 4.1). This chapter will give a definition of local sensors (as opposed to global sensors) first (Section 4.1). Local sensors suited for surgical use (Section 4.2) are proposed and their functionality explained in Section 4.3 together with experimental results. Finally conclusions for their possible application are drawn in Section 4.4.

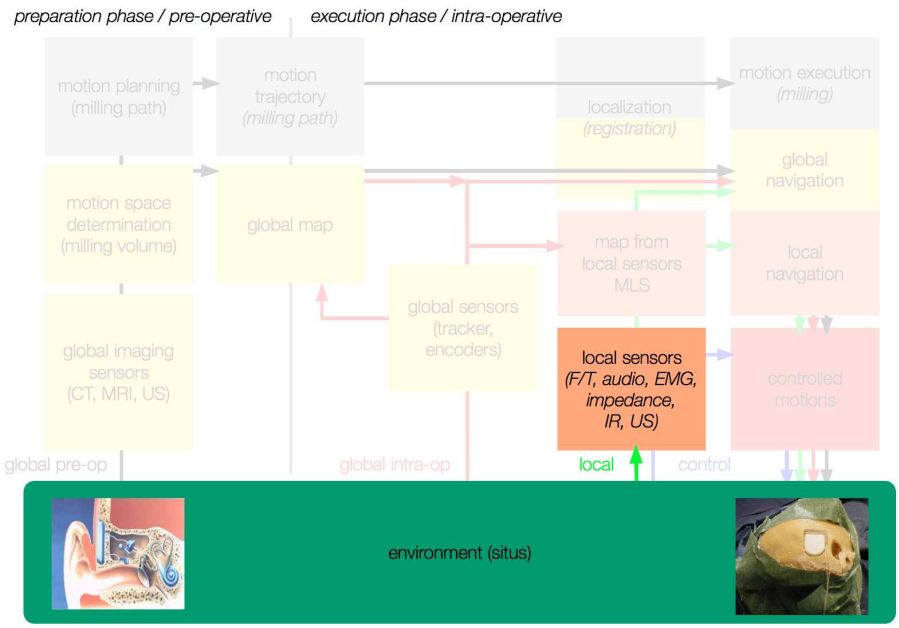


Figure 4.1: Relevance of local sensors within the general navigation cycles framework (modified from [Henrich04]).

4.1 Definition

From the overview in Table 1.2, one can conclude that the sensors currently in use for CAS/RAS applications are either inconvenient to use (high sampling costs, patient stress), not dynamic (high latency), or have only limited capabilities (e.g. no imaging capability). Important overall drawbacks of the complete sensor ensemble are:

- Imaging sensors cannot in general be used concurrently with process execution (apart from specific exceptions like magnetic resonance imaging (MRI) with MRI-compatible actuation mechanisms or like optical cameras which only sample limited subsets of the region of interest).

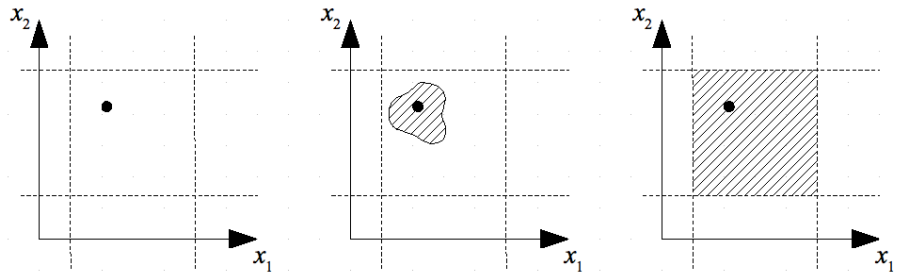


Figure 4.2: Comparison of the sensor resolution and known range (for image dimension $n = 2$ and also space dimension $q = 2$) of specific local sensors, as opposed to global sensors (black dot: sensor location; left: local with diffuse data, center: local with structured data, right: global; hatched area: sensor range where features could be distinguished) [Stolka07b].

- Additional, available information is not used at all.

An overarching question of particular importance for CAS/RAS is whether imaging or, more generally, map-building can be done concurrently. It seems reasonable to look for sensors that provide dynamic data, have low sampling cost, and are able to detect relevant environment features (this is of course desirable for applications outside of CAS/RAS as well.) One way of receiving such information is to sample data which is already being generated in the process without additional procedures.

4.1.1 Local Sensors

As briefly introduced in Section 3.1.1.2, a certain group of sensors – local sensors – is defined by their ability to sample data (only) from a local neighborhood of the actual sensor. More formally, we will call an appropriate sensor with the following locality characteristic a *local sensor*:

> For a *local sensor*, its m-dimensional sampled local sensor data $\varsigma(p) \in \mathbb{R}^m$ is descriptive only of the immediate ε-environment of the *interaction location* $p \in \mathbb{R}^q$ (a descriptor of the place the data is referring to, e.g. a tool tip position, in a space with q degrees of freedom), although the actual sampling can take place elsewhere (in a microphone, a laser scanner etc.). Within this ε-environment, the sensor data can be structured (i.e. have image dimension[1] $n > 0$, e.g. an image or A-mode echogram) or diffuse ($n = 0$; this does not need to imply that the sampled data originates from just one point, as e.g. with ambient temperature or neurological reaction to external stimulation by the tool).

This is contrasting with *global sensors*, which provide $\varsigma(p) \in \mathbb{R}^m$ for all $p \in \mathbb{R}^q$ simultaneously[2]. This difference in the descriptive power of local and global sensors is highlighted in Figure 4.2, with the coordinate axes x_i denoting the image and space dimensions (the information dimension m of their single sample elements is irrelevant for the definition of local sensors).

One important aspect of local sensors as defined above is that, in order to generate an encompassing environment map, they need to be tracked using some kind of localization device to associate the set of sampled ε-environments with poses $p \in \mathbb{R}^q$.

[1] The image dimension n refers to the dimensionality of the whole instantaneous sensor data sample (a subset of an n-dimensional space), to be distinguished from the information dimension or *modality*, referring to the dimensionality m of a single sample element (e.g. the number of color components of a sample pixel).

[2] "All" and "simultaneously" are obviously to be understood within limits dictated by the sensors' time resolution and spatial range.

4.1.2 Tool-based Local Sensors

Certain kinds of local sensors – used together with tools – can have additional properties:

- *tool-mounted:* The sensor is attached directly or indirectly to the (possibly robot-mounted) tool. Normal tool interaction drives the physical process that generates the input data, so there is no additional, extrinsic procedure necessary to sample the environment. (An example used in the present work is a force/torque (F/T) sensor mounted between robot tool flange and an electric surgical miller, sensing the forces and vibrations generated by bone milling. Other less direct tool mounting variations could be audio sensing for laser-based bone ablation.)

- *tool-determined:* The sensor's spatial range and the set of discernible features directly depend on the tool used for interaction. (In the above example, the range is determined by the size of the milling tool head, whereas features need to have substantially different F/T signatures to be distinguished by the sensor.)

- *dynamic:* Interpreting data originating from the process in real time, the sensors' latency can be kept to a minimum, providing a dynamic view of the intervention. (Unlike global but slow imaging modalities such as CT, local sensors provide incremental but immediate information.)

As a corollary from these latter properties, such *tool-based local sensors* incur only minimal additional stress and cost, as their input is a by-product of the process which is taking place anyway. Although their *information dimensionality* can be high (i.e. can be a high-dimensional feature vector per sample element), their *image dimensionality* may be zero, as their diffuse data is describing just the immediate tool position environment; that is, their data may not naturally embedded into an associated local coordinate system.

Examples for tool-based local sensors are force sensors mounted between a robot and a tool (e.g. a miller) and audio sensors (microphones sampling process noise). Both are used in the presented system.

Another local sensor – not tool-based – is a robot-mounted ultrasound sensor. The presented system includes such an A-mode (i.e. one-dimensional) ultrasound sensor specifically adapted to measure bone thickness.

4.1.3 Destructive Sensing

One important property of local sensors is that they may be associated with destructive sensing applications. There, sensor data sampling is performed during interaction of the tool and the sensing area which involves (at least partly) an irreversible modification of the current interaction region – e.g. milling away the intersection volume of miller head and workpiece. This modification exposes regions of the sensing area to the local sensor to which they were previously unreachable.

For certain sensing principles, this interaction can be the very process by which the actually sensed signal is created (e.g. force/torque or sound originating in the milling process). While signal-generation and sensing are thus side-effects of the interaction, another resulting effect is the disappearance of the destructively explored region. It is therefore very important to provide for the sampling of sub-regions extending beyond the main region currently being destroyed, to allow for any *post factum* analysis at all. In the case of signals being generated by the milling operation of a miller head, this *active interaction region* – when viewed on a small scale – is actually only the region of space spanned up by the miller blades extending into the milled material in the direction of miller motion (possibly distorted or extended slightly by the vibration of the miller head at the time scales relevant for the sensing process).

4.2 Local Sensors for Surgical Milling Applications

Before describing the single sensors chosen for the presented RONAF system, first its specific properties such as system states (Section 4.2.1) and the states' relative dominance behaviors (Section 4.2.2) need to be outlined.

4.2.1 State Identification and Definition

The local sensors are part of a system which moves through a state space S_{all} specific to the given application (given as the cross-product $S_{all} = S_0 \times S_1 \times ...$ of the sub-state-spaces S_j defined by the constituent system components). It it desirable to define a sub-space $S \subseteq S_{all}$ which is both observable through the sensors and relevant to the control of the application. Whenever this is possible, a "cheap" and "useful" method of system introspection is available to the system designer. In this case, the local sensors in a given true state (which may be unknown in detail) from S_{all} yield a vector of observations $\Sigma = (\varsigma_0, \varsigma_1, ...)$ from the most current data window (ς_i being the sensor information of the ith sensor) which is then mapped by a classification function Γ to said observable sub-space S:

$$\Gamma : S_{all} \times \Sigma \rightarrow S \tag{4.1}$$

Considering map-building in particular, it is important for S to contain information about the persistent features of the environment (i.e. not only transient features of the system). Such a persistent property can be e.g. an environment material class, derived from a contact state C between sensor and environment, or more generally any local environment feature. Only then can the local sensors' information be used for map-building.

For the investigated surgical application, the system comprised of robot, control PC, tools, and patient is moving through such a state space S, which is shown in table form in Figure 4.3. Its important features – from the point of view of the control system – are defined by the dimensions

- *robot motion* R: The robot is either standing still (short $R = R_0$ for $v_{robot} = 0$) or moving ($R = R_1$ for $v_{robot} \neq 0$, with v_{robot} measured as the robot end-effector's position differential on a discrete time scale),
- *miller activity* F: The milling tool – being the main interaction device of the robotic system with the patient – is either switched off (short $F = F_0$) or switched on ($F = F_1$),
- *contact state* C: The tool interaction with the environment – in particular with the patient – can be either milling in air (short $C = C_0$), milling in bone ($C = C_B$), milling on the dura mater ($C = C_D$), or an unwanted resonance condition ("tool chatter", $C = C_R$).

Together, the sub-system hopefully observable by the local sensors can then be described by its *(aggregate) system state*

$$s_{RFC} \in S = R \times F \times C \tag{4.2}$$

which is changing according to an *evolution (transition) function* $\delta : S \times A \rightarrow S$, where A is the combined input from the control system, the operator, and environment influences. However, especially the latter is considered to be beyond the control of the system (in particular of the generic, stand-alone sensor classification part under discussion here), so effectively δ could be only described stochastically, if at all.[3] Therefore, to keep the classification algorithms simple,

[3] A natural extension of this approach would be a description by a set of Hidden Markov Models (*HMMs*), which models the uncertainty of the current (by maintaining the overall probability for each single HMM) and the next state (as transition probabilities to the next state in each HMM).

		R Robot			
		0		1	
C Contact	0 (Air)	000	010	110	100
	Bone	00B	01B	11B	10B
	Dura	00D	01D	11D	10D
	Reson.	00R	01R	11R	10R
		0	1		0
		F Miller			

Figure 4.3: System and contact states s_{RFC} considered relevant for the investigated milling application (R/Robot: "0" stopped, "1" moving; C/Contact: "0/Air", "Bone", "Dura" milling in the respective materials, "Reson." resonance state; F/Miller: "0" stopped, "1" operating). The red states are detectable by audio sensors, green states by F/T sensors (adapted from [Stolka05]).

system history can be ignored, δ is left out of consideration, and the problem of state estimation can be reduced to the *(aggregate) steady-state classification* (i.e. assuming stationary signals, ignoring system history and the current state)

$$\Gamma' : \Sigma \rightarrow S \tag{4.3}$$

of sensor information Σ. In this notation, sensor data fusion of the $(\varsigma_0, \varsigma_1, ...)$ can naturally be included as well.[4]

For each local sensor K with its own Γ', a mapping $f_K = \Gamma'|_C : S \rightarrow C$ may be defined, which projects or "down-converts" the specific sensor's detailed classification results $s \in S$ into more general (less specific) contact states from C, i.e. $c = f_K(s)$. In the following, the considered tool-based local sensors $K \in \{\text{Force}, \text{Audio}\}$ are described in this simplified way, returning contact states $c \in C$ instead of classification results $s \in S$. Here, the mappings f_K are basically simple projection functions $f_K : S = R \times F \times C \rightarrow C$, with $f_K() = f_{Force}()$ or $f_K() = f_{Audio}()$.

4.2.2 Classification Dominance Relation Sets

The physical area or volume of the local sensor component contributing to the sensor measurement by sampling the signal may be non-negligible (i.e. cover a non-point-sized region, cf. Section 5.2.3.1).

Sampling of certain environment features across part of this active shape and thus classification into the corresponding state can mask other states occurring across other parts of the shape. Examples of this would be the differentiation between contact and non-contact states while only a part of the miller head is in contact and another is not (Figure 4.4). This behavior can be expressed in terms of a *sensor classification dominance relation set*

$$D_C = \{(C_i > C_j)_k\} \quad \text{with} \quad i \neq j; C_i, C_j \in C; k = 1...n \tag{4.4}$$

in which detection of a state $c \in C_i$ will suppress detection of any $c \in C_j$, even if a C_j-corresponding signal were generated across part of the tool.

Note that this definition of D_C is a phenomenological description of the specific sensors' operational characteristics, and can be specific to individual sensors.

[4] In the presented system, the values of R and F influence the classifier for C in a cooperative fashion, by modifying prior probabilities.

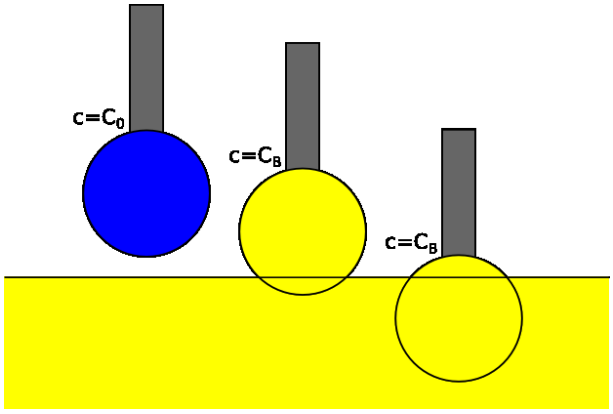


Figure 4.4: Effect of classification dominance, shown schematically for a miller tool head (from left to right: sinking from air into bone contact; even slight bone contact "overrides" air detection).

4.3 Investigated Local Sensors

In this section, the investigated sensors will be described in detail – their characteristics (stemming from the sensor technology itself), the chosen extracted features, and the final classification results for the two tool-mounted local sensors force/torque (Section 4.3.1) and audio (Section 4.3.2). As will become clear in the following subsections, both sensors cover different subsets of S, with high-pass audio discriminating between contact and non-contact situations (and several states within that latter subgroup), and low-pass force/torque differentiating between bone and dura situations.

As a third local sensor, independent of any milling procedure, A-mode ultrasound offers the possibility to measure bone thickness non-invasively, intra-operatively, and without additional registration to pre-operative data (Section 4.3.3).

4.3.1 Force/Torque Sensor

The force/torque sensor (called *F/T sensor* or *F/T* for short in the following) provides low-frequency (down to steady-state or zero frequency) contact information [Stolka05]. It has four distinct functions in the presented system.

First, its role as contact or milling force sensor during the actual milling process allows the concurrent control of resulting forces, keeping them below a specified limit while at the same time allowing fast progress. This *force-based speed control* serves to prevent thermal injuries (*necroses*) of the bone tissue due to excessive heat energy generation of the bone removal process, as well as to prevent process abortion due to force protection exceptions (where absolute forces exceed another predefined system limit). This control functionality is described later in Section 6.1.2.3.1.

The F/T sensor's second role is its use as a human input interface for *force-following registration* (zero-force control of the end effector). In order to define the location of the implant coordinate system within the robot workspace, the operator can manually drag the robot-attached tool to three locations in space (near the intended intervention position) which in turn define the resulting implant position. This is achieved during the registration phase by monitoring the 3D forces and computing motions in their directions so as to minimize the forces. This will be described in Section 5.4.2.

Third, during robot-based 3D ultrasound scanning, the force sensor data is used in a *hybrid pose/force motion controller*. This adapts the scan path incrementally during an orthogonally force- and pose-controlled scan to ensure both following a pre-specified path as well as sufficient sensor pressure (measured indirectly by force) onto the scanned area for good acoustic coupling.

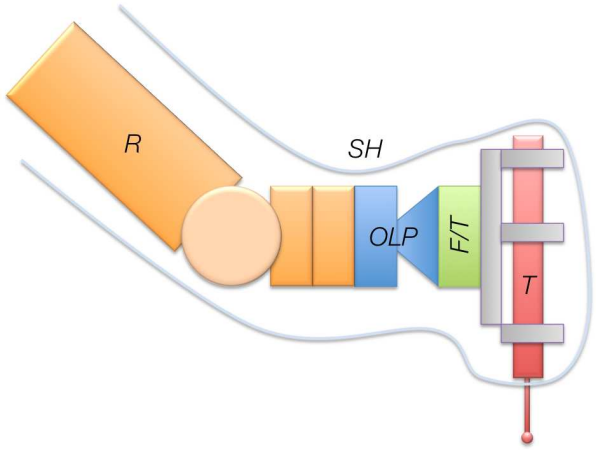


*Figure 4.5: Schematic view of robot (*R*), pneumatic overload protection (*OLP*), force/torque sensor (*F/T*), tool (*T*), and sterile hull (*SH*).*

Figure 4.6: Six-axis force-torque sensors used in the presented system (left: JR3 90M31A for use with the Stäubli RX130, right: JR3 50M31A for use with the Stäubli RX90/CASPAR) [JR3]

This is detailed in Section 5.3.1.7.

Finally, the most specific task w.r.t. navigation is the *classificiation of discrete contact states*. This is achieved by continuous sampling of F/T data during the milling intervention and its concurrent submission for classification. This yields information about the local environment of the tool attached to the F/T sensor, and thus renders the latter a true local sensor as defined before. This will be described in more detail in the following sections.

4.3.1.1 Setup

The used six axis force-torque (F/T) sensor is based on the strain gage principle, containing an internal bridge-like structure separating the robot- and tool-side flanges which holds the metal foil strain gages. External forces and torques effect varying resistance across the gages, from which a decoupling step computes the current values with minimal overall distortion.

For the two robots used consecutively in the presented system, two different F/T sensors were necessary (Figure 4.6): initially a JR3 90M31A for use with the Stäubli RX130, later a JR3 50M31A for use with the second-generation system based on a Stäubli RX90/CASPAR. Both provide high-frequency high-precision six axis force and torque data directly to the robot controller (for details cf. Section A.2). They were mounted in between a pneumatic overload protection module (serving as a last-resort emergency stop mechanism; only present on the RX130 setup) and the tool holder itself, i.e. very close to the tool-tissue interaction location itself (Figure 4.5).

4.3.1.2 F/T Sensor Data Processing

The F/T sensor used here is sampling and providing force and torque vectors continuously at a rate of up to $8kHz$. However, only a fraction of this bandwidth can be used as real-time input, due to the multitasking system architecture comprising several intertwined real-time tasks on the robot controller itself (cf. Section A.1). The low-level F/T data can be filtered within the F/T subsystem, but this functionality was turned off so as not to introduce additional latency; the data to be processed further is thus a merely geometrically transformed version of the data measured by the strain gauges.

Force/torque sensor data can monitor the contact-based, low frequency system response. Its sensing principle allows this to extend down to include steady-state (i.e. zero or *DC frequency*) information, in contrast to e.g. the audio sensor described later. In the RONAF system, 3-dimensional force data vectors $\varsigma_i = F(t_i)$ are sampled continuously at a variable rate of $20...200\,Hz$ (varying depending on the system setup). From this data, all functions of the F/T sensor – speed control, force following, contact force control, and contact state classification (Section 4.3.1.4) – derive their input. However, for proper interpretation of force/torque data irrespective of the current orientation, it is necessary to compensate for gravity effects first (Section 4.3.1.3).

As mentioned before, the same data serving as classification input is used directly on the robot controller for speed control purposes as well (in the "Control" navigation cycle).

4.3.1.3 Gravity Compensation

As later versions of the presented system introduced variable-attack-angle steps (including ultrasound scanning oriented along bone surface normals and variable-orientation milling), it is necessary to compensate for the time-varying gravitational forces $F_g(\theta)$ (for the tool orientation $\theta \in SO(3)$ (the special orthogonal group, i.e. all proper rotations around the origin of Euclidean 3D space $\mathbb{R}$)) on the F/T sensor, exerted by the tool holder's own mass m_{tool}. These weight-induced forces can differ from the calibration offset force vector F_0 measured during system start-up, since the tool holder's center of gravity generally does not coincide with the origin of the force measurement coordinate system. Therefore, a separate procedure for the determination of the mass m_{tool}, the center-of-gravity offset vector Δ_{tool}, and for subsequent compensation for off-calibration force vectors $F_g(\theta)$ had to be devised and implemented [Harrer07].

4.3.1.3.1 Determination of Tool Mass The tool mass m_{tool} influences all static[5] measurements ${}^{FRF}F_{measured}$ of forces and torques (F/T) in *force reference frame* (or *FRF*) coordinates by an additive offset force (the *tool-mass force*) $F_g = m_{tool} \cdot g = m_{tool} \cdot 9.81\frac{m}{sec}$ along the global (world) Z axis, i.e.

$$ {}^{global}F_{expected} + \begin{pmatrix} 0 \\ 0 \\ F_g \end{pmatrix} = \left({}^{global}M_{FRF} \cdot {}^{FRF}F_{measured} \right) \tag{4.5} $$

To determine the offset force, a three-step procedure is performed. First, the robot moves the tool into an upwards-pointing position P_1 (Figure 4.7, top left) and resets the F/T measurements ("zeroes the forces") after standstill, so the measured forces $F_{off1} \approx 0$. Second, the robot moves the tool into a downwards-pointing position P_2 (*dito*, top right) and measures the forces F_{off2} (which because of the previous reset now amount to twice the actual tool-mass-induced

[5] I.e. ignoring any dynamic force-inducing effects like inertia. This can be justified by the low driving speeds attained by the system – because of safety considerations – at any time during the intervention where F/T measurements are actually relevant.

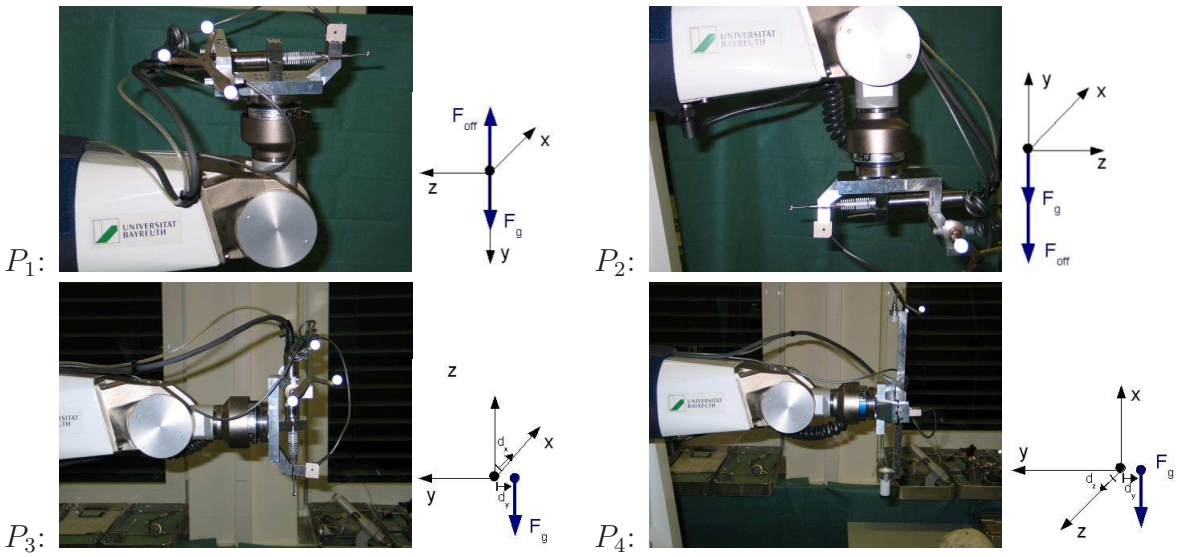


*Figure 4.7: Robot poses to determine parameters for gravity compensation, with corresponding force reference frame (*FRF*) poses shown [Harrer07].*

force F_g). Finally, $F_{offset} = \frac{1}{2}(F_{off2} - F_{off1})$. The absolute value of the tool-mass force $F_g = \|F_{offset}\|$ is used for later gravity compensation.

4.3.1.3.2 Determination of Center of Gravity Having determined the tool-mass force F_g, it is now possible to determine the center of gravity (CoG) for the tool (and the complete tool holder combination). To do this, torque vectors T_3 and T_4 are measured in two additional positions P_3 and P_4 (Figure 4.7, bottom left and right). Because the origin of the force reference frame FRF is in general away from the CoG, those torques will in general be non-zero.

In P_3 there is no torque around the local ${}^{FRF}Z$ axis (which is pointing upwards), and the geometrical offsets $\Delta_X = \frac{T_{3Y}}{F_g}$ and $\Delta_{Y,3} = \frac{T_{3X}}{F_g}$ can be computed. In P_4 there is no torque around the local ${}^{FRF}X$ axis (which is now pointing upwards), and the geometrical offsets $\Delta_{Y,4} = \frac{T_{4Z}}{F_g}$ and $\Delta_Z = \frac{T_{4Y}}{F_g}$ can be computed. Together, the geometrical offset vector can be composed as

$$\Delta_{tool} = \begin{pmatrix} \Delta_X \\ \frac{\Delta_{Y,3}+\Delta_{Y,4}}{2} \\ \Delta_Z \end{pmatrix} \tag{4.6}$$

The force reference frame FRF is then shifted to this offset vector value (from its original position at the origin), which reduces future torques due to the tool mass in different orientations to near-zero.

4.3.1.3.3 Algorithm for Gravity Compensation Given both the tool-mass force F_g and the CoG offset vector Δ_{tool}, it is possible to compensate for the forces $F_g(\theta)$ that are varying with tool orientation θ and which are overlaid onto the measured force vectors ${}^{FRF}F_{measured}$. Unlike the measured torque vectors ${}^{FRF}T_{measured}$, the forces will still vary even after the shift of the FRF as described above.

To compensate, each force ${}^{FRF}F_{measured}$ measurement is accompanied by a tool-orientation ${}^{global}M_{FRF} = \theta$ transformation. The FRF-based force vector is transformed into a global force vector ${}^{global}F_{measured} = {}^{global}M_{FRF} \cdot {}^{FRF}F_{measured}$, then the previously determined tool-mass force F_g is subtracted from the now-known global force Z component, and the resulting force transformed back into FRF coordinates (Eqn. (4.7)):

$$
\begin{aligned}
{}^{FRF}F_{expected} &= \left({}^{global}M_{FRF}\right)^{-1} \cdot \left({}^{global}F_{measured} - \begin{pmatrix} 0 \\ 0 \\ F_g \end{pmatrix} \right) \\
&= \left({}^{global}M_{FRF}\right)^{-1} \cdot \left(\left({}^{global}M_{FRF}\right) \left({}^{FRF}F_{measured}\right) - \begin{pmatrix} 0 \\ 0 \\ F_g \end{pmatrix} \right) \qquad (4.7)
\end{aligned}
$$

This complete procedure yields force vectors ${}^{FRF}F_{expected}$ with only externally applied forces (on the tool tip) in the tool-based force reference frame, compensated for the varying tool-mass induced force. These can be combined with the torque vectors after the FRF shift based on the above CoG determination. Together, the resulting F/T values can be used as input for later force-based milling speed control as well as contact state classification.

4.3.1.4 F/T State Classification

After gravity compensation has taken place on the robot controller after raw data sampling, a feature vector $v(t_i) = (\mu(|F(t_i - d < t \leq t_i)|), \sigma(|F(t_i - d < t \leq t_i)|))$ of the sliding average and standard deviation of absolute forces $|F(t)|$ over a period of $d = 0.3\,s$ is computed[6], and finally a Bayes classifier assigns system states. These particular features were chosen because of their relative invariance under "irrelevant" transformations (such as changes in direction of milling motion or sample rate).

In principle, this classifier can be sensibly used to determine the states of the F (miller state) and C (contact state) variables. For training, the force data stream of a complete milling intervention was recorded and manually segmented. The feature vectors were computed and their values discretized (at steps of $\Delta\mu = \Delta\sigma = 0.2N$ within the range of $0...5N$) to fill non-parametric class-(state-) conditional density function matrices $p(v|s_{RFC})$. As the progress of milling is unknown from the viewpoint of the classifier, the a-priori probabilities $P_{RFC} = P(s_{RFC})$ are set to the same value for all states s_{RFC}. During later execution, the a-priori probabilities for s_{1**} and s_{0**} (robot moving/standing still) are set to 1.0 or 0.0 according to the result of the external robot motion indicator (the result of the classifier for robot motion R[7]). The general error-rate-minimizing Bayes decision rule of deciding for maximum posterior $P(s_{RFC}|v)$ is applied. Classification then takes place every $\Delta t_{aa} = 0.05s$ during execution.

This force-based classifier is extended to detect different contact states C, especially for differentiation between $C = C_B$ and $C = C_D$. As $C = C_R$ is characterized by high frequency components, one cannot hope to distinguish it with force measurements in this system due to the low sampling frequency.[8]

4.3.1.5 F/T Classification Experiments

Data for classifier training was collected in experiments using animal (pig skull) *ex-vivo* phantoms. Dummy implant volumes were milled into the calotte using process parameters determined from earlier experiments: $30,000rpm$ miller rotation speed (this has produced both the

[6]This value for d is chosen empirically; for relevant motions (i.e. milling motions through the bone with speed $v_{segType} = 4mm/s$ (cf. Table 6.1) this corresponds to a maximum robot displacement of $1.2mm$, i.e. an average expected data localization error of $0.6mm$, which can be supposed to be acceptable for the given application.)

[7]This binary classifier relies on the definition of a *minimal robot speed* $v_{min} = \frac{d_{eff}}{t_{max}}$ to reliably prevent thermal necroses, with d_{eff} being the effective tool diameter in contact with the bone (and can be approximated as $d_{eff} = 1.5mm$) and $t_{max} = 5sec$ being the maximum allowed interaction time for a rotating miller on the bone (from [Stolka02], based on [Fuchsberger86]).

[8]Depending on the system setup (cf. Section A.1), the cutoff frequency for F/T sampling generally was in the range of $20...200Hz$.

smoothest bone cavities as well as the smoothest force-controlled milling progress); $1.8mm$ horizontal milling path spacing (resulting in an acceptable ratio of remaining ridge height to overall path length for the used $4mm$ diameter burrs [Waringo01]); $6mm/s$ maximum milling speed (forward feed rate); $0.8mm$ vertical feed per iteration; and vertical miller orientation (perpendicular to the bone surface). Actual milling speed was controlled using force-based speed control from [Stolka01] set at $\|F\|_{target} = 10N$.

The resulting data is summarized in Figure 4.8, showing the histograms for states s_{000}, s_{010}, s_{100}, and s_{110} (varying R and F under $C = C_0$, respectively) as well as for s_{11B} and s_{11D} (milling in bone and on the dura). It is apparent that the relevant states are distinguishable using the F/T classifier. While generally milling on the dura shows substantially lower reaction forces on the miller, contact with bone surrounding "dura islands" (cf. Section 1.4.1) is frequent and results in spurious high-force spikes. Also, under blunt angles of attack (perpendicular milling orientation) the dura is very resilient against damage by the rotating miller, even under high force pushing the miller against the dura, so *transient* dura contact during implant bed creation does not make abortion of the robot-based intervention necessary.[9]

4.3.1.6 Classification Dominance Relation Sets

For the presented F/T-based local sensor in the discussed milling application, it may be assumed that its classification dominance relation set is $D_C = \{(C_B > C_D), (C_B > C_0), (C_D > C_0)\}$. This would imply that even small miller/bone contact regions between mask out miller/dura regions (i.e. $(C_B > C_D)$, because of the substantially higher interaction forces related to the bone), same for miller/bone vs. non-contact states (i.e. $(C_B > C_0)$), and finally even slight dura contact masks out non-contact states (i.e. $(C_D > C_0)$, because dura contact entails higher forces, higher standard deviation, or both).

Although this (reasonable) assumption has not been experimentally verified, it will be used later in the simulations of Section 5.2.4.

4.3.2 Audio Sensor

Apart from using the *in-situ* available force data, there is another source of information about the system state: recorded sound from the process. While forces can be measured only through a low-pass filter inherent to the sensor system, the audio channel serves as a complementary high-frequency, contact-free data source with no steady-state or zero-frequency information (comparable to measuring vibration data, but without spatial direction) [Stolka05]. Due to the high sample rate and computational load, processing and classification takes place on a personal computer (PC) – either the process control PC or a separate computer.

In the presented system, audio data ς_i is sampled through a directly attached room microphone mounted on a tripod stand (44.1 kHz mono/16 bit, directed at the region of intervention from approx. $1m$ above and to the side). Feature extraction takes place on a PC workstation, and classification results are transmitted to the main control module over TCP/IP. Time stamping allows synchronization of data with those from other sources that carry attached spatial information (e.g. contact state information), which is necessary to associate the audio information with a position. This approach classifies robot motion and milling without contact, under bone contact, and with resonance. Thus, the operational state of the miller and the contact state C are detected with a high sample rate (approx. $f_{audio} = 86Hz$).

[9] For a full description of observed effects and time-domain force graphs cf. [Stolka02].

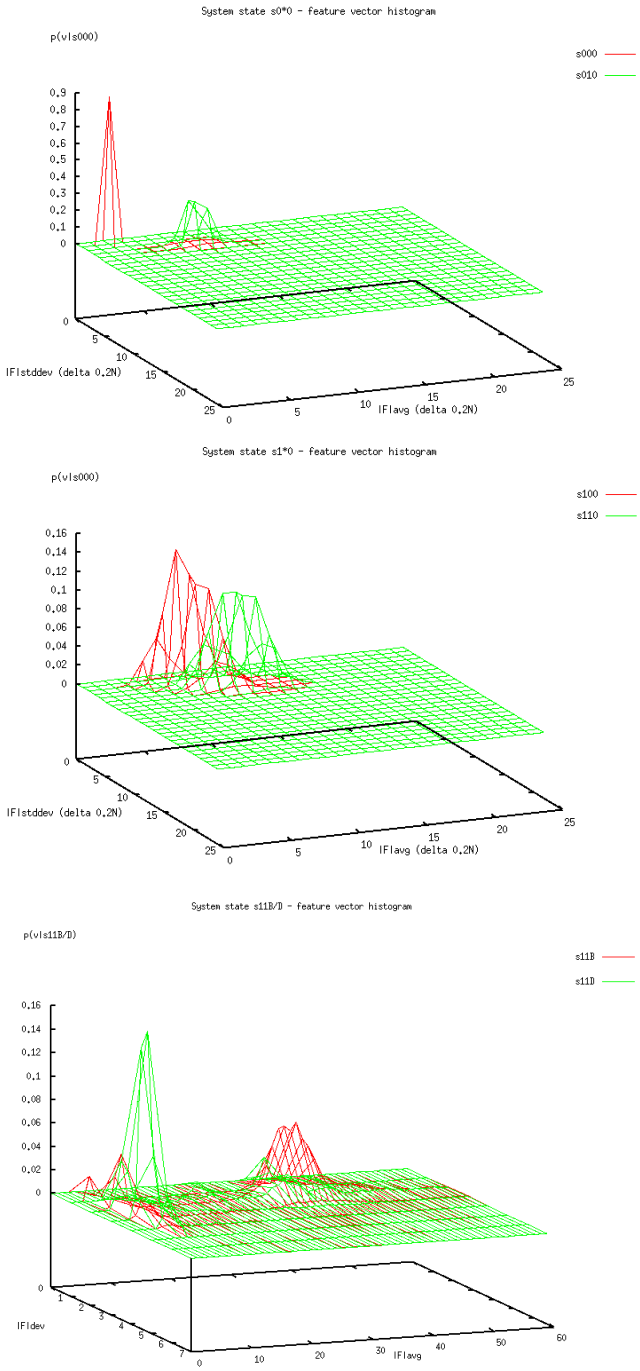


Figure 4.8: Feature vector histogram for s_{0*0} – robot in standstill, miller stopped/rotating (top left; units 0.2 N)*; feature vector histogram for s_{1*0} – robot moving, miller stopped/rotating* (top right; units 0.2 N); *feature vector histogram for $s_{11B/D}$ – robot moving, miller rotating, contact with bone/dura* (bottom; units N) *[Stolka05]*

4.3.2.1 Audio Sensor Data Processing

Feature extraction from the audio sample vector over a defined window size $w_{audio} = 512$ (approx. $11.6ms$) takes place both in the time as well as the frequency domain. Time domain features include maximum, minimum, mean, standard deviation, vector norm, mean crossing rate, zero crossing rate, and auto correlation. For frequency domain features, the FFT is computed using the FFTW library [FFTW] (after Blackman windowing to reduce aliasing artifacts [Harris78]) first, then the bandwidth, the median frequency, the volume, and subband energy distribution over n bands are computed. Various combinations of these features are composed into a final feature vector v_{FFT}, with or without element-wise normalization [Davy02].

4.3.2.2 Audio State Classification

As creating reliable, completely-covering histograms for the $p(v_{FFT}|s_{RFC})$ class-conditional probabilities in such high-dimensional feature spaces is nearly impossible – at least with limited training sets – each single feature vector v_{FFT} is classified using statistical *"leader-follower clustering"* [DudaHartStork02][Rojas93] for cluster learning with a Nearest-Neighbour approach for classification[10]. The classifier aims at differentiating between the classes s_{000}, s_{100} (i.e. robot standing still without milling and moving without milling), s_{*10}, s_{*1B}, and s_{*1R} (milling without contact, under bone contact, and with resonance). (As will be seen, robot motion R is not determinable with audio sampling.)

To this end, a complete milling intervention was recorded (cf. next subsection). For later validation, the whole audio stream was manually segmented and classified into clearly distinct process phases. The stream (classified, with newly attached s_{RFC} labels) was subjected to feature extraction algorithms (as described above) and the resulting feature vectors $v_{FFT,i}$ were alternately assigned half to a training set V_{FFT_train}, half to a test set V_{FFT_test} (of about 4760 vectors each).

For training, the class labels of training vectors were discarded, and a set N of m cluster centers n_i was initialized as $n_i = v_{FFT,j}$ (random samples in V_{FFT_train}). As the distribution of feature vectors was unknown, the number m of cluster centers had to be chosen large enough to ensure that all possible sub-clusters are covered (for the described system with the mentioned five classes, $m = 50$ later proved to be sufficient, i.e. classification rates deteriorated with fewer clusters.) Now, all samples $v_{FFT,i}$ from V_{FFT_train} were presented several times in random order (for the described system, three presentations were enough). The nearest center $n_{i'}$ by Euclidean distance was adjusted towards $v_{FFT,j}$ (in *"winner-takes-all"* fashion [DudaHartStork02]*; learning phase*):

$$i' = argmin_i \, \|v_{FFT,j} - n_i\|$$

$$n'_{i'} = n_{i'} + \gamma \cdot (v_{FFT,j} - n_{i'}) \text{ with } \gamma > 0$$

After this step, samples were presented to the classifier again to determine the represented class $s_{RFC}(n_i)$ of every cluster center n_i by creating class activation histograms $H_i(s_{RFC})$ over input classes for each cluster center and then setting their respective maximums as cluster labels (*association phase*).

4.3.2.3 Audio Classification Experiments

A series of audio recordings was made from clearly distinct system phases, based on the Stäubli RX90/CASPAR setup, with $30,000rpm$ miller rotation speed and standard $4mm/s$ robot speed,

[10] These functions are implemented in the *neuronetz* (clustering) and *claudio* (feature extraction and classification) software modules.

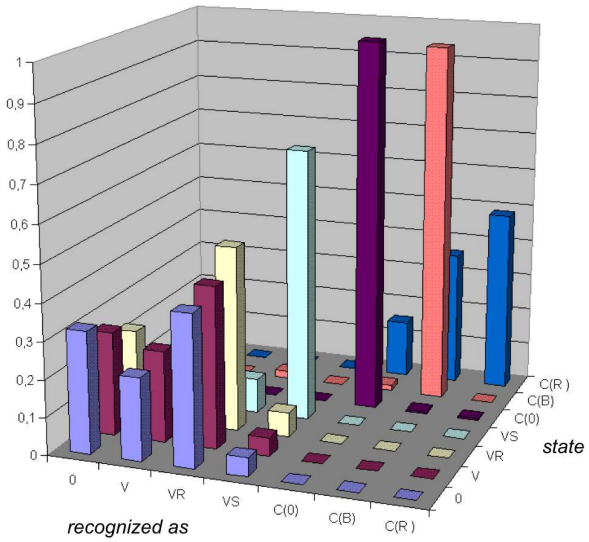


Figure 4.9: Confusion matrix for the audio state classification. Mapping of these states to the previously defined s_{RFC} system states is as follows: $\{0, V, VR\} = s_{00*}$, $VS = s_{10*}$, $C_0 = s_{*10}$, $C_B = s_{*1B}$, $C_R = s_{*1R}$ (resonance/tool chatter).

where applicable. While recordings of non-contact states were taken from a generic system startup position, recordings of contact states were taken in a similar experiment setup to the one from the F/T data recordings (Section 4.3.1.5), i.e. milling in an *ex-vivo* pig skull.

Testing classification rates on the test set V_{FFT_test} yielded results as shown in Figure 4.9 (for a feature vector of $n = 257$ FFT components and one absolute volume component), where the raw confusion matrix was constructed, with rows corresponding to "true" class labels and columns corresponding to the labels estimated by the classifier.

Except for samples from s_{100} (for audio classification purposes: states 0 (*"silence"*), V (*"ventilation"* or robot controller fan noise), and VR (*"ventilation+robot servos without motion"*)), all other states are mostly classified correctly. As there occurs a mix-up between s_{000} and s_{10*} (state VS (*"ventilation+motion"*)) judging only from audio data, one cannot reliably distinguish between a standstill and motion of the robot (noise obscures the robot hum). However, robot motion R can be deduced from other sources, as described before. Milling in air (C_0 or s_{110}) and in bone (C_B or s_{11B}) can be distinguished clearly using audio, while resonance (C_R) is mixed up mainly with bone milling, which is a closely related system state (resonance samples were manually selected from the more general C_B sample set).

Overall, standstill, free robot motion, milling in air, milling in bone, and partly resonance are distinguishable with the proposed simple audio sensor data classifier.

4.3.3 A-mode Ultrasound

The class of tool-independent local sensors is represented by the robot-mounted A-mode ultrasound.[11] This sensor provides precise distances for the acoustic impedance jumps associated with upper and lower calotte boundaries.

4.3.3.1 Physical Sensor Operation

A certain degree of imprecision is inherent when using CT data for intra-operative navigation, mostly due to partial volume effects and eventually the CT slice thickness (more generally,

[11] This physical and logical sensor was not developed by the author within the presented work. However, the robot-based 3D ultrasound mapping process presented in a later chapter is built on top of this sensor.

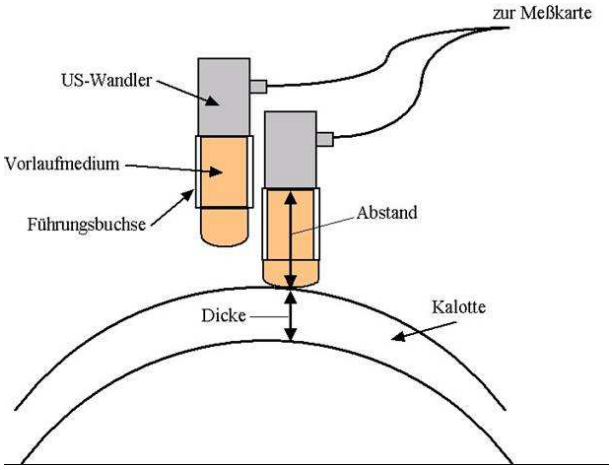


Figure 4.10: Diagram of the A-mode ultrasound probe developed for the presented system, touching down on the skull ("Kalotte"*) surface. Sound waves are emitted from the sound source (*"US-Wandler"*), traverse the delay line (*"Vorlaufmedium"*, contained within a sheath* "Führungsbuchse"*), are reflected partially at each skull boundary, and finally return as echoes into the probe and are transmitted to the sampling electronics (*"Meßkarte"*) (image courtesy of S. Tretbar; as presented at the author's technical presentation at the DFG-SPP1124 Workshop 2006).*

the reconstruction voxel size). Precision can be greatly improved – down to the submillimeter range – when assisted by ultrasound. However, as standard ultrasound is not optimized for hard tissue imaging, system requirements forced the development of a specialized A-Mode transducer/interface system (*SEM II*, Figure 4.10) which implements novel signal processing algorithms specifically geared to the measurement task at hand [Tretbar03] [Tretbar04a].

4.3.3.2 Sensor Data Processing and Sensor Integration

The overarching goal of the introduction of ultrasound into the discussed medical application is the automation of skull thickness measurements using ultrasound local sensing. This requires several specific sensor development and data processing steps.

The main challenge was the development of a specialized ultrasound system with accompanying methods to measure bone thickness e.g. with coded signal excitation (*chirps*). Due to the strong acoustic impedance mismatch between air (or coupling material, cf. below) and bone as well as bone and skull interior, a very high acoustic energy loss occurs on those boundaries. Furthermore, the spongious *diploe* between the twin layers of *tabula externa* and *tabula interna* absorbs a large fraction of the total energy on the two signal traversals in and out through the skull bone. Therefore, only a minimal signal amplitude is received from the second boundary, with energy close to the noise level, especially for the high frequencies needed to achieve high axial resolution.

To increase detectability of the second boundary, a combination of two techniques was employed. First, instead of a single-frequency "ping" pulse, a frequency sweep or "chirp" was emitted. The higher bandwidth decreases the average expected signal attenuation by the inhomogeneous tissue (as the signal is spread across a larger frequency range, some frequencies will usually be attenuated less than others), while a higher overall energy signal is emitted (due to the longer signal duration), which together increase total reflected energy. Second, the received signal is cross-correlated with the emitted signal, which increases the signal-to-noise ratio (*SNR*) of the detection signal and thus improves localization of the reflections (Figure 4.12).

To ensure optimal acoustic signal coupling for a sensor with a non-negligible acoustic aperture size onto an uneven bone surface, a flexible delay line had to be developed (Figure 4.11). Furthermore, both speed of sound and spectral transmission functions for ultrasound signals

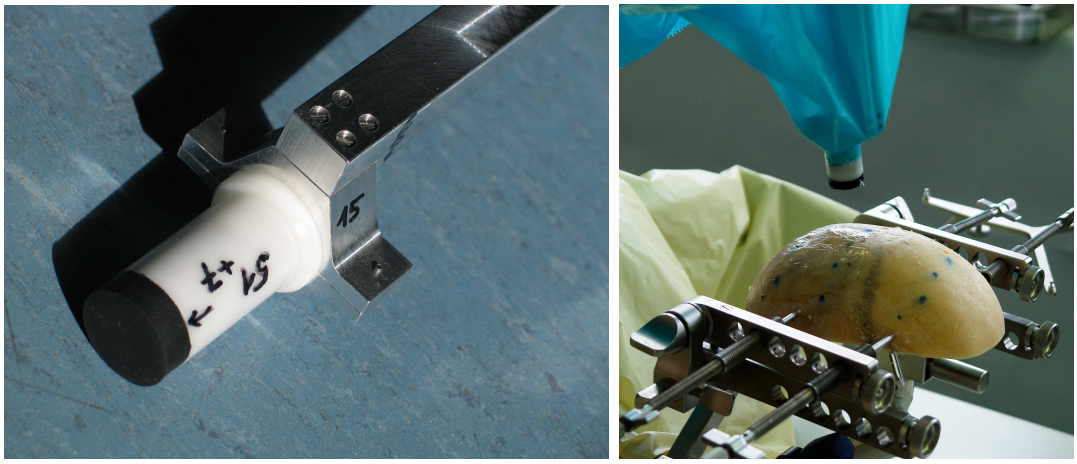


Figure 4.11: Ultrasound probe hull in the robot holder attachment (left; delay line in black); probe in use during a robot-based 3D ultrasound scan on a human calotte specimen (right).

in human skull bone had to be investigated. Finally, these were validated and optimized on physiological specimens using the three-step "*measure-mill-measure*" (*MMM*) procedure – first measuring the bone thickness, then milling away a certain amount of bone material, and finally measuring again; thus allowing to calibrate the value for speed of sound according to the measured decrease in depth vs. the known depth of the material milled away. All of these steps were performed mainly at the IBMT facilities and will not be discussed in detail here (cf. [Federspil05]). Nevertheless it is important to point out that using the described approach, 97.9% of thickness samples were measureable, as compared to only 62.5% using standard pulsed A-mode ultrasound. Also, the average bone thickness difference as estimated by this probe compared to gold-standard mechanical caliper measurements amounted to $0.04mm \pm 0.62mm$ [Tretbar09].

Other relevant steps for the integration of the new local sensor were the physical attachment of the ultrasound probe to the robot for the execution of robot-based scans, and the extension of the proposed system with an optical tracking system to acquire 3D ultrasound scans of the skull regions of interest [Stolka07a]. Both steps are detailed in the present work.

4.4 Discussion

In this chapter, local sensors were introduced, and the three particular ones that are the basis of the presented system were described and validated. Importantly, we set out to investigate Hypothesis 1 (Section 3.3.1):

> *"There exist local sensors, applicable for surgical milling interventions, which satisfy the evaluation criteria for sensors [cf. Section 3.3.1.2]."*[12]

4.4.1 Results

Three different sensors were introduced in this chapter: F/T, audio, and ultrasound sensing. The first two provide contact and system state information concurrently. The F/T sensor acts as a 3D low-pass filter providing bone/air/dura differentiation, but miller activity is not detectable reliably with this classifier, i.e. this requires a complementary high-frequency information source. Such a source is present in the form of the audio sensor, providing differentiated

[12] For reference, those criteria were *discriminatory power, reliability, precision, timeliness, speed, effort/cost,* and *burden/stress/exposure.*

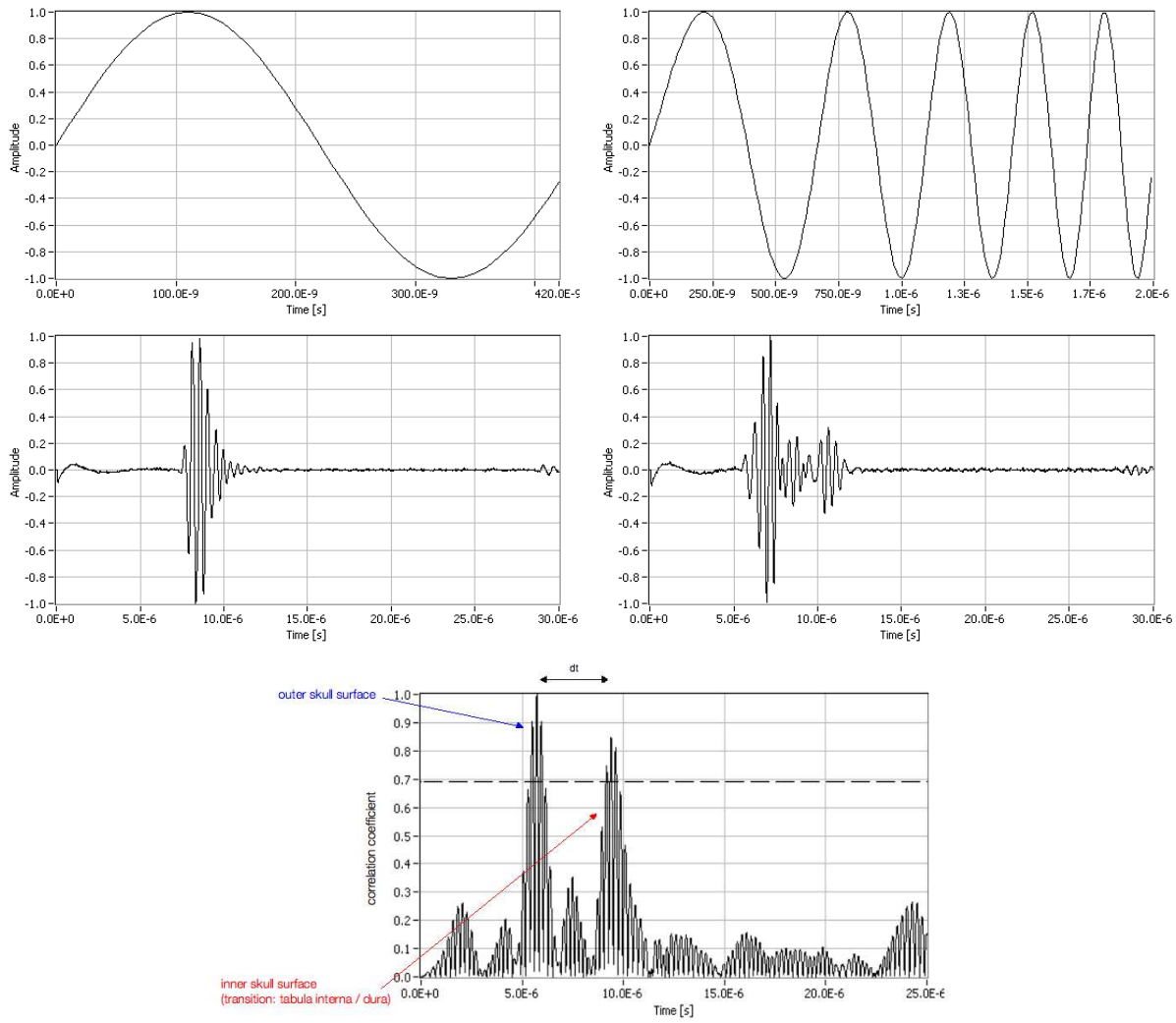


*Figure 4.12: Simple "ping" pulse (top left), "chirp" sweep (top right), received response signal from emitted ping (with practically nonexistent inner boundary reflection; center left), received response signal from emitted chirp (with weak inner boundary reflection signal; center right), and cross correlation of emitted chirp and returned echo signal (bottom), shown over signal time of flight. The first cross correlation peak correlates with the outer skull surface, the second with the inner skull surface, while the actual skull thickness was 5.7*mm *(all graphs courtesy of S. Tretbar).*

information about system and contact states – robot operation, miller operation, resonance, and bone/air differentiation – at a high data rate, acting as a 1D high-pass. The ultrasound sensor is able to provide precise bone thickness information (on the order of the later robot positioning accuracy) at an interactive rate.

Altogether, the system can classify the miller state F for both stopped and moving robot using audio data. Furthermore, the robot motion state R can be classified as well (using position information). Once both these information components are available, the a-priori probabilities for contact state C classification using F/T data are adjusted and C is classified.

The resulting system states are listed in Figure 4.3 and are discoverable through different logical sensors or classifiers. The robot motion substate R is classifiable through direct measurement of the differentials of position over time. The states marked green can be detected by force measurements, those marked in red by audio sampling. The (grayed-out) states s_{00R} and s_{10R} are impossible, as there can be no miller resonance without miller operation. The (white) states s_{00B} and s_{00D} (complete standstill on bone or dura) are not only indistinguishable with the presented sensors, but also improbable to occur except in case of complete system failure, which needs to be recovered through the human operator anyway. The (also white) states s_{10B} and s_{10D} (motion under contact without miller operation) lead to exceedingly high forces which cause system shutdown through several cascaded safety circuits [Federspil03] (also cf. Section A.2). All other states are covered by one or more classifiers.

4.4.2 Conclusion

It is probably safe to state that the combination of introduced sensors covers the range of relevant system states sufficiently well, and discriminates between these states reliably enough, to provide the information necessary for navigation and control of the presented robot-based system. In particular the two tool-based local sensors (audio and force/torque) sample data allowing to differentiate between the contact states *air/bone/dura* needed to build maps from local sensors in the next step. Furthermore, both provide relevant data concurrently with the actual intervention, and the third (ultrasound) sensor does so with only a short delay due to the independent measurement procedure. All three sensors are sampling data fast enough so that they are not forming a performance or accuracy bottleneck. In terms of effort/cost, data from both tool-based sensors is "free" (i.e. the sensors do not incur any additional overhead), while the later use of ultrasound for 3D volume reconstruction is cheap compared to competing imaging options like intra-operative CT or open MRI, although it still requires a certain small amount of preparation and execution work and time. Finally, the tool-based sensors do not have any discernible burden/stress/exposure effect on either patient or surgeon whatsoever, while ultrasound is a well-known and accepted imaging modality without any (relevant) side effects. (For an overview and additional information cf. Table 4.1.)

Compared to state-of-the-art sensors, the presented local sensors are examples that provide cheap, current, and precise environment information that can be used to concurrently update rich maps for navigational use. It should be noted that within the context of this work, the introduction of local sensors was not specially aimed at developing optimal classification procedures or at a full description of the range of possible local sensors; rather, it was important to provide a proof-of-concept set of such sensors to justify the subsequent navigation procedures (building maps from local sensors, and navigating on these maps).

4.4.3 Outlook

For the force-based state classification, using estimated environment compliance (absolute measured force normalized with robot speed) might be preferable over absolute force as basic data

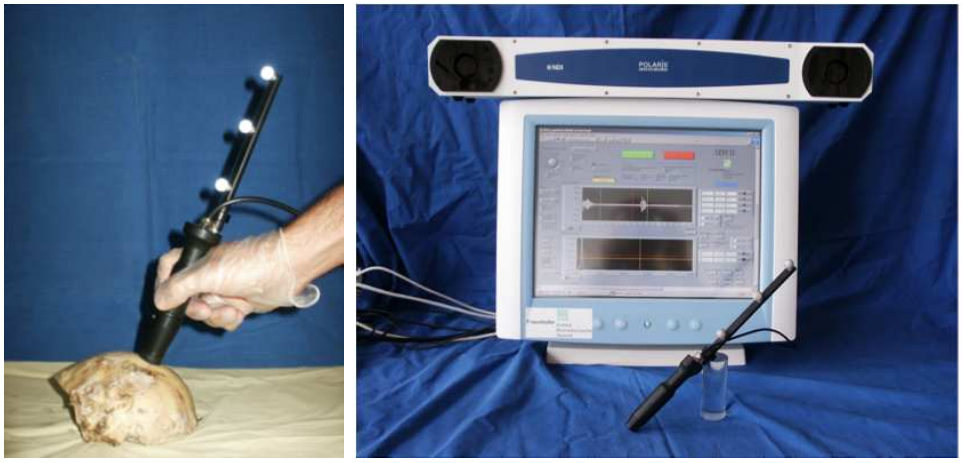


Figure 4.13: SonoPointer system for tracked handheld, precise 3D ultrasound bone thickness measurements – comprised of a handheld component with the ultrasound probe and a 5-DoF optical tracking marker (left) and a control component with the sampling software and optical tracking system (right) [Federspil06].

stream, since the influence of different speed control algorithms would be minimized. Furthermore, to make use of the directionality of force measurements, the force vector $\mathbf{F}(t)$ could be used instead of its absolute, possibly with a split 1D/2D representation separating the axial force component (indicating reaction force differences over bone vs. dura) from the lateral ones.

For audio classification, some system states exhibit strongly non-stationary behaviour in terms of spectral frequencies (with e.g. click-like transients, or environment changes due to destructive sensing). Therefore, a classification method that tracks the state over time might be useful (joint time-frequency methods like Hidden Markov Models to smoothen classification results, or principal frequency tracking may be valid options). Specialized pattern matching approaches might be able to detect contact state changes explicitly. The choice of features can be optimized with combinatorial testing or principal components analysis (*PCA*).

Furthermore, exchanging information between classifiers in a networked fashion and sensor fusion on a feature level, creating a composed feature vector, might enhance result quality by avoiding the information loss associated with the current hierarchical classification architecture. For both force and audio classification, subtraction of a sampled reference base-state signal (s_{000}) might help to calibrate the system to different environments.

Finally, the benefits of the directionality of force sensing and the bandwidth of audio sensing could be combined by introducing a 3-DoF vibration sensor, which would allow directional data sampling for handheld "smart tools", where steady-state force information may not be available.

Furthermore, the presented ultrasound sensor can be integrated with tracking into a handheld device as well. This can be the basis for quick and nonintrusive mapping using ultrasound for manual image-guided interventions, both with conventional tools as well as for tools employing "*Navigated Control*" to guarantee region constraints by controlling the instrument power (e.g. for implant bed milling in the skull [Lueth02] as well as spinal surgery [Trantakis09]).

Such a handheld 3D-US scanning device has been presented as the SonoPointer (Figure 4.13) [Federspil06]. The mean difference of thickness estimates measured with the SonoPointer and mechanical reference measurements can be as low as $0.3mm \pm 1.0mm$ [Federspil10], allowing the use of this system for both manual and robot-based intraoperative search of the optimal intervention region.

	Force/Torque	*Audio*	*Ultrasound*
spatial resolution	dependent on tool shape and temporal resolution		axial: high ($< 0.5mm$); lateral: minimal focus size $1.3mm$ [Stolka06a]
reliability	(?)	(?)	depending on angle
discernible features	bone, dura mater, air	bone, air, resonance; miller operation	bone signal entries / exits
latency	dependent on window size and network transmission latency		dependent network transmission latency
temporal resolution	medium (varying sample rate)	high (high sampling rate)	high (high sampling rate)
sampling cost	none (F/T sensor already present, active)	minimal (microphone placement in OR)	relatively high (additional sampling step)
patient stress	none (additional security by control)	none	minimal (non-invasive sensing, ROI exposition already done)

Table 4.1: Comparison table for presented local sensors (used for CAS/RAS)

Chapter 5

Maps from Local Sensors

In this chapter, the process of how to incrementally build maps from the local sensor data is described (Figure 5.1). In reference to Hypothesis 2 from Section 3.3.2, "faithful", "current/-concurrently built", and "relevant" are the requirements these maps have to fulfill.

Following an initial definition of maps from local sensors (Section 5.1), the map interaction possibilities (map projections, building and reading functions, and resolution optimizations) will be discussed for concurrently-built maps from local sensors (Section 5.2) and for independently-built maps from local sensors (Section 5.3). An important influence factor on map precision is localization (Section 5.4). Finally, experiments show the feasibility of building maps from local sensors in a RAS system (Section 5.5).

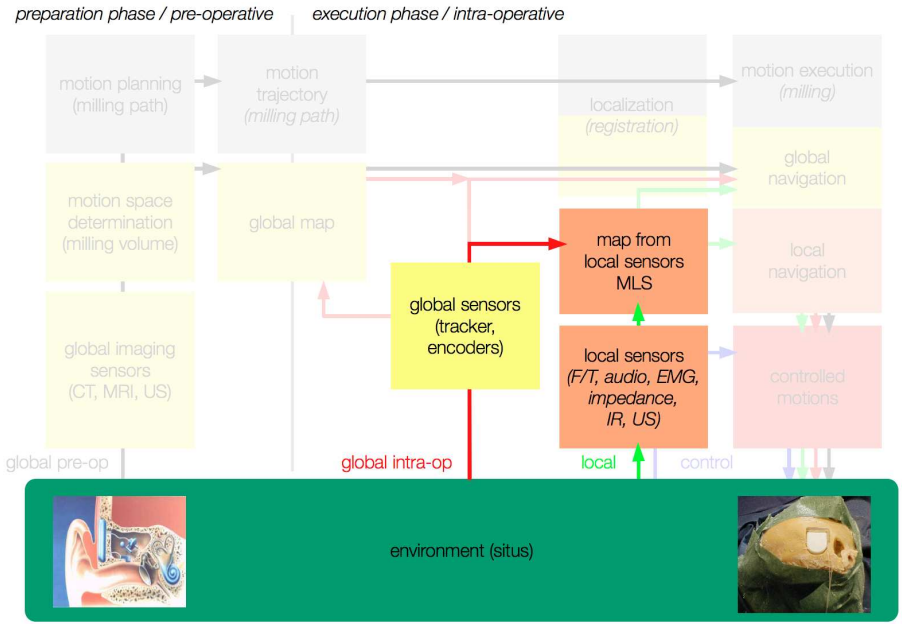


Figure 5.1: Relevance of maps from local sensors within the general navigation cycles framework (modified from [Henrich04]).

5.1 Definition

A *map from local sensors (MLS)* is an incrementally built global map M_{LS}, with its data originating from local sensors (possibly associated with a tool) incrementally traversing the environment E to be mapped (equivalent to the application's ROI). The sensor may perform non-destructive or destructive sensing, which in turn means that the mapping process may be destructive (i.e. irreversibly modify the region which underwent mapping). The tool-based

local sensor may have varying active shape components which actually contribute to the sensor information, depending on its orientation and direction of motion. The incremental nature of an MLS implies that its entries may be added to the map distributed over time, and repeated entries of updated information into the same position are also possible.

This type of map is different from the global maps defined earlier, which are static and/or complete at the time the sampling step is finished. In contrast to this, an MLS can be used after completion as what is effectively a static map (e.g. after a 3D ultrasound sampling run, cf. Section 5.3), or during creation or intermittent update steps as a dynamically changing map (e.g. one which is based on force-torque sensing, cf. Section 5.2).

Voxel grids are a useful environment representation for many applications with irregularly fragmented unstructured environments. For the investigated application, cartesian rasterizations of the environment are a common representation method for geometrical data (e.g. in CT, MRI, or other imaging modalities). However, the MLS concept can in principle be based on implicit primitives (e.g. vector- or spline-based) as well; its coordinates need not be cartesian either (but can be polar or based on robot joint angles). Important constraints are space (memory) requirements and the existence of distance measures.

5.2 Concurrently-Built Maps

The map M initially starts out empty. Together with location information (provided by localizers like tracking systems or robot encoders) suitable local sensors can be used to build maps of the regions they have already passed through, similar to a scanning process (Figure 5.2). If this can be done during the process execution, these maps are built concurrently with the intervention progress, and can thus be called concurrently-built maps from local sensors (short *"conMLS"*).

Map positions p can be labeled with $c \in C$ being either of three basically different environment feature classes from the set

$$C = \{C_0, C_+, C_?\} \tag{5.1}$$

which are C_0 (*empty space*), C_+ (*occupied space*), and $C_?$ (*unknown*), where C_+ can comprise several application-dependent sub-types (e.g. admissible/safe and non-admissible/unsafe regions, or application-specific states like $C_+ \supseteq \{C_B, C_D, C_R\}$, cf. Chapter 4)[1].

The map building function

$$b_M(c, t, p) = M'_{LS} \tag{5.2}$$

serves to insert local sensor information c into a possibly sparsely filled map M_{LS} based on location information $p \in \mathbb{R}^n$ with appended time parameter $t \in \mathbb{R}$ (this sensor datum shall thus be called $c_{p,t}$), creating an updated map M'_{LS} (Section 5.2.4). The query function

$$q_M(p, t) = c_{p,t} \tag{5.3}$$

extracts environment features $c_{p,t} \in C$ from this map for p and time t (if such a contact information was present at that place and time, otherwise $C_?$; cf. Section 5.2.5).

New entries $c_i \in C$ are sequentially appended to feature lists L_p of the MLS M_{LS}

$$L_p := \langle (c_0, t_0), \ldots (c_i, t_i), \ldots \rangle_p \in M_{LS} \tag{5.4}$$

[1] In and of itself, C_+ is not a feature class as such, but a set of feature classes, all of which denote a "state of established environment contact". For the sake of simplicity, however, it will be used analogously to C_0 or $C_?$.

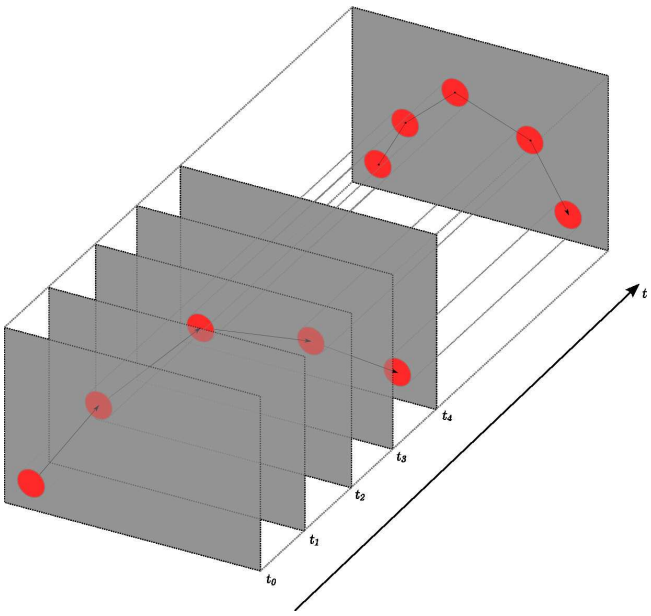


*Figure 5.2: Concept of maps from local sensors (*MLS*). New sensor information (red circles) associated with positions (2D in this illustration, foreground planes) at times t_i is entered into feature lists of the MLS at corresponding different positions (background plane).*

in their respective map positions $p \in \mathbb{R}^n$, time-stamped t_i, thus storing a complete history of the known states of features for that position. Each map voxel – here the rasterized grid nature of the map becomes apparent – then stores its own respective feature list. As local sensors have only a limited sensing range (cf. definition of local sensors), information in the MLS at different positions stems from different points in time. An alternative description of the MLS is thus a sparse, temporally (not necessarily spatially) sequential, $(n+1)$-dimensional representation of a traversed environment (Figure 5.3). In general,

$$M_{LS} : (p, t) \mapsto c$$

or equivalently

$$M_{LS} \subset \mathbb{R}^n \times \mathbb{R} \times C = \{(p, t, c)_i\}$$

In the following sections, only *rasterized* environment representations will be considered, i.e. a variety of occupancy grids, which are appropriate for tool-based local sensors. Another – point-based – type of map from local sensors (based on ultrasound measurements) will be described in Section 5.3.1.

5.2.1 Map Projections

An $(n+1)$-dimensional MLS can be reduced in dimension for navigation purposes in two different ways – spatially and temporally.

A *spatial projection/cut* $M_{LS}|_p$ of the MLS M_{LS} at position p results in a one-dimensional history of sensor information associated with p:

$$\tau(M_{LS}, p) = M_{LS}|_p = \langle (c_0, t_0), \ldots (c_i, t_i), \ldots \rangle_p$$

This can naturally be extended to allow spatial projections based on regions S as well:

$$\tau(M_{LS}, S) = M_{LS}|_S = \bigcup_{p \in S} M_{LS}|_p$$

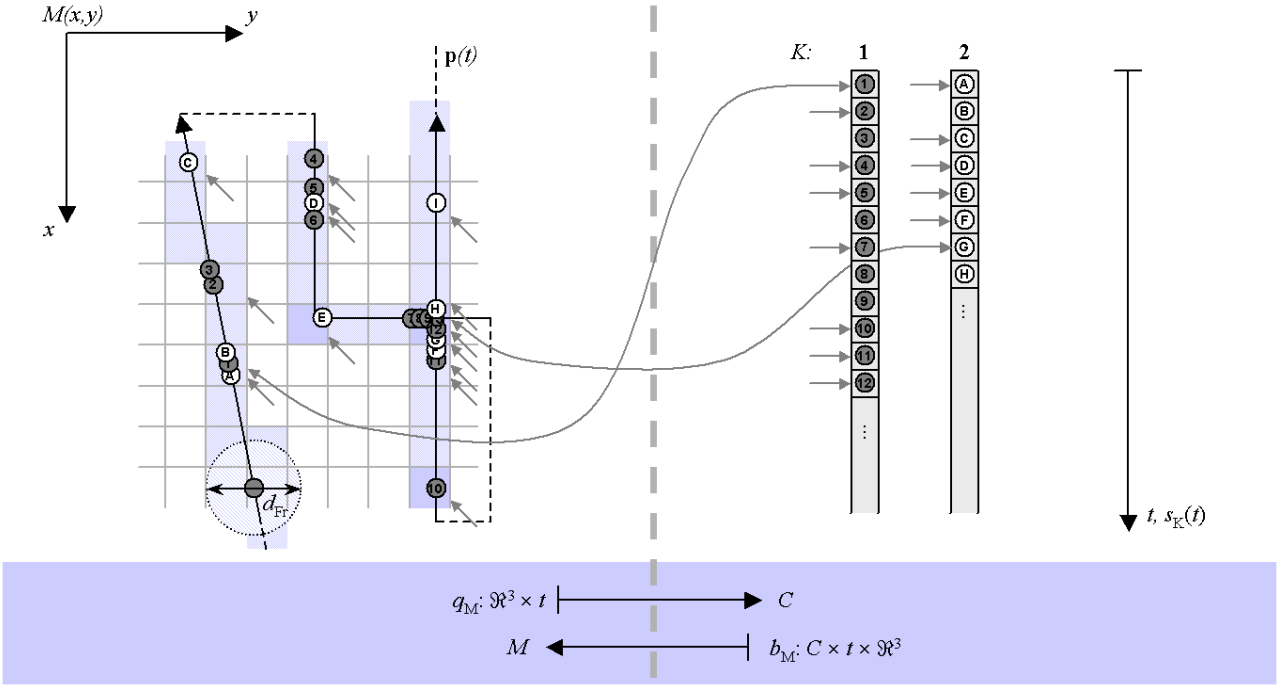


Figure 5.3: Spatial and temporal orderings related to maps from local sensors. Left side shows rasterized map M with tool head passing through represented region (diameter d_{Fr}) along milling path points ($p(t)$, black line), covering map entries along the way. Right side represents temporally ordered local sensors information (s_{K1}, s_{K2}). Both are connected by means of localizer information, which allows to store the sensor information in a spatial order [Stolka05].

Now, since maps from local sensors may be generated by destructive mapping, for each map position the initial feature ($C_?$), the first known feature (possibly C_+), and the last feature (possibly C_0) may differ, and what constitutes "interesting" information from the map depends on the goal of the query – i.e. whether to determine initial environment features, plan the next motion, or reconstruct the temporal development of the encountered environment (it is necessary to extract the head, tail, or an intermediate element of the features list in each respective case; cf. Figure 5.4). In fact, the most current view of the map is mostly empty except for narrow "bands" of information extending slightly beyond the explored region when using volume-removing tools (i.e. in destructive sensing) – the regions already explored have been emptied and marked C_0, while other regions are still unexplored and therefore marked $C_?$.[2]

A *temporal projection/cut* $M_{LS}|_t$ of the MLS M_{LS} at time t results in an n-dimensional map containing environment features known (if there is an entry present for time t), assumed (if there is no entry present for time t or earlier), or inferred (if there is only an entry present for an earlier time than t) at time t:

$$\tau(M_{LS}, t) = M_{LS}|_t = (c_{p,t'} | \forall p : L_p \in M_{LS})$$

where only rarely $(c,t)_p \in L_p$ will hold directly (because only very few positions of the map will actually contain information sampled at the very instant of t); instead the "most similar" t' for which actually $(c,t') \in L_p$ holds will be searched. The similarity measure $f_{sim}(t,t')$ yielding

[2]The volume traversed by such a tool contributed to the gathered contact information, but has been completely removed in the process. However, it is also clear that some information-contributing environment features may remain in the environment in the immediate vicinity of the tool. Their exact localization is unknown, though (this is elaborated further in Section 5.2.3), therefore the corresponding information is distributed across this neighborhood.

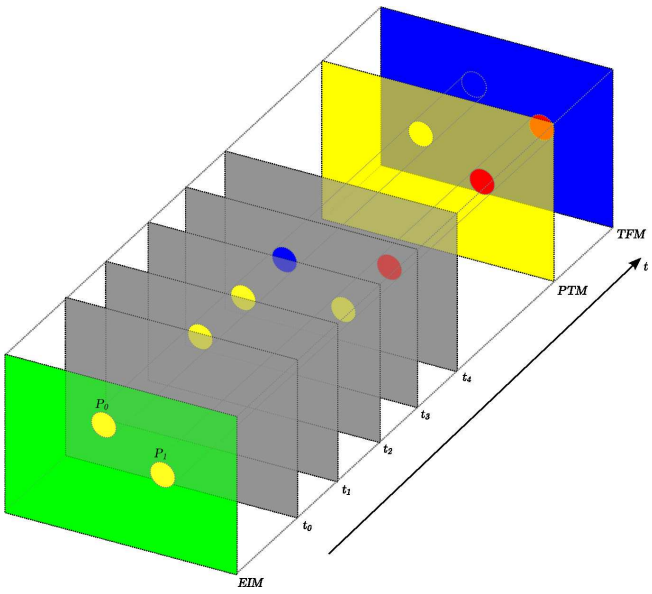


Figure 5.4: Temporal projection function τ selecting different feature list entries depending on time parameter (results for three different times for EIM (green), PTM (yellow), and TFM (blue) of a 2D map shown).

$t' = f_{MLS}(t)$ will vary depending on the chosen projection, as described for several temporal projections of special interest:

- *Earliest Information Map* (*EIM*): Defined as the 3D map containing the earliest known information about the environment which is different from $C_?$, the EIM carries the first estimation of environment features, if available (Figure 5.5):

$$\tau(M_{LS}, \text{EIM}) = (c_{p,t'} | \forall p : L_p \in M_{LS}; t' = \min\{t | c_{p,t} \neq C_?\})$$

- *Terminal Fate Map* (*TFM*): Containing the latest/most current known information, this is the best available approximation to the current environment situation (Figure 5.5 on page 88). However, it is clear that the TFM will carry only a very limited amount of information other than showing wide regions of $c = C_0$ – only the thin sensed boundary of the traversed region with the unknown region beyond, and even this only with active shape optimization (cf. Section 5.2.3.1):

$$\tau(M_{LS}, \text{TFM}) = (c_{p,t'} | \forall p : L_p \in M_{LS}; t' = \max\{t | c_{p,t} \text{ defined}\})$$

- *Pre-Terminal Map* (*PTM*): There exists another special temporal projection, however, which carries the maximum amount of information for each position *before* destruction: This map holds the last known information other than $c = C_0$, if available – i.e. environment information which underwent the most optimizations and refinements in the map during the existence of the corresponding environment feature before it was destroyed (Figure 5.6):

$$\tau(M_{LS}, \text{PTM}) = \left(c_{p,t'} | \forall p : L_p \in M_{LS}; t' = \begin{cases} \max\{t | c_{p,t} \notin \{C_?, C_0\}\} & \text{if} \quad \{t | c_{p,t} \notin \{C_?, C_0\}\} \neq \emptyset \\ \max\{t | c_{p,t} \text{ defined}\} & \text{else} \end{cases}\right)$$

In the following, mapping of a three-dimensional cartesian workspace will be usually assumed unless noted otherwise (i.e. $n = 3$).

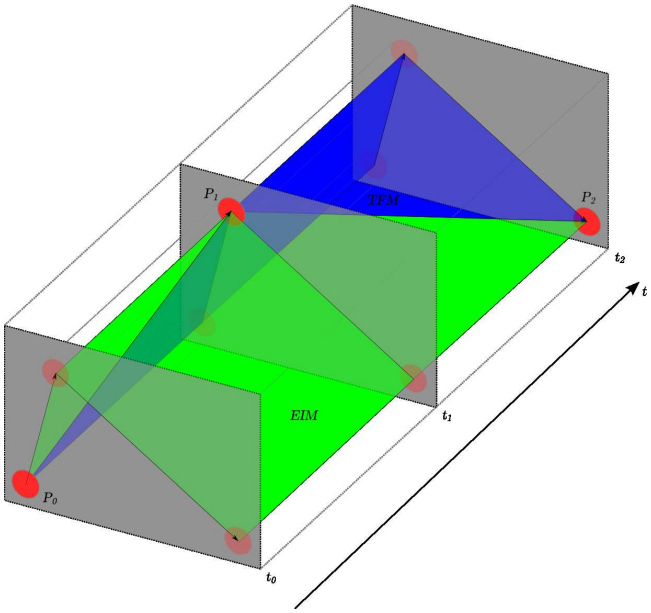


Figure 5.5: Concept of Earliest Information Map and Terminal Fate Map. For EIM*, earliest information is gathered from the temporally "front" search space (green) searching forward, and for* TFM*, it is gathered from the temporally "rear" search space (blue) searching backward. The actual sensor information is sampled at three positions $p_i \in \mathbb{R}^2$ and corresponding times t_i.*

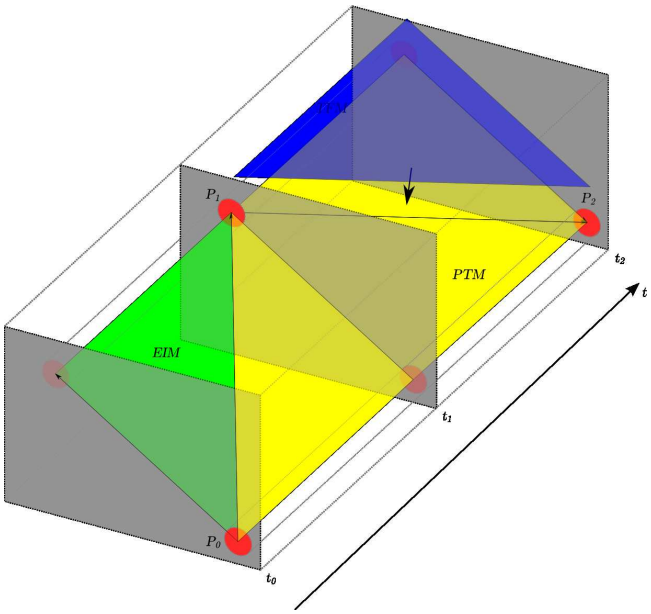


*Figure 5.6: Concept of Pre-Terminal Map (*PTM*, yellow). The sensor information is sampled at three positions $p_1 \in \mathbb{R}^2$ and $p_0 = p_2 \in \mathbb{R}^2$ and times t_i. With destructive sensing, the data in the PTM and TFM differ, since features were already destroyed by the $p_0 \to p_1$ motion.*

5.2.2 Sources of Error in Map-Building

While this concept of an incrementally built and updated map, carrying information created online and without additional cost, theoretically provides the most up-to-date view of the interaction region, it is important to point out its limits and the challenges to consider during actual implementation. The achievable resolution of maps from local sensors is determined by the following factors:

1. *Input/Output:* The foremost limitation pertaining to any map is its achievable precision of data input/output manipulations. How precisely the sensor information can be theoretically localized in space, and determination of the appropriate place to enter it into an existing map, is extensively discussed in Section 5.2.3.

2. *Localization Precision:* Another factor defining the map precision which is internal to the system is dynamic localization (cf. Table 5.1). Practically speaking, this includes the constant component of tool calibration (Section 5.4.1) and the dynamic deformation errors arising from physical tool-patient interaction (Section 5.4.4). The physical operation of the tool itself can introduce errors as well.

3. *Sensor Reliability:* The usefulness of a map implemented in a specific system heavily relies on the underlying sensor and classification reliability (discussed in Chapter 4). *Sensing error* can be both mislocated features (i.e. correct samples at wrong locations) and misrecognized features (i.e. measuring a wrong environment feature at a certain location). The latter is mostly a segmentation or classification error, while the former originates from a combination of registration and position errors (see above).

4. *Sensor Synchronization:* An important requirement for the proposed map – and the system it is implemented in – is to accommodate different sensors into one representation, and correspondingly to degrade gracefully with limited sensor data rates or wrong information. This includes sensor fusion mechanisms within map entries, and temporal and spatial synchronization between different local sensors as well as localizers. The sensor fusion aspect is not considered to be within the scope of this work, but localization is discussed in Section 5.4.

5. *Feature Shifts:* Finally, the usefulness of a map is determined by its ability to reflect environment change. Feature shift includes motion of the environment features – surgically speaking, insufficient fixation of the patient (*patient shift*) and/or moving soft tissue (*tissue shift*) – and is an unfortunate event which should be avoided. Coping with tissue and patient shift (dynamic registration/tracking) is not within the scope of this work.

5.2.3 Map Resolution and Optimizations

When defining the map building function b_M, it becomes immediately clear that for tool-based local sensors, the *tool shape* plays an important role in the localization of sensor information as well. More precisely, the spatial resolution of the local sensor – and thus also that of the local sensor map – is limited by the sensor's (spatially invariant, but defined in sensor coordinates) discrete point spread function (*PSF*) $h_S(x, y, z)$: Sensor information $c = c_p$ from the environment at position p is implicitly convolved with the tool shape S according to its PSF during sampling. Practically speaking this defines the impact of situations where the tool sweeps regions of different subtypes $C \in \{C_0, C_+\}$, while the local sensor returns only a single contact state, thus blurring boundaries between regions in a non-trivial way that cannot be undone with simple morphological operators (Figure 5.7). A simple and very common example of this

Error Type	*Relevance for the presented work*
Registration error is the difference between the expected location of world model features as defined through some registration procedure expressed in system, e.g. robot coordinates, and their actual counterparts' location in the environment. This error is introduced by imprecise localization of environment features or by faulty matching of features between sensed model and real environment.	Registration errors can be both static – caused by imprecise initial matching – and dynamic – caused by gradual or sudden drift of the environment or the tools from their expected position. Optical tracking provides a source of position information independent from the internal robot encoders, and provides static registration – easier than registration procedures based on force-following, but with a certain associated error. Dynamic registration (tracking) was not considered.
Position error in the context of the present work is the difference between expected position p_{exp} – of the tool, robot, or other entities – in robot work space as measured or commanded, and the actual position p_{act} as resulting from the combination of servoing errors, encoder inaccuracies, miscalibrations and misregistration, and deformation from external loads.	Deformation of the robot and tool combination is estimated based on measured forces to reduce the position error. This also helps with sensing error as far as local sensors are concerned. Tool calibration deals with the effects of changing tools or their varying positions relative to the robot.
Tool error is inaccuracy of execution due to operation of the tool outside its modelled operational envelope, and can include effects like the irregular width of saw cuts, tearing due to milling, and other operation-related issues.	This type of error is only considered insofar as the system is designed so as to allow the milling tool to be only used within its range of operation, i.e. not operated under excessive force etc.
Motion planning error is caused by misestimation or ignorance of system parameters, like robot dynamics, tool sizes, tool effectivity etc.	Careful definition of those parameters helps to eliminate these effects. They have been extensively investigated in [Waringo03c].
Logical errors arising from the omission of constraints, as well as special examples like non-optimal placement of the intervention volume relative to the patient.	This is beyond the scope of this consideration as well, but has been investigated in [Waringo03b].
Motion error has a dynamic origin, i.e. inaccuracies in following the planned path caused by operating the robot beyond its maximal speed or acceleration, or originates in planning, i.e. caused by faulty or at least unexpected behaviour of the motion planner at system level.	This type of error can be minimized by conservative path planning (i.e. slow motions) and extensive testing.

Table 5.1: Localization error types

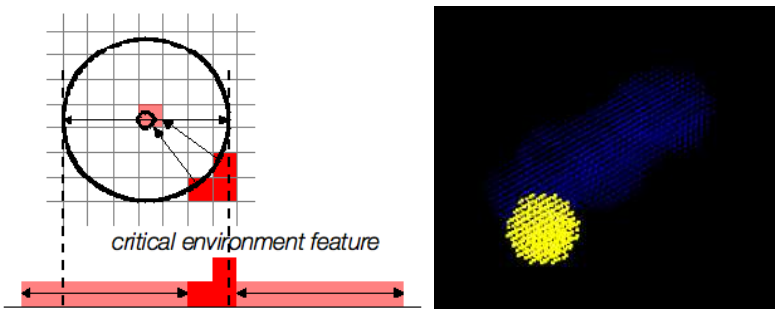


Figure 5.7: Left: Schematic representation of "feature smear" of a single – critical – environment feature (red) due to binary convolution with the local sensor tool shape (black circle). Due to the unknown actual contact area, information about the critical environment feature is distributed into a large region of the map (light red, projected onto a 1D view). Right: Resulting distribution effect after simulated entering of a single-point contact into a 3D map (blue: no contact/C_0, yellow: bone contact/C_B).

situtation is when a miller tool head is entering a bony area from air, e.g. just touches down on the skull bone – in that case the contact changes from C_0 to C_B instantaneously. However, as the actual contact area is unknown, this cannot be easily remedied considering only current sensor information. Since tools can rarely be approximated as points (or Kronecker deltas, for sampling purposes) when e.g. surgical miller heads have diameters of around $5mm$, this places a severe limit on the map resolution and its usefulness. When entering the environment features originating from position p into the map M, the information c_p thus needs to be distributed out over the non-zero tool volume S to correctly represent the limited available knowledge about the situation.

The convolution can be made explicit at two points in time:

- *immediately*, i.e. already when writing the new sensor data c into the map, or
- *delayed*, i.e. when reading from the map later.

Delayed convolution implies that c is entered into M exactly at position p only, usually resulting in a very sparse map. This is more efficient when writing is expensive (map-building $b_M()$ may be time-critical), but the information in the map will be tool-shape- and tool-orientation-dependent – the later convolution-upon-reading becomes more complex since information about those factors needs to be retrieved or reconstructed.

For later use it is desirable, however, to make the information localization within the map independent of the tool or sensor shape. Therefore, immediate convolution spreads out the sensor information already during data entry, i.e. enters data into all the covered map positions. Obviously this makes any later interpretation of M independent of the aforementioned factors.

Furthermore, immediate convolution has another advantage that is difficult to capture in the delayed variant: Additional information already present can be integrated in a straightforward way to improve data localization – most notably motion direction v and application knowledge. As described in the following sections, this makes the convolution related to, but not equivalent to morphological operators like Dilation (because it does not work isometrically, i.e. not identically in every direction).

Therefore, other simple operators (e.g. Shrinking) cannot be used to undo this convolution (to improve precision during map reading) for several reasons. First, the map usually does not represent complete coverage of the environment, often resulting in an asymmetrical extension of regions along the explored boundary; second, any non-trivial classification dominance relation behavior (consistently preferring certain states over others when encountering them over parts of the sensor interaction volume) makes overlapping hard to predict; and third, destructive mapping may cause regions to physically disappear before they can be sampled from each side,

resulting in further asymmetry in extension. Instead, two optimizations can be used to enter better localized data into maps from local sensors online, i.e without the need for additional storage of motion or other extraneous information.

5.2.3.1 Directional Active Shape Modification (Optimization 1)

Any physical tool serving as a data generator for the local sensor will have a non-zero spatial extent, which translates into *active shape* components $S_a(v)$ potentially interacting with the surrounding environment "at the front of the tool" depending on its motion vector v, and complementary *passive shape* components S_p (the set of points comprising the actual tool shape itself, which is assumed to be direction-independent) which are not considered to be in active contact with the environment but take up space in the region of interest, being "the rest" of the tool. Shape S and direction v together allow for exact determination of those subshapes (this is easily extendable to tools like e.g. laser beams or saws). Effectively this approach amounts to the definition of a directionally specialized sensor point spread function.

Since the passive shape S_p does not contribute to the sensor information, but can be assumed to be empty after the tool leaves the current position, it is labeled with $c = C_0$ on each data entry, thus integrating knowledge about the tool geometry.

Ideally, $S = S_a(v) \cup S_p$ would hold for any v. However, note that S_a should extend beyond the physical tool volume in order to allow some potential $c \neq C_0$ information to remain in the map after being swept by S_p. This is consistent with volume-removing tools and destructive mapping, since previously occupied environment locations are marked as removed from the current map this way (cf. Section 4.1.3).

Together, the necessary information to compute the two sub-shapes S_a and S_p for spherical sensor shapes with an active radius of r_{active} (larger than the radius of S itself) can be expressed in the relationships $S_a = f(S, v, r_{active})$ and $S_p = f(S)$.

This – first – optimization integrates knowledge (about the removed volume) that could otherwise only be sampled if the sensor reached the same position again (i.e. is a variant of look-ahead). It is especially useful in conjunction with the following second optimization, e.g. in layered milling.

5.2.3.2 Static Environment Assumption (Optimization 2)

Second, a static environment can be assumed when the workpiece is fixed and non-elastic (e.g. bone). This assumption helps to improve the sensor data localization in a MLS if the sensor (tool) shape has an extent bigger than one rasterized map entry, i.e. is "smearing".

When entering new sensor information $c \neq C_0$ into the map according to the active shape principle as described in the preceding section, the second optimization dictates that this contact information may not be appended to locations where the last appended state was $c = C_0$. This is because there can no longer be an obstacle (Figure 5.8) of any kind in a place where a no-contact was detected or recorded before. Similar restrictions apply to other combinations of hard/soft contact combinations, as described in the following paragraph.

More generally, this optimization can be stated as follows: Given a sensor classification dominance relation set $D_C = \{(C_i > C_j)_k | i \neq j, C_i, C_j \in C, k \in \mathbb{N}\}$ (in which detection of a $c = C_i$ across part of the active shape will suppress detection of any $c = C_j$, cf. Section 4.2.2), new sensor information $c = C_i$ is only to be appended to map entry lists L_p in positions p under the current active shape S_a where the last sensor information is not $c = C_j$ with $(C_i > C_j) \in D_C$. More plainly, if the weaker (less dominant) C_j has already been the most probable classification for this position before, it is impossible that a stronger (more dominant) C_i appears there later (Table 5.2).[3]

[3]Here, the relation set D_C serves as a given logical constraint on the data entry algorithms, and may be

new \ existing	C_0	C_D	C_B	$C_?$
C_0	C_0	C_0	C_0	C_0
C_D	–	C_D	C_D	C_D
C_B	–	–	C_B	C_B
$C_?$	–	–	–	$C_?$

Table 5.2: Classification dominance relation sets for miller-tool-based sensing, seen as transitions of MLS entries (most current MLS entry state across, new sensor information down).

Therefore, the typical state development (or transition sequence) of a single map entry's last known entry over time can be described as

$$C_? \rightarrow C_+ \rightarrow C_0 \tag{5.5}$$

with both transitions being optional. This mirrors what is actually happening to the corresponding environment position over the same period of time.

More specific to the investigated application, it is actually desirable (in terms of the discussed intervention, with $C_+ \supseteq \{C_B, C_D\}$) to have only the following transitions occur both in the environment and the map (notwithstanding the possibility that others may actually occur):

$$C_? \rightarrow C_0 \tag{5.6}$$

$$C_? \rightarrow C_B \rightarrow C_0 \tag{5.7}$$

$$C_? \rightarrow C_D \tag{5.8}$$

Transition 5.6 describes the mapping of a location which is outside of any obstacle from C_+, transitions 5.7 are the destructive mapping of a bony obstacle, and transition 5.8 is the (desired) behavior when encountering a critical region – the dura – which prohibits further exploration of this location. From a local point of view of the considered map entry, either of these transitions constitutes successful intervention execution, while any other transition constitutes either a sensing error or a breach of safety. In the presented work, only the "safety breach" (or its avoidance) interpretation will be tackled; sensing errors are outside the scope of this work and should be dealt with in the context of future work incorporating uncertainty into the maps. These considerations will drive the development of navigation algorithms for local path replanning to evade critical regions.

5.2.4 Map Building Algorithms (Tool-Based, Synchronous/Incremental)

In summary, data entry for map building with tool-based local sensors can be expressed as

$$b_M(c, t, p, S, v) := M' \tag{5.9a}$$

$$M' = M + \big((p, t, c) * S_a(v)\big) + \big((p, C_0, t) * S_p\big) \tag{5.9b}$$

returning a new map M', where c is the sensor information updating the old map M with time stamp t. The binary "+" operation is defined here to append the passed localized environment information to the respective locations in the current map. It is only appended at the positions

different for other than the investigated sensors.

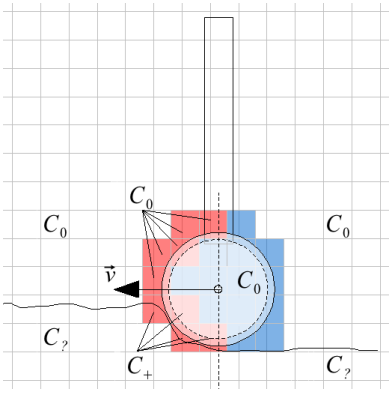


Figure 5.8: Schematic view for entering new local sensor information C_+ (from bottom of environment) into a partly known map by destructive mapping using a miller tool head (new C_+ data only in regions swept by active shape (red), but not filled with C_0; passive shape (blue) generates new C_0 entries).

resulting from the convolution "$*$" of position p with the active shape S_a, which depends on the unit direction of the motion v. The passive shape S_p is labeled C_0.

Furthermore, this update operation is subject to the second optimization, i.e. not to enter sensor information $c_i \in C_i$ where the last map entry was $c_j \in C_j$ with a classification dominance relation set entry $(C_i > C_j) \in D_C$.

5.2.4.1 Active Shape Computation

To simplify active shape computation, a spherical tool is assumed in the described system (although the approach is easily extendable to other shapes), which corresponds to the spherical millers used. This makes mapping under varying attack angles simpler, which are regularly used because of both simple 5D/roll path specialization, and possibly optimization of removal ability through orientation adaptation in future work.

To reduce the online computational load, active shapes are precomputed and stored in a look-up table (*active shape LUT/ASLUT*). Since the active shape data will be used to determine data entry positions in a rasterized map, they themselves are stored in a rasterized representation as well, with a grid constant $\lambda_{ASLUT} = \lambda$ equal to the one of the map (λ). With a spherical tool of non-negligible radius, the active shape $S_a(v)$ is the front (seen from the direction of the motion vector v) half of the outer shell of the miller head, also called *tool-face*.

Each of these shapes is parametrized by inner radius r_S (the radius of the miller head), outer or active radius r_{active} (how far the interaction region extends beyond the physical miller head), and angles ϕ (*inclination* or pitch) and ψ (*azimuth* or direction). The angles are an arbitrary choice for systematic representation of the direction of the motion vector v, relative to the world *up* direction (positive Z axis of the map). The inner sphere of radius r_S is the passive shape S_p, while the active shape $S_a(v)$ is the hollow half-sphere with inner radius r_S, outer radius r_{active}, and points in the direction $(\phi, \psi) = f(v)$ (Figure 5.9).

While the value for r_S is given by the tool, and constant discretizations Δ_ϕ and Δ_ψ for the direction angles are to be chosen according to some optimality principle[4], the attribution

[4]In fact, the later look-up of the required active shape during execution is cheap ($O(1)$), so these can be chosen as high as possible or necessary based on the underlying rasterization. For a regular spherical grid (the scaffold giving the discretization of the active shape look-up table), the largest step sizes between adjacent grid vertices are along the equator. Ignoring aliasing artifacts on a pixel scale, these steps should be chosen to be equal to the active shape voxel grid constant (λ_{ASLUT}). The pixel count n along the equator of a sphere of radius r_S is around $n = 2\pi r_S \frac{1}{\lambda_{ASLUT}}$. Discretizing the overall 360° of the whole circle into n steps guarantees no missed in-between pixels, and when creating the ASLUT based on the discretized directions as keys, the coverage is complete (i.e. a higher discretization would not yield additional entries).

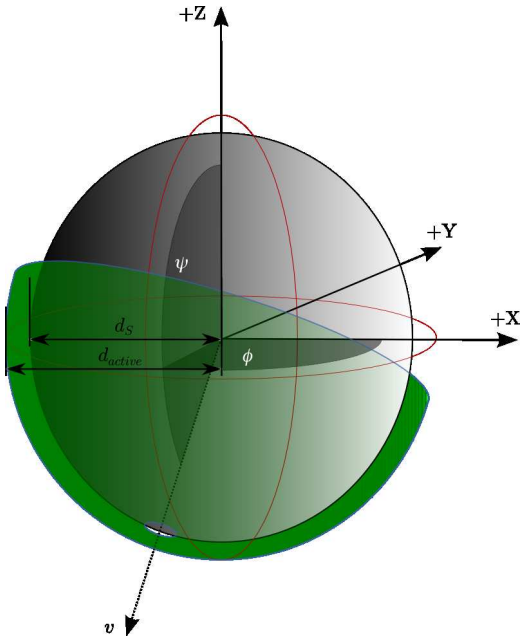


Figure 5.9: Active shape (lower green half-sphere) of a spherical miller tool head, parametrized by the motion vector v (or the corresponding polar angles (ϕ, ψ)), tool radius r_S, and active radius r_{active}.

of an actual value to the active radius r_{active} needs some consideration. While there are many situations where the area contributing to the current sensor reading is variable (depending on tool motion speed, sensor data rate, and possibly others), for reasonably high data rates and fairly small motion speeds the active shape can arguably be set to as small a volume as possible – while still providing some space where $c \neq C_0$ values can be entered into the map. In the limit, for the rasterized representations considered in this work, this comes down to extending the inner-radius-sized passive tool shape by one grid element in each direction, so $r_{active} = r_S + \lambda_{ASLUT}$.

A case that needs special handling occurs when no proper motion vector v can be determined. Since $v_i = p_i - p_{i-1}$, this can occur when either position information was invalid (e.g. beyond the map limits) or non-existing (e.g. during the first mapping step). Setting $v := \overrightarrow{0}$, the active shape chosen is a hollow sphere with inner radius r_S and outer radius r_{active}.

5.2.5 Map Reading Algorithms

Queries to the map that are useful for navigation purposes – i.e. such that are answering questions like *"can the tool move to here?"*, instead of single-map-entry queries – have the form

$$q_M(t, p, S) := c_{class} = \Sigma\left(\tau\left(M\left(p * S\right), t\right)\right) = \Sigma\left(M\left(p * S\right)|_t\right) \tag{5.10}$$

to determine the environment state c_{class} based on data all across the (complete) tool shape S at position p and time t. An application-specific sensor data fusion $\Sigma : \mathbb{R}^n \times C \rightarrow C$ takes place to distill one overall environment feature c_{class} from the selected spatio-temporal map region, which may cover several distinct states.

For the investigated system, this fusion function $\Sigma(V)$ should recognize the hazard level of trespassing on the considered region $V \subset \mathbb{R}^n \times C$. In particular, the returned environment state (the expected contact state) c_{class} should be computed so as to honor dangerous regions at the expense of less dangerous ones, i.e. let single dangerous map entries in V take precedence over (even possibly much larger) less dangerous sub-regions.

For the contact states introduced before, the fusion function Σ shall thus comply to the dominance relation $D_\Sigma \triangleq (C_D > C_B > C_0 > C_?)$ (note that this functional requirement is *not*

equivalent to the phenomenological classification dominance relation set D_C for the investigated sensors, as described in Section 4.2.2). This D_Σ always returns the "most dangerous" known feature from the passed environment, i.e. returns C_D if one is present (to prevent re-visiting dura regions), else returns C_B if one is present (to prevent fast tool-bone collisions), etc.

Together, these methods provide sensor information localization within a map from tool-based local sensors, much improved over the straightforward data insertion into the covered space. Their precision becomes more independent of the original tool shape, especially when there is overlap between successive convolved tool shapes (which helps sharpening the information in the area under scrutiny).

5.3 Independently-Built Maps

This second MLS subtype does not exhibit the intrinsic property of non-intrusive construction like the concurrently-built MLS do. Instead, these independently-built MLS (*indepMLS*) need to be constructed in a separate step. This is due to the use of the more general class of local sensors – non-tool-based – which do not necessarily provide data concurrently with the intervention. In case of intra-operative map building, this implies an interruption of the "standard" operation workflow. Therefore, the use of independently-built MLS needs to be justified on grounds of superiority of the specific MLS (based on specific local sensors) as compared to conventional maps. For 3D ultrasound measurements, this is given by the A-mode sensors' high precision and non-invasiveness.

5.3.1 3D Ultrasound Measurements (Tool-Independent)

Addressing the most important drawbacks of the CT-based intervention planning approach (time requirements, radiation exposure, and obtrusive sensing procedure), a robot-based 3D ultrasound scanning method to generate intra-operative global image data is presented. The method produces data equivalent to that of traditional CT but is less costly and can be used to perform both intervention planning and registration. The patient is not exposed to X-ray radiation, and since the presented method is markerless – it does not rely on (implanted) pins – it can alleviate the need for time-consuming pre-operative preparation and imaging procedures. As registration between the patient and the corresponding image data is implicit in the presented scheme (imaging is performed just before the time of execution), planning and execution take place directly in the image data without any need for external registration methods (except between robot and image data).

5.3.1.1 Design Requirements

Therefore, the design requirements regarding a robot-based 3D ultrasound map can be stated as follows: Discarding CT in favor of US imaging for orthopedic interventions makes high-precision measurements necessary, with distance and preferably bone thickness being sampled with a precision comparable to the approx. $0.4mm$ standard set by conventional CT. Furthermore, because the US probe is a local sensor (it does not provide global positioning information) a means of locating the samples relative to each other is necessary with a spatial and temporal precision resulting in volume reconstruction better than or comparable to CT. Finally, the volume needs to be registered to the robot precisely enough to be useful for navigation with additional local sensor support, i.e. the transformation from planning data to the robot coordinate system must be established.

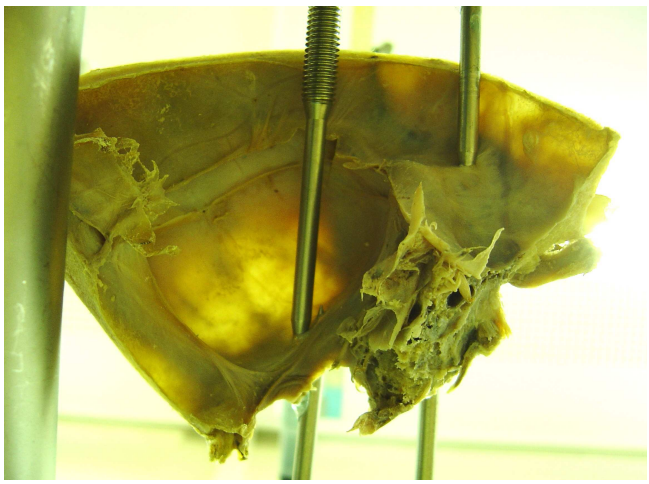


Figure 5.10: Human skull specimen (in laboratory experiment, seen from below). The unevenness of the lower profile is clearly visible, as is the highly variable thickness of the bone itself, evident from the wide range of translucency.

5.3.1.2 Overall approach

With the presented method, several implicitly registered image modalities are generated for navigation. First, a skull surface representation is generated by manual sampling of surface points with a hand-held IR pointer, after which a skull representation is created by scanning the skull with a robot-held A-mode US probe, returning both outer and inner skull boundaries. Both representations can be used in addition to or as a replacement for a CT scan (the latter approach of CT-free interventions is investigated here). Only acquiring the outer profile – e.g. with the IR pointer approach – is not sufficient, as the inner skull has highly uneven thickness (Figure 5.10). The acquisition of both profiles automatically registers the robot with the patient. Furthermore it allows to compute an optimal implant position and milling paths, as well as to perform the milling intervention without additional registration.

To achieve this, several transformations – between the robot, optical tracking system, and IR pointers – need to be established first (Section 5.3.1.3). Robot-based scanning relies on a manually defined path that has to be smoothed (Section 5.3.1.4) before further processing (surface normals determination, Section 5.3.1.5) takes place. Finally, path planning for the robot may bring about kinematic problems that need to be addressed before actual execution (Section 5.3.1.6). Sampling the two strongest echoes during the scan yields two sets of positions which enter the 3D ultrasound volume reconstruction (Section 5.3.1.8).

As described later in Section 6.1.2, the implant position can be optimized within this volume. A last step then computes the robot milling path before the intervention execution.

5.3.1.3 Registration and Scan Path Definition

The registration relationships between the robot, optical tracking system (OTS), hand-held pointers, skull surface, and the US scan path for the presented system are shown in Figure 5.12, and schematically in Figure 5.11. The rigid tool combination including miller and US probe forms one registration entity which needs to be registered to the actual patient in the operating room (OR). Since imaging data originates directly from the current patient situation (contrary to CT data which would need to be sampled separately beforehand), patient and bone representation can be identified as one entity which does not need any additional registration procedures. The milling volume (the implant bed) needs to be positioned within this representation, e.g. through a position optimization procedure as described in [Waringo03b].

The user then samples a cloud (a non-sequential set) of skull surface points $K = \{{}^{OTS}k_{outer,i}\}$

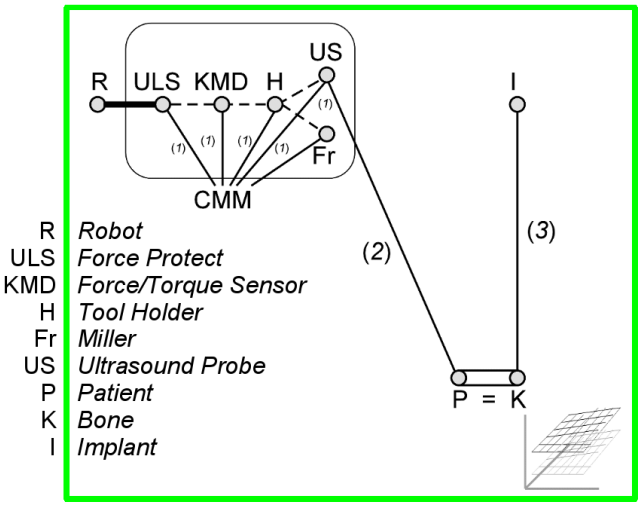


Figure 5.11: Registration entity relationships in the robot-based 3D ultrasound acquisition. The patient (P) *and the bone* (K) *entities can be assumed to be identical, since there is no patient motion after ultrasound bone imaging takes place. Transformations* (1) *are determined by the calibration motions (Section 5.4.1),* (2) *by the ultrasound scanning procedure (Section 5.3.1.7),* (3) *by the implant position optimization (Section 6.1.2.1; graph based on the notation in [Lea95]).*

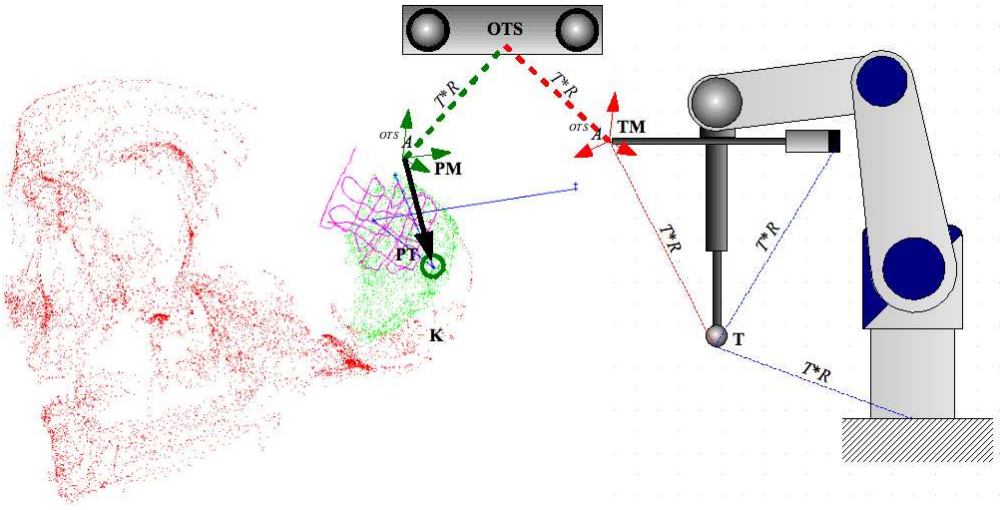


Figure 5.12: Registrations (3D: only translations T, 6D: translations T and rotations R) between robot and tool and US probe (tool tip T, *tool infrared marker* TM*), the optical tracking system (*OTS*), skull surface and scan path* K *demonstrated by hand-held pointer (pointer tip* PT, *pointer infrared marker* PM*) (adapted from [Stolka07a]).*

Figure 5.13: User demonstrating the skull surface with an optical tracking pointer, used for both surface sampling or robot scan path definition.

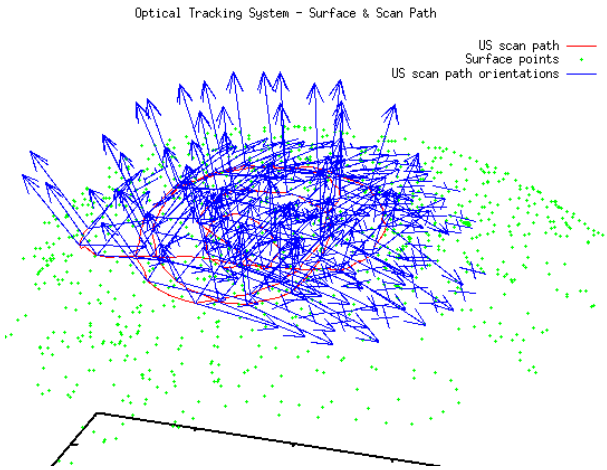


Figure 5.14: Surface normals $p_\perp$ (blue arrows) computed at scan path points $p \in P$ (red line) from a set S of surface points from the skull surface (green points) and the scan path P itself [Stolka07a].

by sliding the pointer over the exposed skull, thus creating a surface representation which is sampled with the OTS, and thus already registered with the robot via the previously determined transformation ${}^{OTS}M_T$ (with T being the reference coordinate system associated with the tool tip position before execution starts; cf. Figure 5.13 and Section 5.4.3):

$${}^{T}k_{outer,i} = \left({}^{OTS}M_T\right)^{-1} \cdot {}^{OTS}k_{outer,i}$$

In a second step, the pointer is used to define the 3D positions $P = \{{}^{T}p_i\}$ making up the scan path on the skull surface to be followed by the robot-held US probe. The user performs this step manually to allow explicit consideration or omission of problematic regions with strong curvature or sensitive structures. However, the US probe will need to be oriented almost perfectly perpendicular to the surface to be scanned in order to receive satisfying US echoes later. This is especially important for correct delineation of the inner skull boundary. Therefore, the sequence of acquired 3D points needs to be processed prior to path execution by path smoothing and surface normal determination procedures.

For suitable, lattice-like scan paths with neighboring parallel passes at distances of approx. $10mm$ over an area of approx. $(50mm)^2$, this results in a scan path length of approx. $600mm$. Now the 3D points acquired must be processed before path execution by smoothing and surface normal determination procedures.

5.3.1.4 Scan Path Smoothing

The manually defined path P often contains agglomerations of path points near the beginning and the end of the path, as well as in the vicinity of surface irregularities which caused the traversing pointer to get stuck temporarily. The result are irregular scanning speeds and jerky motion of the robot, both due to the varying spatial density of path points, and an excessive number of path points flooding the trajectory generator.

In order to allow faster and less jerky scanning, the scan path is straightened and its support point set minimized by successively removing points until the resulting path deviation exceeds

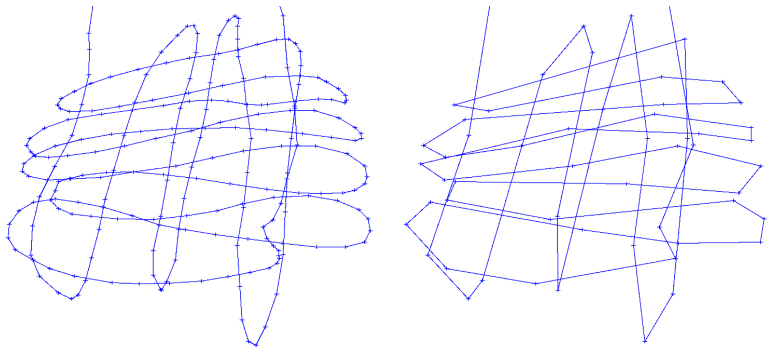


Figure 5.15: Initial ultrasound scan path P (left; 307 path points), straightened path P' after point reduction (right; 90 path points) with a maximum allowed deviation of 1mm between original and straightened path (from [Waringo06]).

an application-specific threshold. In a first step, clusters of points being closer together than a given threshold $d_{min} = 4mm$ in the current setup are reduced to only single points, thus pruning the afore-mentioned agglomerations.

In a second, more sophisticated step, the path is straightened according to a more application-specific criterion. Here, a variety of distance metrics can be chosen from to describe the remaining similarity of a straightened path to the original input path.

The algorithm for removing path points is described in detail in [Waringo06]. Figure 5.15 shows an initial US scan path P and the result of path straightening $P' \subset P$. The runtime of the smoothing step for comparable paths is on the order of a few seconds.[5]

5.3.1.5 Surface Normals Determination

The US probe will later need to be oriented almost perfectly perpendicular to the scanned surface in order to receive satisfying US echoes, especially for correct inner skull boundary samples. In the next step, surface normals over a local sample neighborhood are determined from the scan path P' and surface points set $K_{surface}$ which approximate a possibly non-planar surface.

First, the point $p \in P'$ (taken from the scan path) is needed for which the normal vector should be determined. It is not required for p to be contained in the surface defined by the point cloud $K_{surface}$, although ideally it should be close to it. The algorithm can be subdivided into three steps. First, the space containing $Q = P' \cup K_{surface}$ is hashed allowing for fast access to neighboring points. The next two steps consist of finding relevant points $q_i \in Q$ in the proximity of p (defined by a search radius of approx. $8mm$) and computing the normal vector $p_\perp$ on the surface defined by Q. These steps must be repeated for each normal vector (Figure 5.14). Every normal $p_\perp$ corresponds to one path point p, which is modified or "specialized" to have two rotations fixed, resulting in the tool axis (the miller shaft, aligned with the X-axis of the tool coordinate system) pointing straight away from the surface and thus creating a new path P'_{5D}.

Finding the normalized normal vector from n samples has an asymptotic complexity $O(n)$. Effective calculation times are approx. $1min$ for the data sets encountered in the experiments (representative for interventional purposes; cf. Figure 5.14). More details on this algorithm can be found in [Stolka07a].

[5]The terms "path straightening" and "path smoothing" are used interchangingly here, since the actual procedure involves removing superfluous points – thus straightening the path itself – while the later path execution is smoothed.

5.3.1.6 Scan Path Specialization

The procedure described so far yields a 5D path P'_{5D} – a sequence of 3D positions with two rotational degrees of freedom (*DOF*) fixed. This arrangement leaves each point with one DOF open; in the presented system this is the roll parameter, i.e. rotation around the long tool axis. Since the US probe operates in A-mode, each single shot is rotationally invariant around this DOF, which can therefore be set to any kinematically valid value between 0° and 360°.

By arbitrarily selecting one orientation around the US sensor axis, the scan path can be followed, guiding the probe perpendicularly to the skull surface (cf. Figure 5.17). However, since robots usually have limited DOF and joint ranges, not every orientation results in a valid executable path. In the presented case, the robot has six non-redundant DOF, and following the skull's curvature with the robot-held probe often leads to kinematic singularities and joint limit problems, compromising safety and leading to abortion of the current path execution. Therefore, the remaining DOF is used to alleviate those problems. (This is related to approaches for robot placement optimization, leveraging unconstrained DOF in task space to select a position for the robot in the OR around the patient, e.g. in [Engel03] or [Konietschke08].)

After registration of the robot, the patient, and the 5D path, a valid 6D path is searched from a set of paths generated with different roll values that are held constant over the whole path. First, a finite working set Θ of roll values is selected from a range that is (empirically) believed to contain values that will result in valid paths later. Second, all paths P'_ϑ corresponding to roll values $\vartheta \in \Theta$ are generated by specializing every $p \in P'_{5D}$ and interpolating densely with respect to Euclidean distance and orientation. Finally, each P'_ϑ is validated by applying the robot inverse kinematics to each interpolated path point and checking for joint limits, singularities, and discontinuities (Figure 5.16)[6]. In the current implementation, the first completely traversable path $P'_{\vartheta'}$ is selected for execution. If no kinematically admissible path is found, the robot base location must be changed.[7]

5.3.1.7 Robot-Based Ultrasound Scan Execution

To ensure stable contact between US probe and skull even when the initial scan path definition is sub-optimal due to calibration errors, the scanning is performed under force-based contact control (*force-based servoing* or *hybrid position/force control*; Figure 5.17). To achieve this, the robot is subjected to a position P-law control with force target value $F_{target} = 10N$ along the sensor/tool frame Z axis (Figures 5.19, 5.20). However, this force-based scheme is only activated upon contact and only along the tool Z axis, turning simple path execution into hybrid switched-system control enabling to switch from simple trajectory following to sensor-guided motion – effectively a *sensor-guarded controller* [Kroeger10].

The resulting deviations Δz relative to the initial scan path do not show up in the echo positions – they are cancelled out by the smaller US delay line thickness $d'_{delay} = d_{delay} - \Delta z$ when compressed (up to the measurement error; Figure 5.21). Note that the executed scan path (describing the expected position of the lower boundary of the US probe delay line) lies

[6] For safety reasons, this takes place on the robot controller, using the same inverse kinematics algorithm as during later path execution, thus avoiding possible subtle model differences between this simulation and execution.

[7] Note that this approach of accepting the "first completely traversable path" relies on two assumptions: First, that the respective path has been interpolated densely enough to justify the assumption that the intermediate continuous path segments will not violate safety requirements or kinematic constraints, e.g. by containing singularities; and second, that the selected path is not significantly altered during execution after selection. However, the later force-based contact control ensuring proper probe-skull contact introduces (albeit very small) non-predictable axial displacements of the probe. Should this lead to closer approaches to singularities, the resulting higher joint speeds will be limited by the joint speed control (cf. Section 6.1.2.3.2); but joint limit errors result in intra-operative procedure abortion.

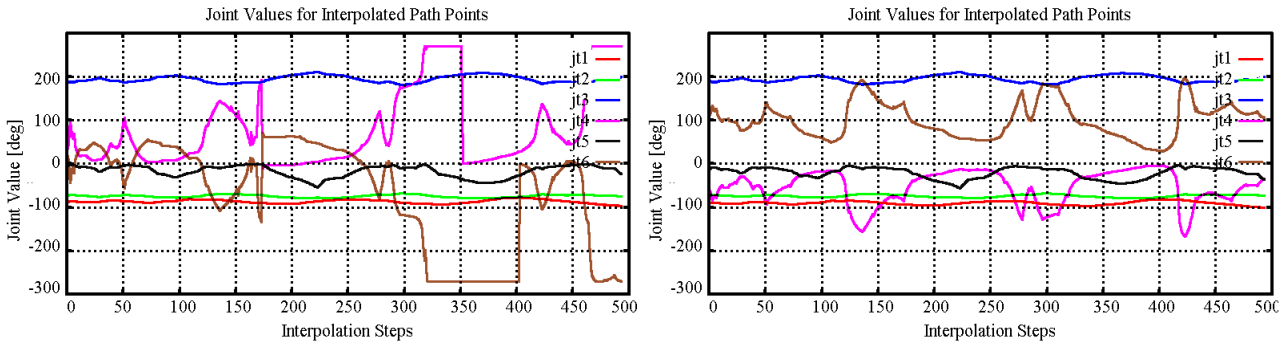


Figure 5.16: Joint trajectories (from inverse kinematic transformation of the scan path) for a given robot scan path and a fixed robot position in the OR. While the first trajectory is invalid (for roll value $\vartheta = 0°$, joints 4, 6 exceed their limits), the second one is accepted as a valid $P'_{\vartheta'}$ (with $\vartheta = +10°$). Note the very fast joint value changes for joints 4 and 6 in the first trajectory, hinting at wrist or α_5 singularities, and also the joint angle saturations for the same joints [Stolka07a].

Figure 5.17: Robot-based ultrasound scan. The ultrasound probe is guided perpendicularly along the skull surface under force-based pressure control.

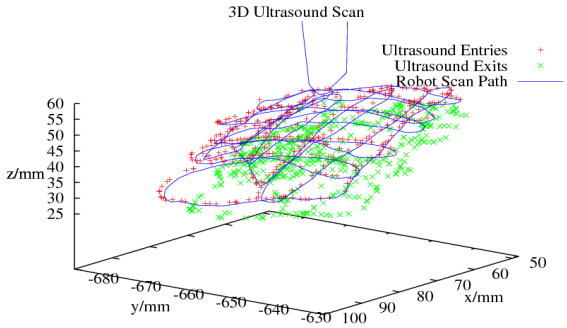


Figure 5.18: Ultrasound scan, showing two sequences of surface points (red: outer skull surface, green: inner skull surface) and the scan path as followed by the robot-held probe (blue). In general, the scan path lies slightly below the outer surface points; this is due to the force-based contact control compressing the delay line to ensure probe-skull contact [Stolka08].

partly below the outer skull surface (Figure 5.18); there the robot pressed the probe deeper into the skull to achieve the contact pressure control target value.

The final scan path execution with concurrent ultrasound sampling returns a sequence of US A-scans, with the single upper and lower skull boundaries detectable by thresholding the filtered radio-frequency signal. For better results, matched filtering is performed with coded excitation chirp signals. Instead of single ultrasound pings, a modulated waveform is emitted and cross-correlated with its echoes, thus improving detection in the face of bad signal-to-noise ratio (cf. Section 4.3.3).

This process creates two sequences ${}^{T}p_{outer,i}$, ${}^{T}p_{inner,i}$ of skull surface points – on the outer and inner surfaces – whose coordinates are computed from the robot pose (more specifically, the US tool tip at ${}^{T}M_{US}$ relative to its reference position T) at each sampling instant and the ultrasound probe's depth measurements $({}^{US}d_{outer,i}, {}^{US}d_{inner,i})$:

$$ {}^{T}p_{outer,i} = {}^{T}M_{US} \cdot {}^{US}d_{outer,i} = {}^{T}M_{US} \cdot \begin{pmatrix} 0 \\ 0 \\ {}^{US}d_{outer,i} \end{pmatrix} $$

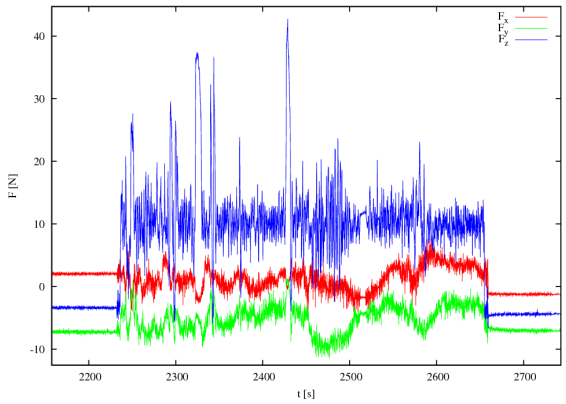


Figure 5.19: Graph of external forces occurring during ultrasound scan, with the axial force component F_z (blue) contact-force-controlled to $10N$.

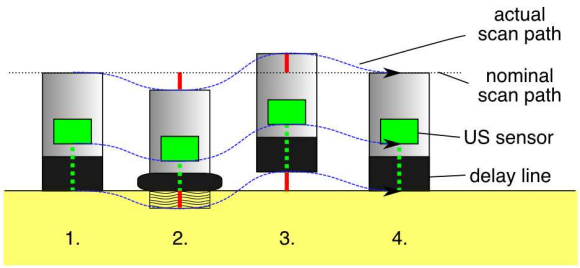


Figure 5.21: Ultrasound probe in different contact pressure situations, following an uneven actual scan path (dotted blue line), where the nominal scan path should be a straight trajectory (straight black line). The measured distances (green dotted lines) and the position offsets (red lines) always add up to a constant value as long as there is contact (i.e. not in Situation 3). Force-based pressure control straightens the actual scan path.

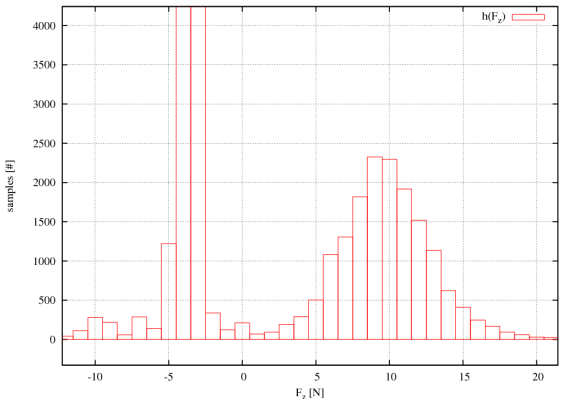


Figure 5.20: Histogram of occurring axial forces during contact-force-controlled ultrasound scan (cf. Figure 5.19). The peak at $-3.5N$ – instead of $0N$ – is due to non-optimal calibration.

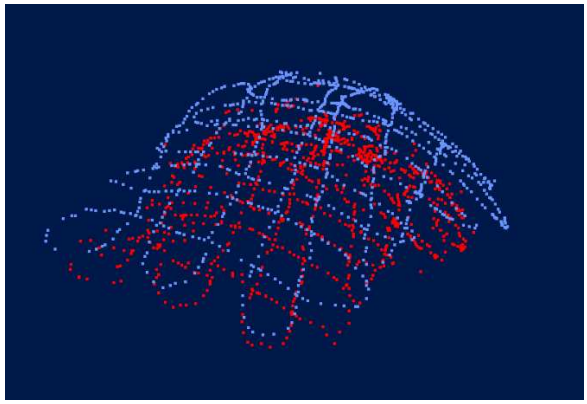


Figure 5.22: Ultrasound scan result, showing two sequences of surface points (blue: outer skull surface, red: inner skull surface) (image courtesy of S. Tretbar).

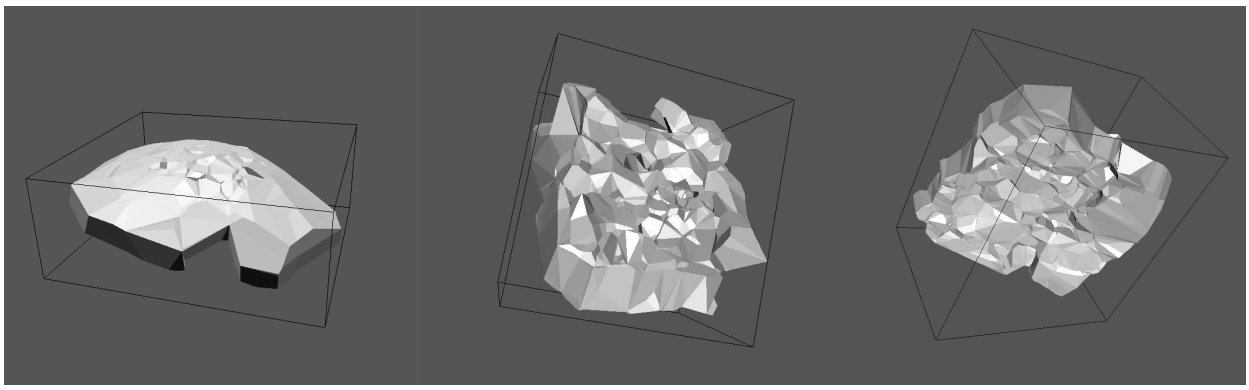


Figure 5.23: Results of skull surface reconstruction based on ultrasound scanning. While the measurements of the upper surface (left) are reliably localized, the lower surface has a lower signal-to-noise ratio that initially resulted in a more crumpled surface reconstruction (center). However, with proper perpendicular orientation of the probe w.r.t. the skull surface, the lower surface could be reconstructed with a higher quality (right) [Stolka08].

(analogously for the inner points). The sequences can also be interpreted as two 2.5D skull boundary representations (Figure 5.22). The inner boundary measurements are less reliable, in the sense that they are not always available (when the received echo energy was not enough to push the echo-chirp correlation value above a pre-determined classification threshold). In this case, only the outer boundary position is returned.

For the previously described lattice-like scan paths and a robot scanning speed of approx. $5mm/s$, the full scan takes around two minutes.

5.3.1.8 3D Ultrasound Volume Reconstruction

The robot-based scan exhibits only low noise of the positions of the lower US boundary. In the next post-processing step, both boundaries are combined into one cloud and serve to create a non-convex, "water-tight" hull with the Powercrust triangulating surface-generation algorithm [Amenta01] (Figure 5.23).[8]

Once this is achieved, the hull's surface representation is rastered into a voxelspace and filled up to reconstruct the skull bone volume, with the final result being an occupancy grid map of the ROI. This whole process generates data structures which are registered with the patient in robot coordinates at all times. The runtime of these steps is negligible.

5.4 Localization

Current approaches to establish or improve precision of registration and execution in (surgical) robotics, especially milling applications, mostly aim at determining an exact relationship between intervention area and the robot. However, this is subject to quality deterioration due to positional shifts and deformations under load. Therefore, in this work two aspects related to more accurate definition and measurement of the actual tool position by optical and force measurements are described: registration and deformation handling. These both allow static (one-time) and dynamic (ongoing) registration and estimate tool and other deformations.

[8]It is worth noticing how the lower signal-to-noise ratio of the lower surface measurements results in noisy position data there (an effect within the ultrasound sensor), which in turn extends the hull downwards after reconstruction. This might be relevant for the later implant position optimization – in future work, the optimization could deal with surface representations containing uncertainty. Another area which is affected by this uncertainty is individual path planning. Here, ensuring safety in spite of the uneven lower boundary representation is delegated to the local navigation based maps from local sensors, cf. Section 6.2).

Detrimental influences to precision include inaccuracies during all of the process phases – sensing, planning, registration, positioning, motion, and tool errors (Section 5.2.2). When using an industrial-grade, approved-for-medical-use robot, and when planning on data acquired from sources like CT, the two largest remaining single sources of error are the precision of the pre- and intraoperative static and dynamic registration and inaccuracies caused by the used tool itself.

The well-known problem of registration deals with establishing a known spatial relationship between data and patient, so that intra-operative localization of tools etc. corresponds to their actual position [Lea95]. First and foremost, this requires precise knowledge of the geometry of the tool which is attached to the robot, which in turn necessitates a calibration step (Section 5.4.1). Furthermore, this requires a precise method to determine the positions of the patient, the tools, and the robot, if applicable (Sections 5.4.2, 5.4.3). However, the tools and the robot can be subject to external forces and moments as well, distorting its supposedly fixed geometry and deteriorating the quality of the intervention result, because the position of the tool is then only known with insufficient precision (Section 5.2.3.2). Especially when using intraoperative data from local sensors for map-building and navigation, exact location data is highly important. Therefore, in the following, approaches to measure and improve the precision of the measured tool position are described in order to make the outcome of high-precision operations like milling more predictable.

5.4.1 Mechanical Tool Calibration

In the preceding sections, the exact transformation to the tool tip T reference position ${}^{rob}M_T$ in robot coordinates was assumed to be perfectly known. However, after tool changes, maintenance, or due to wear or even imprecise CAD modelling this location can change. In this case, even perfect registration between robot and environment coordinates will result in tool paths located off their intended place, or tool rotations occurring not within the defined tool tip, but instead in some arbitrary, nearby rotation center. A tool calibration procedure is necessary to guarantee minimization of this error type.

5.4.1.1 Requirements

It is desirable to perform this calibration with as little extrinsic knowledge as possible, so the procedure should be designed to require minimal information about calibration body geometry and allow large tolerances as to its surface geometry. The calibration errors to be minimized can be assumed to be small three-dimensional translations (up to $10mm$, as the tool type and therefore its approximate position should be known within this margin, although this is only an implementation restriction), i.e. no rotation errors need to be considered. Furthermore, the procedure shall run automatically and repeatedly without operator intervention. Finally, the usual simplifying assumption of a spherical tool head can be made. In particular, calibration shall need neither tool tip diameter nor the exact position or shape of the rigid calibration body. Together, these desiderata lead to the following calibration procedure.

5.4.1.2 Approach

A steel post rigidly attached to the robot base area serves as the calibration body base, equipped with a surgical steel outrigger arm whose level top plane of around $10 \times 10mm^2$ is the actual calibration body itself (Figure 5.24). While the approximate position of this body is known in robot (and uncalibrated tool) coordinates, it needs to be known only precisely enough for the tool tip to contact it during the calibration motions.

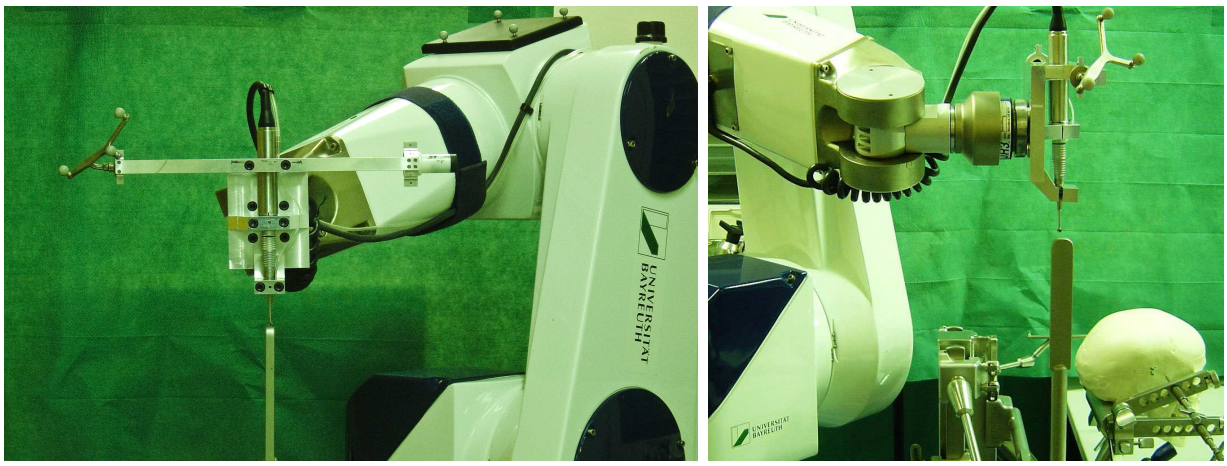


Figure 5.24: Tool calibration setup, with robot-held tool approaching the calibration post.

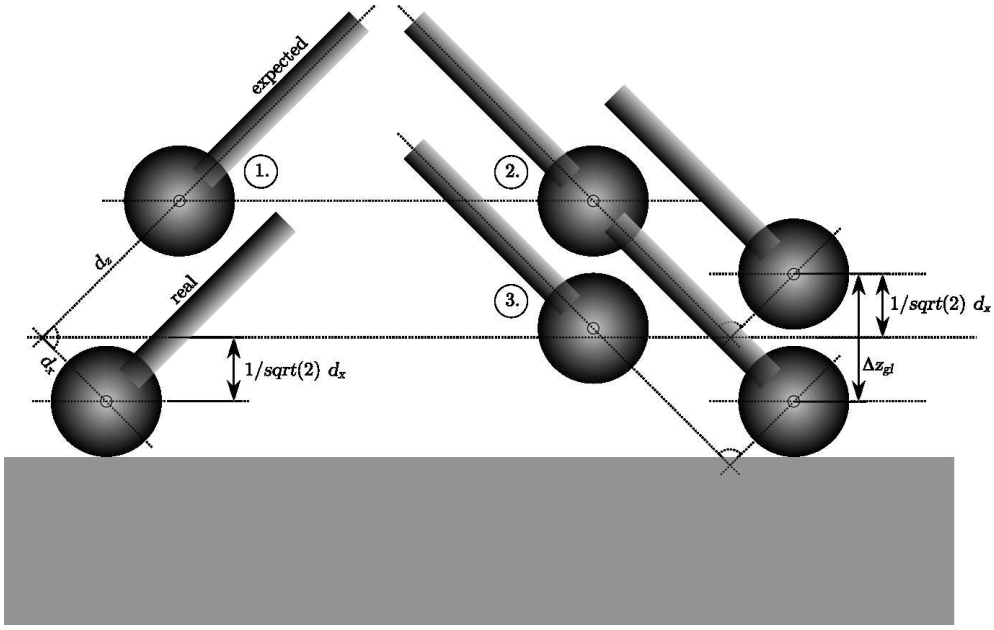


Figure 5.25: Tool calibration procedure – determination of lateral offset d_x (d_y not shown). Two approaches with a $-45°/+45°$ offset towards a rigid obstacle (calibration body, at the bottom in grey) yield a distance difference Δz_{gl}, which helps to determine the respective lateral offset.

5.4.1.2.1 Miller Tool Tip Calibration The calibration itself consists of three sets of two force-controlled approaches along the global Z ("downwards") axis to the calibration body with the approximately known tool tip. Each set eliminates the unknown offset along one tool axis. The first two sets of approaches determine the X and Y offsets (by exploiting the geometrical relationships shown as an example for the X offset in Figure 5.25). Position (1) shows the tool in contact with the calibration body under an angle of $+45°$ around the Y axis – while the physical tool (labeled "real" in the figure) touches the body, its expected position (labeled "expected") as measured by the robot encoders is above and off to the side by offsets of d_z and d_x. A second approach under an attack angle of $-45°$ yields another contact position height (3), off by

$$\Delta z_{gl} = 2\frac{1}{sqrt(2)}d_x$$

from the original starting location (2) as shown. The offset d_y is determined analogously (with rotations around the X axis). Finally, after incorporation of those lateral offsets into an improved intermediate calibration, two more approaches under attack angles 0° (1) and 45° (2)

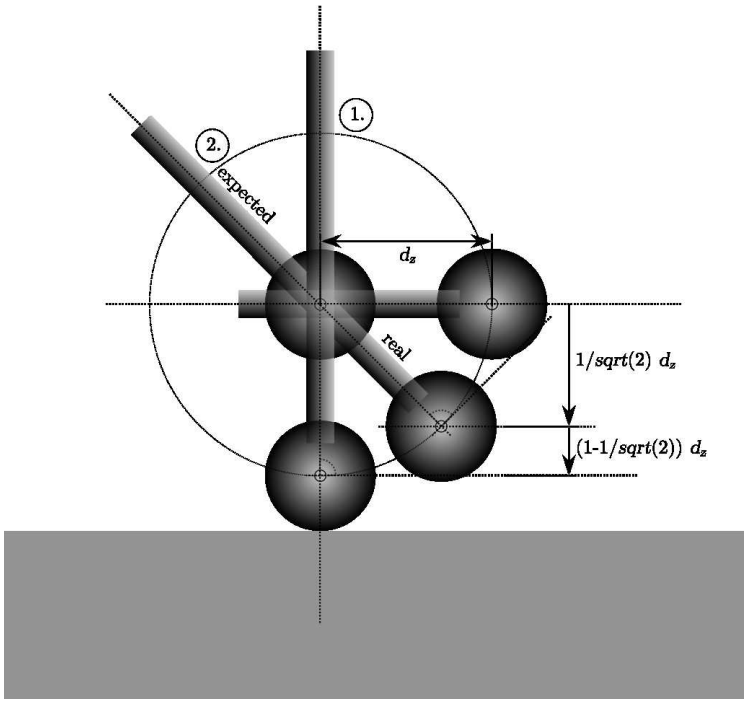


Figure 5.26: Tool calibration procedure – determination of axial offset d_z. After completion of lateral calibration, two approaches with a 45° offset towards a rigid obstacle yield a distance difference of Δz_{gl}.

(Figure 5.26) result in a height difference of

$$\Delta z_{gl} = \left(1 - \frac{1}{sqrt(2)}\right) d_z$$

which yields the d_z offset component for complete tool calibration.

5.4.1.2.2 Ultrasound Sensor Calibration A similar approach is used in the calibration of the robot-held A-mode ultrasound probe. Here, the variabilities that need to be accounted for are more limited than with the miller tool. In particular, the fixation of the probe body is such that it allows only linear translations along the probe's Z axis, i.e. any changes to the setup (like removal of the probe for sterilization etc.) can result in modifications of only one degree of freedom. Calibration pertaining to the other DOF is performed only once (not described here) and need not be changed any more.

The position ${}^{rob}p_{calib} = \begin{pmatrix} x & y & z \end{pmatrix}^T$ of the calibration body in robot coordinates is known after the successful miller tool calibration. A final force-controlled approach with the US probe downwards along the (robot and US probe) Z axis toward the calibration body yields a new contact position ${}^{rob}p'_{calib} = \begin{pmatrix} x & y & z' \end{pmatrix}^T$ which only differs in the z' coordinate from the previously known ${}^{rob}p_{calib}$ (since the positions are returned by the robot via its own encoders, which are also used for all the mentioned approaches with constant x and y coordinates). The difference $z - z'$ is added to the tool transformation's Z coordinate (transformation describing the offset between robot end-effector flange and actual tool tip), thus finalizing the calibration. Given that the depth offset between the lower boundary of the ultrasound probe's flexible delay line and the inlying actual sensor array is known beforehand from the probe design's CAD model, the exact position of ultrasound measurement pairs ${}^{US}(d_{inner}, d_{outer})$ w.r.t. an independently established reference position T can be computed now.

5.4.2 Registration by Force-Following

One very simple way of inducing a static and rigid registration relationship between the planned path P and its execution site in robot coordinates is the definition of a local coordinate system (frame) ${}^{rob}M_P$ by providing three or more distinct points, so that ${}^{rob}P = {}^{rob}M_P \cdot P$. This can be achieved e.g. by a force-following procedure (guiding the robot manually by imposing forces on the tool [Stolka03]; Figure 5.27).

5.4.2.1 Requirements

Because the robot is registered directly to the patient in the OR, without any intermediate transformations related to pins or other localization methods, positioning the robot by force-following has the potential to achieve the maximum possible registration accuracy. Therefore, in order to use the robot directly in such a scheme, the following drawbacks need to be addressed:

- The system needs to be "transparent" in use, i.e. offer an intuitive handling behaviour.
- It needs to be non-threatening and pose no danger to the humans around it.
- Point-definition times needs to be as short as possible.

5.4.2.2 Approach

In the following, a novel hybrid control algorithm is presented which provides the needed precision in small scale movements while allowing for fast and intuitive large scale translations.

It can be assumed that ease-of-use of a force-following robot is connected with intuitiveness, which in turn is determined by clear physical analogies. Therefore, a control scheme is derived based on Newtonian physics (therefore called Newton Control or N control) to be used as a positioning aid in the envisioned applications.

Inertia, acceleration and friction play an important role for modeling a moving mass. As the motion of a body with mass m and without external forces F continues infinitely, its vector v remains unchanged according to the first Newtonian law. Acceleration a is given by the second law $F = m \cdot a$. Friction force F_{frict} is assumed as inversely linear in the exerted force F according to a friction coefficient f_{frict}.

Having defined the virtual mass $m > 0$ and a friction coefficient $f_{frict}, 0 \leq f_{frict} < 1$, the resulting control equation is as follows:

$$v_{comp}(t) = v_{comp}(t-1) + \left(\frac{F_{user}(t)}{m} - \frac{f_{frict} \cdot v_{comp}(t)^2}{2\Delta s} \right) \Delta t$$

The computed speed $v_{comp}(t)$ is defined with reference to the speed from the last control cycle $v_{comp}(t-1)$, thereby providing inertia. The force $F_{user}()$ exerted by the user is sampled at discrete time points and assumed constant over the past control cycle, and through scaling with the mass m, the user-effected acceleration is computed. Recoursing to kinetic energy, the last term defines the negative acceleration owing to friction effects over a path length of Δs. The duration Δt is the control cycle time which is constant due to the real-time operating system on the robot controller unit and amounts to $\Delta t = 16ms$.

The N controller can be transformed into an I (integral) controller when neglecting friction:

$$v_{comp}(t) = v_{comp}(t-1) + \left(\frac{1}{m} \cdot \frac{\Delta t}{T_N} F(t)\right) \tag{5.11a}$$

$$= \left(\frac{V_I}{T_N} \sum_{\tau=0}^{t-1} F(\tau)\Delta t\right) + \left(\frac{V_I}{T_N} F(t)\Delta t\right) \tag{5.11b}$$

$$= \frac{V_I}{T_N} \sum_{\tau=0}^{t} F(\tau)\Delta t \tag{5.11c}$$

Having set the gain factor $V_I = m-1$ and the follow-up time $T_N = 1$, the N control equation is equal to the discretized I control equation.

While this N control scheme proved to be appropriate for large scale and high speed motions in initial experiments [Stolka02], it is not suited for high precision tasks. There, the human operator grips the tool tighter, resulting in higher stiffness and possible oscillations. As the user tries to dampen the oscillation by gripping even tighter, the resonance frequency goes up, and the system becomes unstable. Even when the user does not exert high forces, the system is very sensitive during exact positioning, effectively rendering high precision impossible.

Thus, a control scheme was developed combining both the convenient large scale behavior of N control with the high precision of P control without including their respective drawbacks. To do so, a hybrid controller switches over between P control at low speeds to N control at higher speeds. Experiments with users of different experience levels with this controller showed that a useful threshold speed for switching is about $v_{thresh} = 10mm/sec$ for the task described later. As a sudden control change between P and N control is undesirable, interpolation between both was implemented, governed by the current absolute speed $v_{abs} = |v| = |v(t)|$, allowing for a gradual, sliding mixing of both controllers' effects according to the factor

$$\frac{N}{P} = r(|v|) := \begin{cases} \left(\frac{|v|}{v_{thresh}}\right)^{\frac{1}{\alpha}} & \text{if} \quad |v| < v_{thresh} \\ 1 & \text{else} \end{cases}$$

The exponential factor $\alpha > 0$ determines the slope behavior of the mixing (linear, quadratic, ...). User interviews showed that linear interpolation ($\alpha = 1$) is the easiest to master for most users. The P and N controllers are giving two parallel sets of speed components, $v_{Pcomp}(t)$ and $v_{Ncomp}(t)$. The final speed is a linear combination of both:

$$v_{N/Pcomp}(t) = (1 - r(|v|)) \cdot v_{Pcomp}(t) + r(|v|) \cdot v_{Ncomp}(t)$$

5.4.3 Optical Registration of Pointers and Tools

Another approach is to both define and register features with hand-held pointers (Figure 5.28), similar to imaging-free procedures for orthopedic interventions, by tracking them with an infrared optical tracking system (OTS). Further registration of other image modalities with the patient can then be performed with intraoperative imaging procedures like ultrasound thickness measurements [Federspil05], which additionally serve to improve the quality of planning data.

5.4.3.1 Requirements

These pointers can be used to define single points in space, delineate regions on surfaces, or define trajectories. The markers attached to the pointers carry local frames and can be basically arbitrarily attached to the pointer itself. Therefore, before any points can be provided that

Figure 5.27: Robot approaching an implanted bone marker pin (only used for validation purposes in the presented system) with the miller tip through the manually-guided force following procedure.

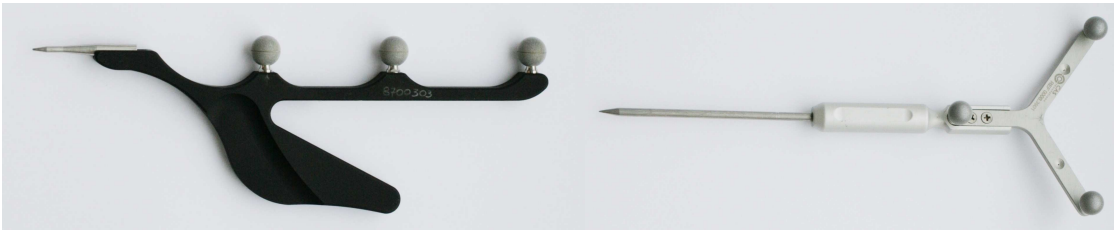


Figure 5.28: Two hand-held pointers with infrared-reflective marker points (5D and 6D)

way, the pointer needs to be calibrated to determine the transformation ${}^{A}p_{Z}$ from the marker A to the pointer tip Z or, in other words, the position of the tip in the marker frame. A technical drawing to do so may not be available, or the transformation be variable, so a more general method would be preferable. The same applies to the registration of the robot-held tool and its respective markers.

5.4.3.2 Hand-held Pointers

The presented approach is based on pivoting (rotating) the pointer around a fixed end point, i.e. the user calibrates a hand-held OTS pointer, determining the specific translation between pointer markers and pointer tip Z. The optical tracking system continuously reports the locations ${}^{OTS}p_{A,i}$ of the local marker frame in OTS coordinates, which are collected and continuously subjected to a sphere fitting algorithm to determine whether they lie on a sphere surface, and extract its parameters.[9] While some sphere can always be fitted to the location samples in the current window, it is only during pivoting that the successive sphere centres ${}^{OTS}c_{A,i}$ and radii $r_{A,i}$ stabilize long enough on valid values (Figure 5.29). When those parameters are established, the calibration is finalized by taking one more location sample ${}^{OTS}p_{A}$ and applying its inverse on the determined centre ${}^{OTS}c_{A} = {}^{OTS}p_{Z}$, so ${}^{A}p_{Z} = {}^{A}M_{OTS} \cdot {}^{OTS}p_{Z}$, where ${}^{A}M_{OTS} = \left({}^{OTS}p_{A}\right)^{-1}$.

The whole semi-automatic pointer calibration procedure requires the user to pivot the pointer for a short time to define a stable end point, switches into fitting mode to determine it

[9]The sphere fitting algorithm is adapted from [Umbach00]. There, a closed-form solution (which is thus a reliably real-time algorithm) for the determination of the sphere parameters center $(x, y, z)^T$ and radius r is outlined. The sphere fitting is performed by a Modified Least Squares method, yielding a sphere centre which minimizes the sum of the squared distances from this centre to all planes which are bisectors of pairs of sampled points. (For every two sampled points that should lie on a sphere, the plane that is normal to their connecting line and that contains their mid-point should ideally also contain the sphere center.) The radius can then be computed as the mean of the resulting distances between samples and the centre. This method is described as comparatively robust against both outliers and close neighbors.

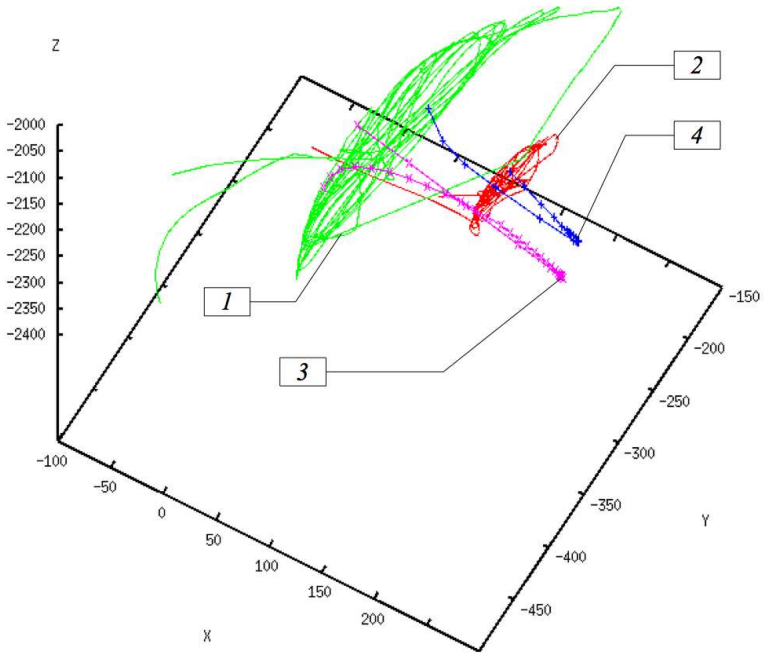


Figure 5.29: Traces of two optical pointers (1, 2) with resulting fitted sphere centers (3, 4) in OTS coordinates over time. Note how the sphere centers converge to the true pivot centers [Stolka06b].

exactly, and switches back after one second (Figure 5.30).

5.4.3.3 Robot-held Tool

The same procedure is used to determine the transformation ${}^{TM}M_T$ from tool marker TM to tool tip T to calibrate the tool. This is important since the tool marker must allow quick reconfiguration for good visibility in the operating room. However, unlike with the pointer calibration, for full registration between the tool (and thus the robot) and the OTS, their relative orientation needs to be determined, too. Therefore, these motions consist of pivoting around the miller tip T (to calibrate the exact position of the rotation center in OTS coordinates, where after successful tool calibration (Section 5.4.1) this should be coincident with the tip center) and subsequent small translation motions along the tool frame X and Y axes (each of length $150mm$), which together immediately span up the respective coordinate system of T and yield the transformation ${}^{OTS}M_T$ when projected into OTS coordinates. However, the absolute positioning accuracy of the robot may negatively influence the calibration quality.

Thus, the transformation relations between the OTS and the tool marker TM, the tool marker and the tool tip T, and thus the tool tip and any hand-held pointer tips Z_j are known. In the future, additional markers at the robot base can be included to serve as dynamic reference bases e.g. to detect accidental motion of the OTS (cf. Figure 1.7).

5.4.4 Tool Deformation Estimation

An important source of position errors is motion under contact, i.e. the influence of mechanical load to the whole structure (base, robot, tool holder, and tool) on the position of the tool tip. This influence results in wrong position information, i.e. incorrectly estimated tool tip position due to elastic deformation (bending or deflection). To model and estimate this error, it is necessary to characterize the possible loads and their mechanical effects to the system structure first.

Several simplifying assumptions can be made prior to modelling. First, the application allows mechanical contacts between robot and environment only at the tool tip; other significant

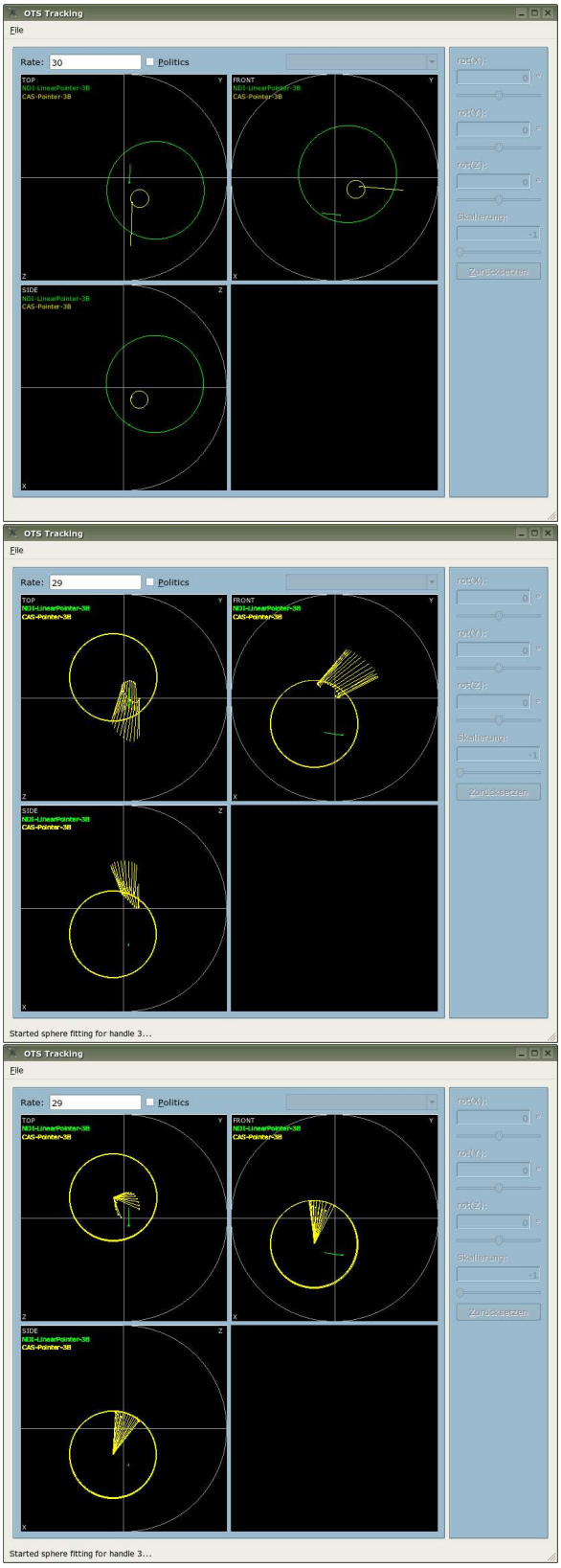


Figure 5.30: Pointer calibration by means of sphere fitting during pivoting. The uncalibrated pointers are moved into the OTS workspace, when fitting starts (top). When the estimated pivot center point stays stable long enough, collection of the sphere centers begins (center). When a certain time threshold is reached without disturbance of the centers, the calibration is completed (bottom).

loads to the arm or base are prohibited. Therefore it is sufficient to investigate the effects of forces and torques at the tool tip. Second, as the relative size of the tool tip – i.e. the miller head – is negligibly small ($\sim 2...5mm$) compared to its distance from the F/T sensor ($\sim 130mm$), it can be approximated as a dimensionless point of force application on an appropriate beam lever. As a practical consequence, torques on the tool tip apart from the one along the tool axis cannot be applied. Similarly, tensile, compressive, or shearing forces need not be considered. Third, the miller's electric motor is not powerful enough to effect relevant torsion of the remaining structure from the fixed position of the miller tip.

In a first step, the stiffness of the whole structure needs to be estimated. During the later online phase, tip displacements are accounted for by adding a corresponding error vector. Since there is no structural reason – like predetermined bending directions – for non-isometric deformation, the error model can be assumed as basically a linear combination of orthogonal deformations.

Stiffness data gathered from the experiments exhibited highly linear force-to-deflection ratios $k_i, i = x, y, z$, so the axis-aligned deformations themselves can be modelled as linear relationships. Together, the force vector F caused by the 3D deformation Δp is representable by a diagonal stiffness matrix as

$$\Delta p \cdot \begin{pmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_z \end{pmatrix} = F(\Delta p) \tag{5.12}$$

The position measurements $p_{measured}$ corrected with this deformation yield the estimate

$$p_{corrected} = p_{measured} + \Delta p(F) \tag{5.13}$$

This force-based deformation measurement allows more precise mapping when used in the local navigation cycle, since sensor samples can then be entered into the map at their actual location of origin [Stolka05]. A similar deformation estimation approach on the same robot kinematic was described previously in [Engel02], although apparently with a more limited validation as compared with the experiments in Section 5.5.3.4.1. Note that this procedure does not include controlled correction of the tip position itself; such active compensation was described e.g. in [Sugita08] based on spline interpolation of the estimated deformation for a 7-DoF redundant kinematic.

5.5 Experiments

Experimental verification of this chapter's statements is split up into directly map-related aspects for concurrently-built maps (map-building in simulations; Section 5.5.1), map-building with ultrasound sensors (in a phantom experiment using the actual system; Section 5.5.2), and localization aspects (registration, calibration, and deformations for the actual system; Section 5.5.3).

Discussion will focus on functional requirements of map-building (do the MLS represent the environment faithfully, how precisely, what space complexity do the MLS have?) as well as non-functional requirements (can the MLS be built either concurrently or at least "fast enough", are setup procedures an obstacle to actual use, and is map-building safe?).

5.5.1 Validation of Concurrent Map Building

The experiments performed related to the map building investigated the influence of sensor motion type, active optimizations, and different environments. The relative quality of the various described map projections was assessed through simulations.

5.5.1.1 Concurrent Map-Building Algorithms

The validity of the proposed mapping methods with and without optimizations needs to be evaluated in a series of experiments. As real setups with biological tissue preparations yield non-reproducible results (thus precluding repeated experiments e.g. with parameter variations), and more importantly, since there is no ground truth for the tissue distribution, simulations should be adopted during this phase to choose appropriate mapping algorithms. These simulations need to include *(1)* a model of the environment, *(2)* a geometrical model of the tool, *(3)* interaction between both, *(4)* a suitable representation of the local sensors associated with the tool, and finally *(5)* the mapping data structures and algorithms necessary to build the maps[10]. Using a simulation is a valid approach here, as our focus lies on evaluating the relative performance of mapping algorithms, not actual sensor data classification.

5.5.1.1.1 Simulation Experiment Setup Different mapping algorithms (or optimizations combinations) were used in parallel to build several maps to be compared eventually. As intermediate mapping results during the simulation may differ because to increasing veracity of the map that is due to repeated visits to specific locations, at least two sets of temporal snapshots were necessary – one after traversing only a fraction $f < 1.0$ of the occupied portion of the represented environment, and one after completely traversing the environment. Confusion matrices were created for each set and evaluated.

To this end, the proposed mapping scheme was tested including four different optimization levels (optimization 1/*directional active shape modification*, 2/*static environment assumption*, both, and none). In a 3D environment with a layered obstacle (top-to-bottom: air/C_0; bone/C_B ($6mm$ thickness); dura/C_D; $40 \times 40 \times 40mm^3$; $1mm$ discretization; Figure 5.32), a simulated spherical miller tool (tool diameter $r_S = 4.5mm$; active shape diameter $r_{active} = 5.5mm$) interacted with the environment with both a Brownian motion (a random path with a maximum step size of 1 in the three-dimensional 26-neighborhood) and a systematic horizontal-parallel path (horizontal path offsets at 75% of tool radius; similar to the milling paths of RONAF), engaging in destructive mapping. Contacts were determined according to a single simulated local sensor (see below for details). The resulting maps were compared with the initial environment (reading their feature lists' last entries $c \neq C_0$, Pre-Terminal Last-Known Map/PTM) and with the final environment (their last entries, Terminal Fate Map/TFM).

In the following, classification denotes the result of mapping single map entries. As the relative cost of misclassification is highly application-specific, only the ratio of correctly/incorrectly classified map entries is discussed. For Brownian paths, the simulation was run repeatedly since the maps are dependent on path history (although results did not differ significantly). Furthermore, for the simulated local sensor, a spherical tool shape and an underlying classifier dominance relationship $C_B > C_D > C_0 > C_?$ of sensor data classifier results (i.e. the dominance of partial contact states within the active shape over others; a reasonable approximation of the actual tool-based local sensor behavior in the investigated application) was assumed.

5.5.1.1.2 Simulation Experiment Results With random walk motion (a very non-restrictive assumption on tool behavior on a discrete grid[11]; Table 5.3, Table 5.31) and partial exploration (200 non-air contact steps), activation of either Optimization 1 and Optimization 2 generally yielded better results than respective deactivation, while activating both was generally best w.r.t. correct classifications. As for misclassifications, Optimization 1+2 was always 2nd-best

[10] All these aspects are encoded in the simulation program *MapSpaceTest*.

[11] A *random walk* is a series of discrete steps of width 1 in each dimension with equally-distributed direction. Since any limit in any dimension will almost surely be crossed by a random walk, the motion needs to be clipped (to the finite map's borders) in the investigated case.

	Opt –	Opt 1	Opt 2	Opt 1+2
PTM (partial)	12261 / 3362	15791 / 3836	13847 / 1776	17351 / 2276
TFM (partial)	13358 / 2265	17669 / 1958	15149 / 474	18399 / 1228
PTM (complete)	50707 / 13281	50246 / 13754	56402 / 7586	54978 / 9022
TFM (complete)	63986 / 2	64000 / 0	63988 / 0	64000 / 0

Table 5.3: Map Building with random walk motion (correct/incorrect classifications; data of one example run for partial exploration; colored: improved cases).

	Opt –	Opt 1	Opt 2	Opt 1+2
PTM (partial)	28160 / 7520	28182 / 9138	35680 / 0	35666 / 1654
TFM (partial)	24480 / 11200	32622 / 4698	32000 / 3680	36959 / 361
PTM (complete)	44800 / 19200	43200 / 20800	54400 / 9600	61979 / 2021
TFM (complete)	24236 / 39764	44894 / 19106	33836 / 30164	64000 / 0

Table 5.4: Map Building with Horizontal-Parallel Motion (correct/incorrect classifications; data of one example run for partial exploration; colored: improved cases).

(2nd only to Optimization 2).

The necessity of the proposed optimizations becomes immediately clear when comparing their effects on paths with less random influence which approximate the milling paths of a surgical robot better, together with a simulation environment derived from anatomical CT images. When following a horizontal-parallel path, the TFMs improve massively under both optimizations combined (Figure 5.33). With horizontal-parallel motions (200 contacts; Table 5.4), there was a huge positive influence by Optimization 2, while activation of Optimization 1 yielded vastly better results for TFMs (not PTMs) w.r.t. both correct classifications and misclassifications; activation of both Optimizations 1+2 was best overall. For complete environment explorations (approx. 2000 contacts), the results were basically the same.

From these results, it can be concluded that the proposed optimized mapping methods improve the localization of classified map entries (i.e. putting the right information in the right place) for reasonable approximations to real sensor motions. Depending on the purpose of mapping, one might argue for the use of different optimization combinations; however, using both presented methods simultaneously is generally superior to any other approach – both for (especially) partial mapping and, in the limit, for complete mapping. It should be noted, however, that dominant classification results "bleed over" into neighboring regions, as described before. This is an inherent property of the local sensing approach; its significance needs to be evaluated based on the intended application.

5.5.2 Validation of Independent (3D Ultrasound) Map Building

The entire described procedure for 3D ultrasound map building, including pointer and robot calibration and registration, sampling of skull surface points, demonstration and processing of scan paths, and their execution with concurrent US scanning, was performed on five human skull preparations (cf. Figure 5.17) and one plastic skull dummy.

5.5.2.1 Input

The manually generated surface point cloud is a 2.5D global map registered to the robot, with points localized down to a precision of $< 2.5mm$ (mean error distance between handheld pointer tips and robot-held tool tip, [Stolka06b]). This relatively large error stems from an accumulation of smaller error components. Looking closely at the procedure used to create

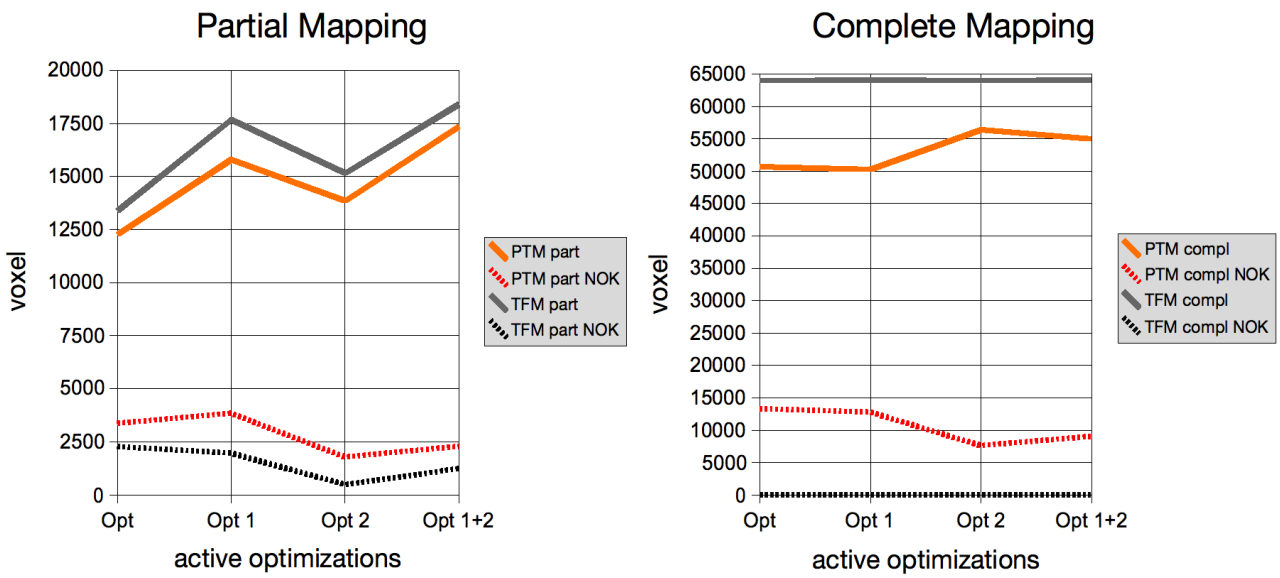


Figure 5.31: Results of simulated map building with random walk motion, depending on which optimizations were activated (left: one example run of partial mapping with 200 contacts; right: complete mapping with approx. 2000 contacts; correct classifications with solid lines, incorrect classifications dashed/marked "NOK").

this map, it is apparent that the calibration accuracy of the hand-held pointer, its position as reported by the OTS, the position of the tool marker as reported by the OTS, and the tool calibration accuracy all contributed to the sequential deterioration of the registration quality between the pointer and the tool tip (i.e. between map and robot). For an overview of error sources see Table 5.1 [Stolka06b].

5.5.2.2 Scan

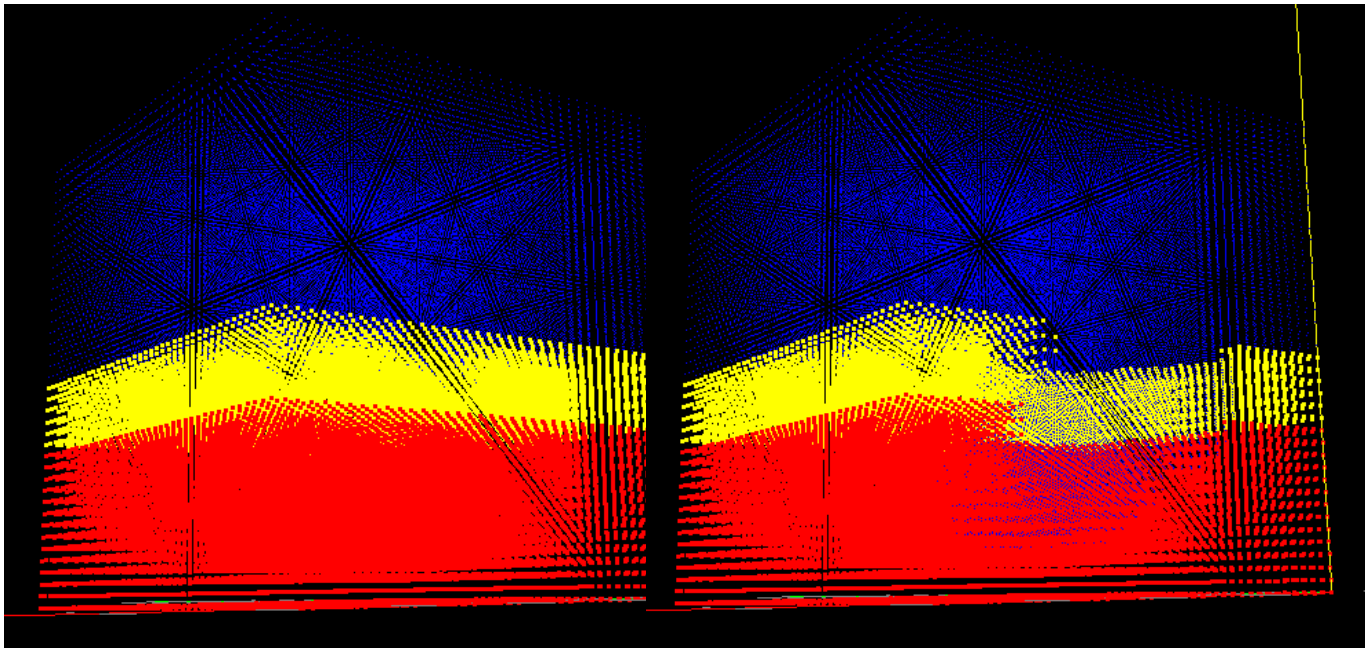
For the scan path specialization (i.e. the constraining step from the normals-aligned 5D path to a completely constrained 6D path with roll values fixed), a simple procedure of selecting a fixed roll value ϑ for a complete path was used. For the set $\Theta = \{\vartheta_k\}$, $\vartheta_k = \vartheta_0 + k \cdot \Delta\vartheta, k \in \mathbb{Z}$, the variables were chosen so the roll values ranged from $-30°$ to $+30°$ in 15 steps. For the given setup, this simplistic approach worked sufficiently well. In future work, this is expected to be replaced with a search algorithm for optimal-joint-motion specialized paths.

Low registration quality in this step sometimes resulted in scan motions outside the actual volume of or within the skull bone. However, with the flexible delay line attached to the front of the US probe, all scans could be completed successfully, i.e. no excessive forces were exerted during the robot-based scan.

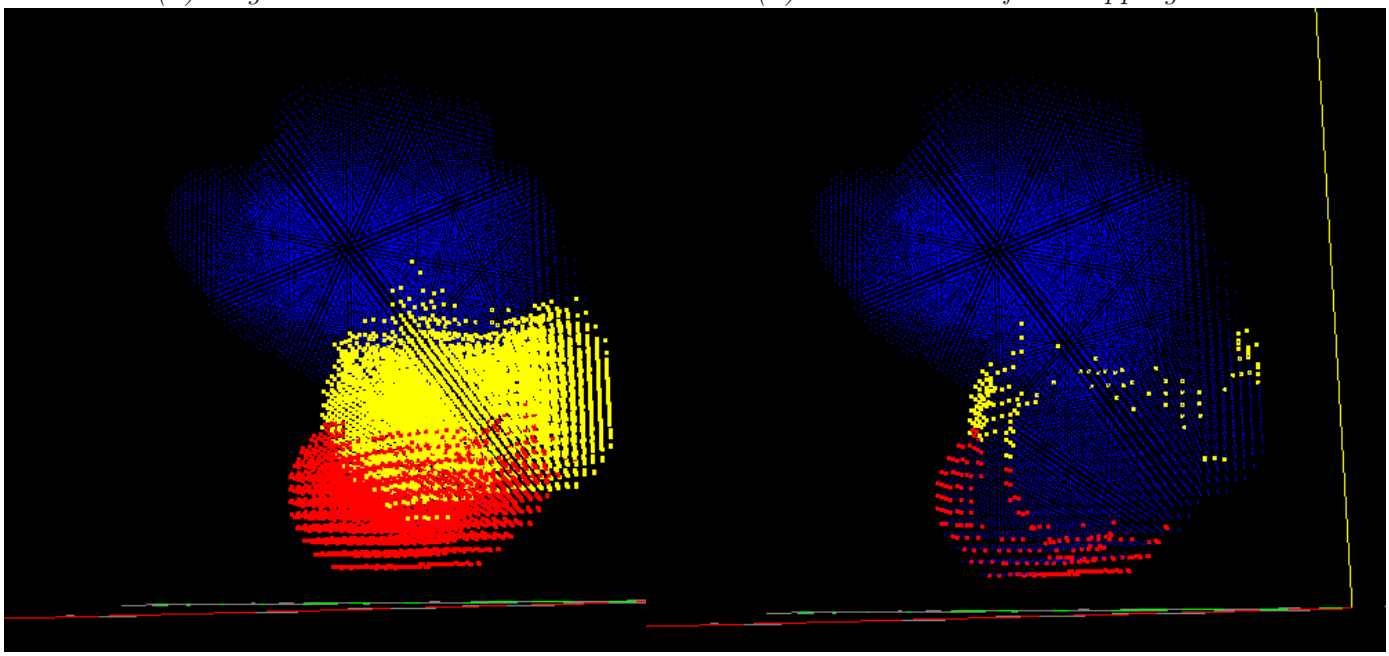
Still, after its introduction the force-based contact pressure control proved very useful. The relative number of measureable ultrasound thicknesses increased (to approx. 75%), and in these the signal strength, especially of the second (inner) echo, increased. This effect is due to the improved contact between delay line and the skull surface, i.e. no gaps between them occurred.

5.5.2.3 Output

The US scan finally generates a map consisting of two 2.5D skull surface point clouds, both of which are registered to the robot with a precision equal to the robot's absolute positioning accuracy (close to the robot's relative accuracy of approx. $0.35mm$) (Figure 5.18). Another factor in the scan map's precision is the accuracy of thickness detection in the single US shots, which is approx. $0.5mm$ (standard deviation for US pulse code excitation with direct coupling

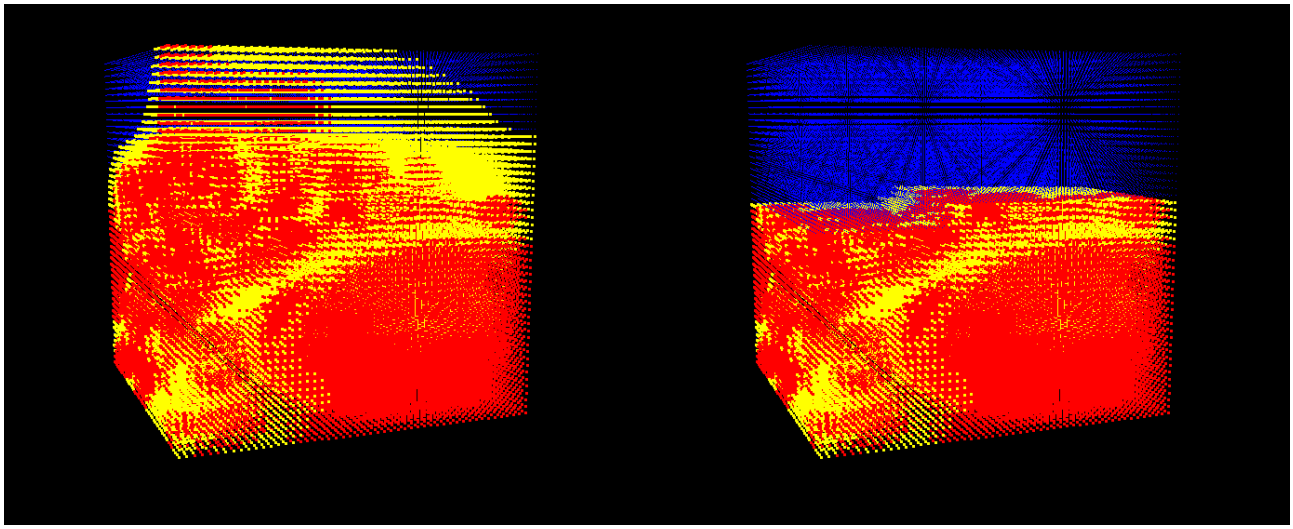


(1) original environment *(2) environment after mapping*

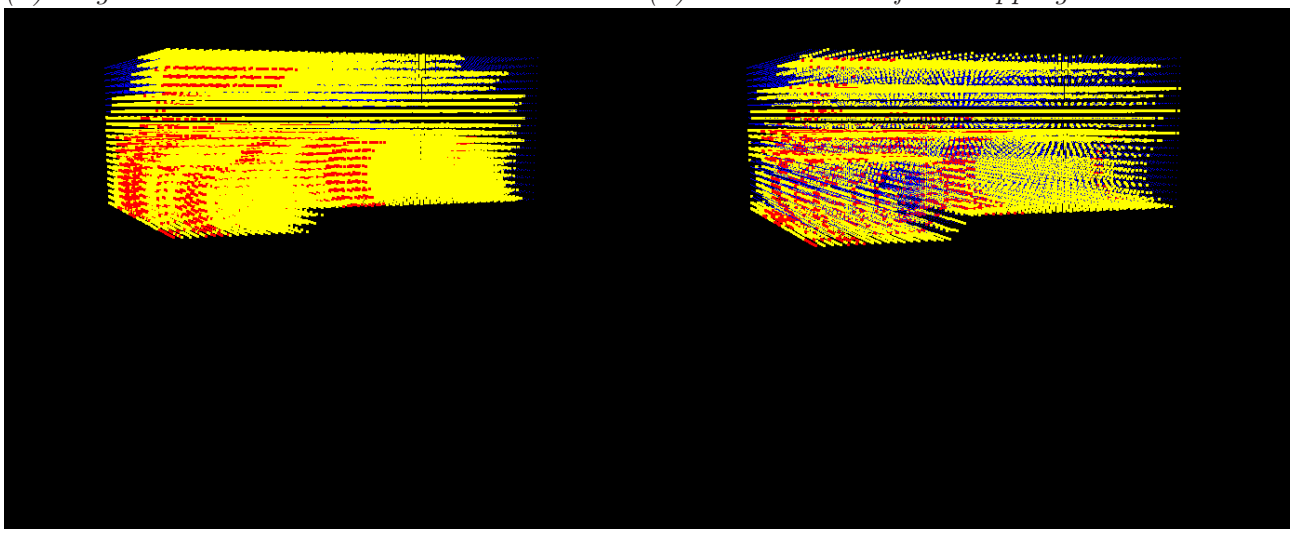


(3) PTM *(4) TFM*

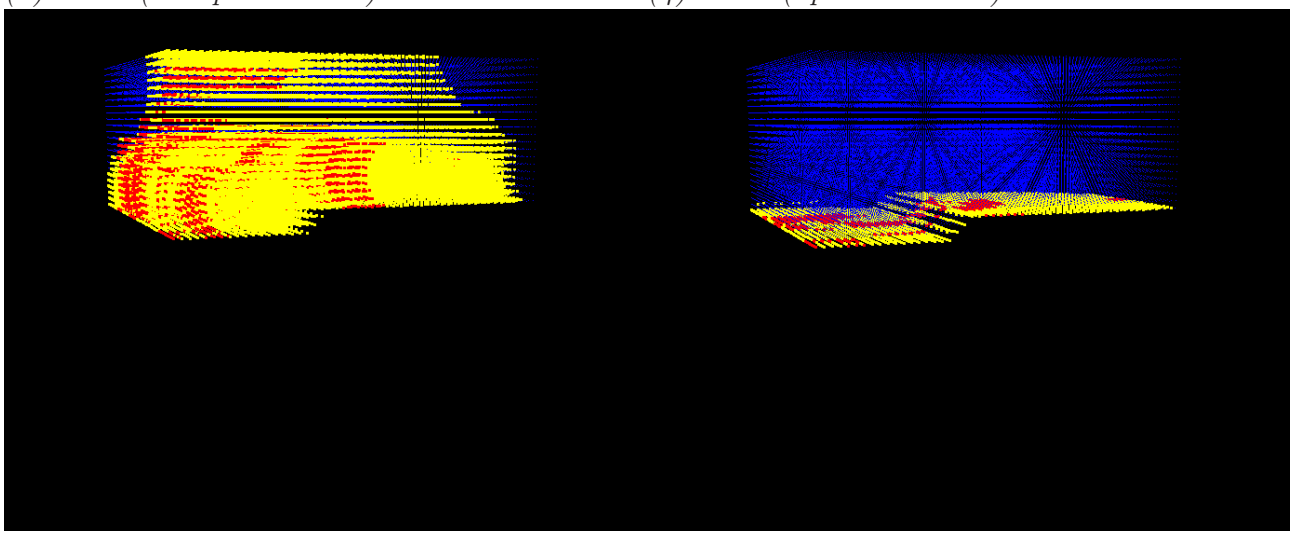
Figure 5.32: Simulation results after partial random walk path exploration with destructive mapping and optimizations 1+2: Initial environment and final situation (top left/right); PTM and TFM (bottom left/right) (all images from same perspective; dark blue: C_0, bright yellow: C_B, red: C_D, not visible: $C_?$) [Stolka07b].



(1) original environment *(2) environment after mapping*



(3) TFM (no optimization) *(4) TFM (optimization 1)*



(5) TFM (optimization 2) *(6) TFM (optimizations 1+2)*

Figure 5.33: Simulation results after partial horizontal-parallel path exploration with destructive mapping. Initial environment and final situation (images 1, 2); TFM with optimizations: none, optim. 1, optim. 2, and optim. 1 and 2 (images 3, 4, 5, and 6) (all images from same perspective; dark blue: C_0, bright yellow: C_B, red: C_D, not visible: $C_?$).

of transducer and bone [Tretbar04a]; or $0.3mm \pm 1.0mm$ in [Federspil10]), but this value is highly patient-specific.

The upper profile points correspond to the path actually traversed by the US sensor's calibrated delay line edge up to a precision of around approx. $0.1mm$, i.e. well within the robot's positioning accuracy. The lower profile points' precision is less easy to estimate. The only ground truth available that does not introduce additional positioning errors apart from its own resolution is a CT of the region of interest, registered to the 3D ultrasound volume. Initial comparisons resulted in deviations of the lower profile relative to the the CT of around $0.7mm$.

5.5.3 Validation of Localization

Experiments were conducted to determine the attainable precision of the tool calibration (Section 5.5.3.1), registration by force-following (Section 5.5.3.2), the optical tracking method (Section 5.5.3.3), and the deformation estimation (Section 5.5.3.4).

5.5.3.1 Tool Calibration

The precision gained by the automatic tool calibration procedure (Section 5.4.1) can be expressed in terms of the translational error or offset $d_{xyz} = \sqrt{d_x^2 + d_y^2 + d_z^2}$ of the tool transformation.

This offset added up to an improvement of as much as $2.1mm$ in the experiments compared to the initial transformation determined from tool blueprints and manual measurements. Further repeated calibration runs yielded only marginal improvements on the order of $< 0.02mm$ (i.e. well below the robot repeatability, so near-perfect calibration was achieved already with the initial calibration). These values were measured with the robot encoders themselves, which implies that there are no additional registration errors introduced into these measurements.

5.5.3.2 Registration by Force-Following

Plotting exerted absolute force $|F|$ vs. absolute speed $|v|$ from an experiment (approximating a single point five times in a row in contact-free motion) results in the following graphs (Figure 5.34). For a simple P controller, the graph shows a linear relationship. High forces result in only small speeds. The slope can be influenced by changing the controller gain, but obviously choosing too high a value results in coarse small scale motions.

For the N controller, no direct relation between $|v|$ and $|F|$ exists anymore, so different speed levels may correspond to a single force level and vice versa. Especially irritating is the large variation in speeds for small forces, reflecting the controller's unsteadiness in exact positioning tasks.

Finally, the Hybrid N/P controller shows a clear and repeatable hysteresis cycle. Starting from small forces and small speeds, increasing forces map quite linearly to increasing speeds until the speed threshold v_{thresh} is reached. Then, forces can be lowered until a steady high speed is attained with only a comfortably small effort (still, friction requires a tiny net force). Decelerating towards the goal is achieved by zeroing the exerted force, which reduces speed slowly through friction until v_{thresh} is reached again from above. This results in a rapid rise of P control influence, decelerating nearly instantly. Final exact positioning occurs within P control bounds.

The three implemented control algorithms (P, N, and Hybrid N/P control) were compared through the following experiment. The setup consisted of a fixed table with two marked spots P_0, P_1 (distance $l = 1m$) within the robot's working space. Both points were taught manually (driving the robot via the manual control pendant) before the experiment took place in order to have as exact position data as possible. The task was to approach each point and to confirm

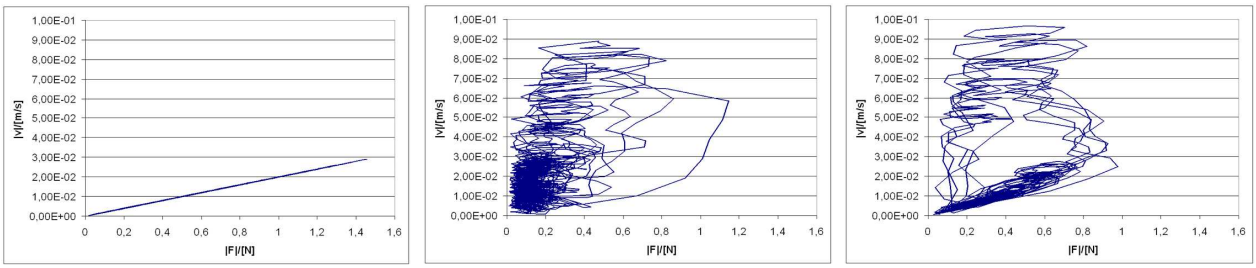


Figure 5.34: Absolute speeds $|v|$ vs. absolute forces $|F|$ with P, N, *and* N/P *control (left, center, right) [Stolka03].*

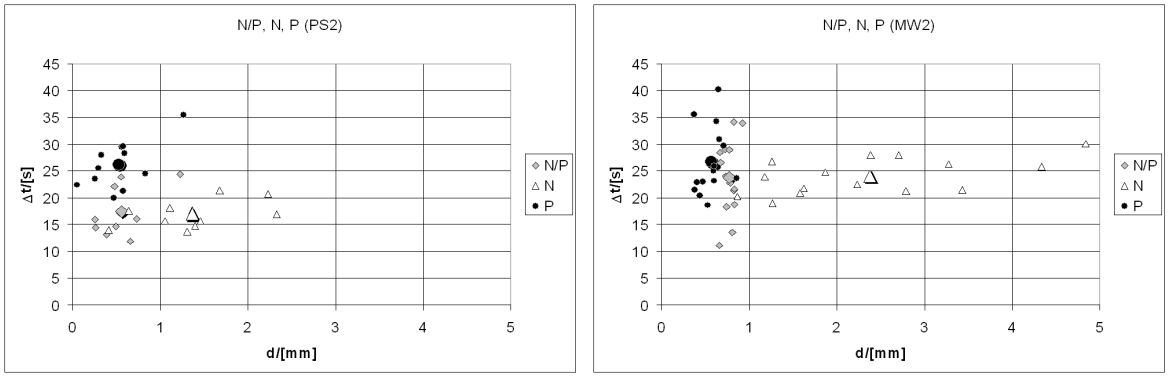


Figure 5.35: Final position deviation d vs. approximation duration Δt (for two participants) (larger symbols: mean for each controller algorithm) [Stolka03].

the position, whereupon automatic force-based approximation along the z-axis took place. This was repeated $n = 14$ times in a row for each control algorithm (N, P, N/P). The time of each target point approximation t_i and the remaining position deviation d was recorded. Besides, during each control cycle, forces $|F(t)|$ and the current position $P(t)$ were logged. The resulting data for each controller thus consisted of n 2-tuples $(\Delta t, d)$ (Δt being the duration of each approximation).

The results confirmed the expectations. While P control proved to be exact and slow, N was fast, albeit with large end deviations. The Hybrid N/P controller combined the others' advantages (fast, precise) without their drawbacks (Figure 5.35).

5.5.3.3 Optical Tracking

Registration of the robot with the intervention region is performed by optical tracking (three-point-based [Stolka03] or by delineating a region to scan with an A-Scan ultrasound probe to reconstruct the 3D intervention volume [Stolka07a]). The precision of this procedure (better than $2.5mm$) is unrelated to the navigation precision achievable by using any later map from local sensors, because local sensor data is later sampled in a separate step with a precision only limited by the localizer accuracy (i.e. the robot encoders).

Determining the precision of the OTS subsystem includes two steps: First, measuring the precision of the pointer calibration (Section 5.5.3.3.1), as the pointers are used to define points in the robot frame later, and second, measuring the precision of the registration between robot and OTS to allow correct projection of optically sampled points into the robot frame (Section 5.5.3.3.2).

One important factor pertaining to the OTS itself is its own sensor noise. The RMS value of position differences can be bounded to a very small value of approx. $0.05mm$, however. After an initial warming-up phase of at most $45min$, noise (short-term position differences) and shifts (long-term position differences) stay at their values (Figure 5.36).

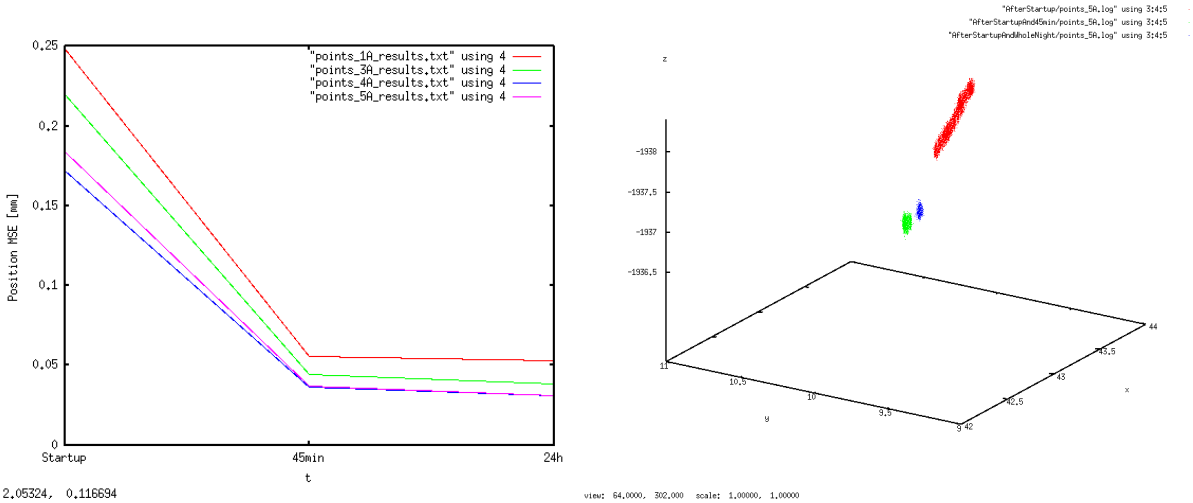


Figure 5.36: Noise (left, four different markers) and position drift (right, three different time windows) returned by the NDI Polaris optical tracking system. After 45min, both stabilize. Positions gathered during the initial warm-up phase may be up to 2mm off from their true positions and are subject to as much as 0.25mm RMS noise (standard deviation from mean).

5.5.3.3.1 Pointer Calibration First, the precision of the sphere fitting procedure was determined experimentally (Figure 5.37). Data samples were generated by pivoting one pointer continuously around its tip Z and taking the coordinates of its (rotating) marker A as returned by the OTS. The standard deviation (also root-mean-squared error/RMS) of the distances $d_{c_i} = \left\| p_{C_i}, p_{C_{i+1}} \right\|$ between $n = 50$ repeatedly collected sphere centres p_C was computed as $s_C = 0.33mm$ (average for several runs) for a pivoting radius of $210mm$ and $s_C = 0.25mm$ (average for several runs) for a pivoting radius of $88mm$. The maximum distance between the reconstructed sphere centres was about $2.6mm$. These values can be interpreted as the maximum accuracy achievable for OTS-based definition of 3D points, as they describe the precision of the sphere fitting based calibration (in calibration, for the pointer tip Z its position is set to $p_Z = p_C$, or ${}^{A_i}M_{Z_i} = {}^{A_i}M_{C_i}$).

The accuracy of the OTS-based pointing itself was tested then. The tip Z of a tool with radius $210mm$ was pointed at constant locations with different angles, and the distance d_{Z_i} between successive tip position samples p_{Z_i}, p_{Z_i+1} was computed. Being highly dependent on the quality of the initial calibration and the type of pointer used, this distance oscillated around mean $\mu(d_Z) = 0.69mm$ ($0.42mm$ excluding outliers) with sample standard deviation $s_Z = 0.32mm$.

Finally, the "two-sided" precision of the calibration procedure was tested with two pointers. After calibrating each pointer (i.e. determining the ${}^{A_i}M_{Z_i}$ for $i = 1, 2$), their tips Z_1 and Z_2 were pointed at each other at different locations within the OTS work volume and at different angles w.r.t. each other. The resulting distance $d_{ZZ} = \|p_{Z_1}, p_{Z_2}\|$ between both tips can be interpreted as the maximum achievable precision at which any two OTS-tracked objects can be controlled relative to each other, and was determined to be around $\mu(d_{ZZ}) = 0.5mm$.

5.5.3.3.2 Robot-Camera Registration The tip-to-marker registration ${}^{TM}M_T$ of the tool was tested with a hand-held pointer in a similar setup as the two-sided precision above. The pointer tip Z was pointed at the tool tip T, both of which have been calibrated as described above. Then, the distance between the (single) computed locations of T and Z at different angles of pointer and tool was calculated; it amounts to the comparatively large value of mean

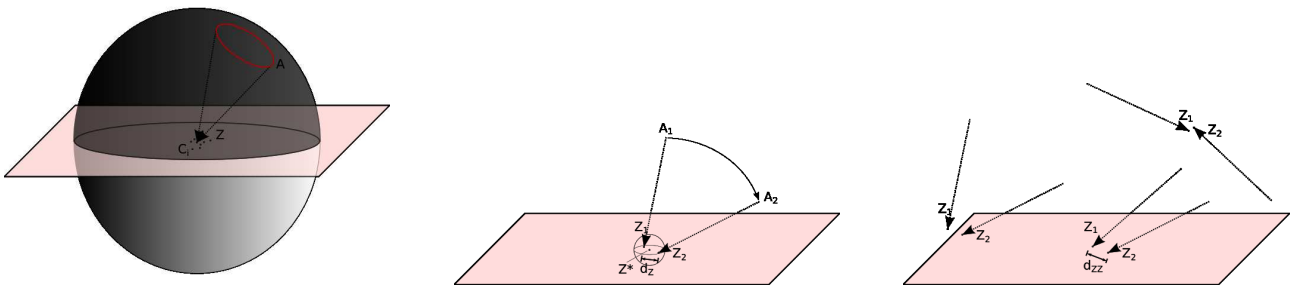


Figure 5.37: Pointer calibration validation: Distances d_{C_i} between centers C_i of repeated sphere fitting (left); distance d_Z of repeated pointing at same position $Z*$ with single pointer (center); distances d_{ZZ} of pointing at same positions with two pointers (right).

$\mu(d_{TZ}) = 2.5mm$.[12]

Finally, the tool-to-OTS registration ${}^{OTS}M_T$ was tested. The robot encoders were assumed to be "ground truth", i.e. giving accurate positions relative to the robot base, and were used to define a set of positions ${}^{ROB}p_i$ within the relevant robot workspace by force following. The positions ${}^{OTS}p_i \approx {}^{ROB}p_i$ were demonstrated a second time with a pivoting pointer relative to the origin of the tool calibration position. The mean distance between corresponding pairs of positions was then measured as $\mu(d_{TP}) = 2...4mm$. This value includes the whole uncertainty chain of robot repeatability (for the initial point definition), precision of defining points by pivoting (d_{C_i} as above, for the subsequent OTS-based point definition), and the tool calibration precision (d_Z as above, for tool tip-to-marker registration).

5.5.3.4 Force-Based Deformation Estimation

In the following, details of the stiffness estimation step and their consequences for deformation estimation are described.

Deformation estimation experiments for the robot and tool setup were performed by approaching a rigid obstacle with varying orientations, measuring the forces and positions starting from the point of contact to crossing a predefined absolute force threshold $F_{max} = 30N$. The gathered data was used to compute structure compliances along the tool axes.

First, the input force values need to be compensated for gravity influence under varying orientations. Only then can the deformation be properly decoupled from the force from the tool combination mass (around $1.6kg$) without calibrating the F/T sensor ("zeroing") for every deformation measurement. Then, stiffness was estimated through measurements along the tool main axes (Section 5.5.3.4.1). In a final step, the assumptions and resulting values were successfully cross-validated by exhaustive deformation testing under varying angles of attack and in a milling experiment (Section 5.5.3.4.2).

5.5.3.4.1 Stiffness Estimation The stiffness of base, robot, tool holder, and miller along the three tool axes was determined by deforming the structure against a known ground truth, i.e. a fixed and rigid obstacle, and measuring the resulting forces and positions. The range of interest for the absolute applied forces is about $|\mathbf{F}| = 0...35N$ for milling applications (as a result of force-based speed control, aimed at limiting bone heat trauma [Federspil03b], [Fuchsberger86]).

As the results show that the observed force-to-deflection ratio is constant for the system setup and for the range of forces applied (Figure 5.38), there is no need to apply higher-order approximation functions or implement a lookup table for the deformations. A simple error estimator of the form

[12] As this value is considerably larger than the d_z and d_zz above, this would merit closer investigation in future work.

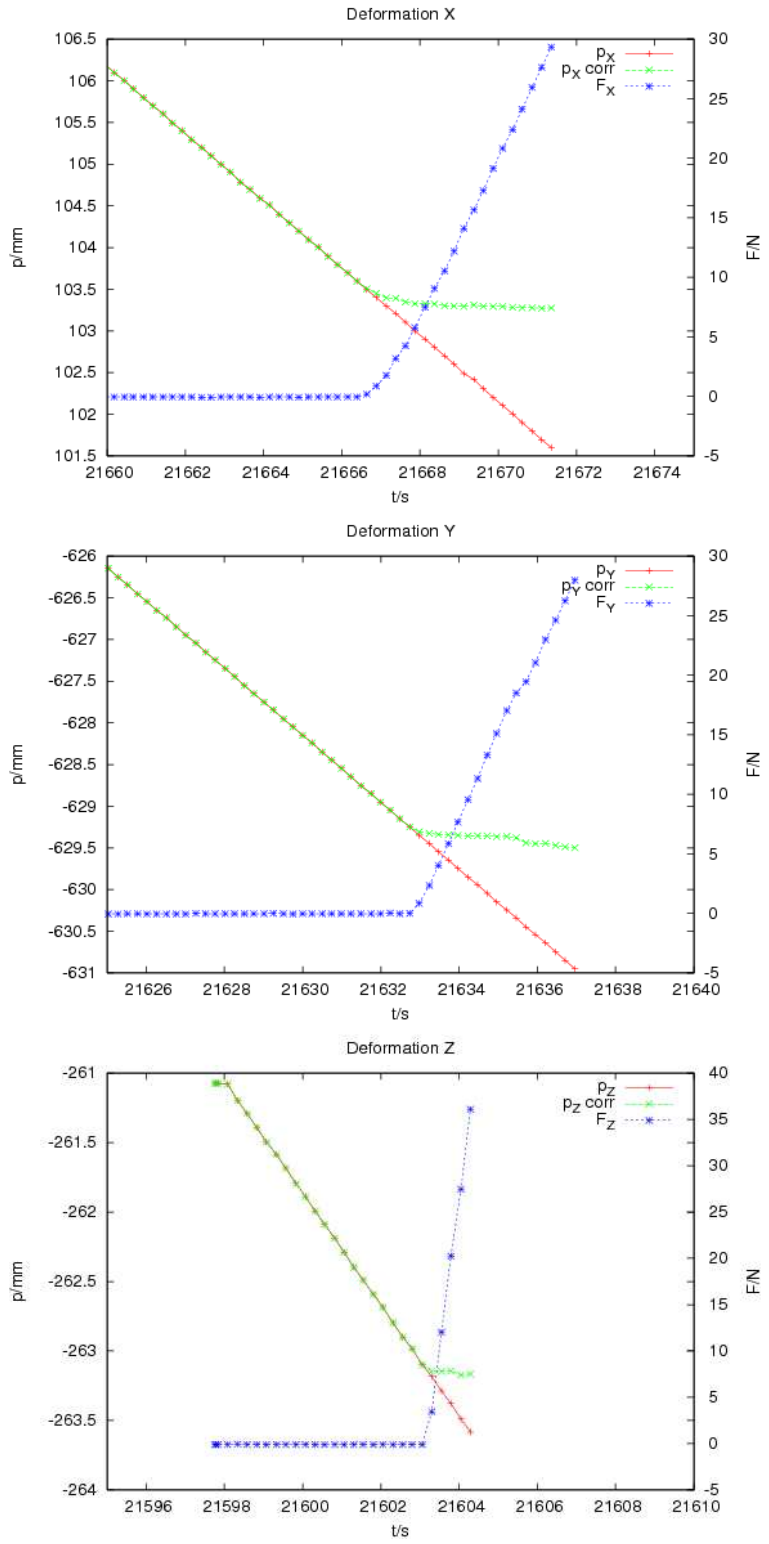


Figure 5.38: Experimental results of stiffness estimation: Expected/commanded tool tip position (straight red, p_i), applied forces (rising blue, F_i), and resulting estimated position based on forces measurements (falling green, $p_{i,corr}$) along tool axes X, Y, Z (top to bottom) [Stolka08].

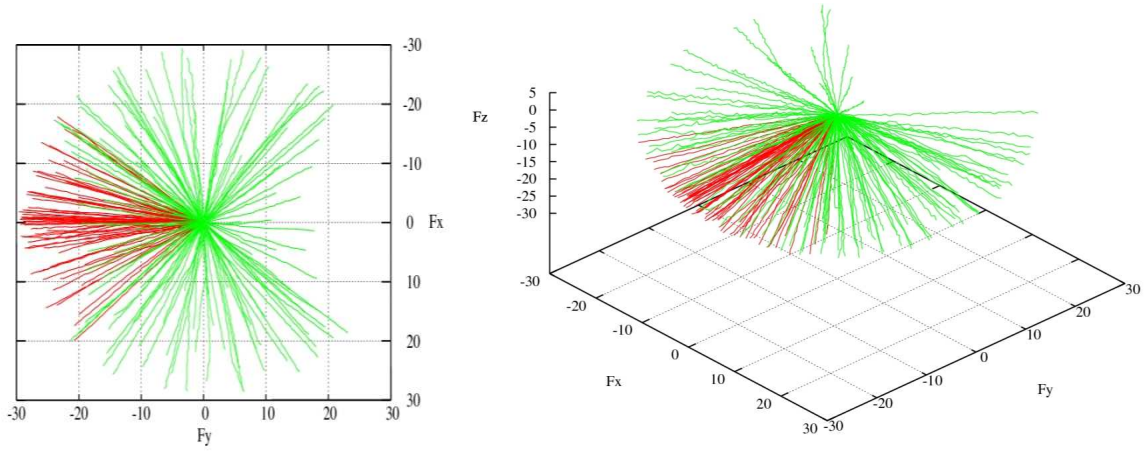


*Figure 5.39: Experimental validation of the assumption that deformation forces are linearly linked to the actual tool tip deflection. The graphs show the trajectories of 3D force vectors (in [*N*]) pointing away from the point of contact for a set of linear deformation motions (limited to a force protect limit of $F_{max} = 30N$). The red and green subsets corresponds to two distinct set of approaches, covering most (but not all, due to kinematic limitations) of the "approach directions sphere".*

Stiffness	k_x	k_y	k_z
	$17.52\frac{N}{mm}$	$19.27\frac{N}{mm}$	$86.66\frac{N}{mm}$

Table 5.5: Measured structural stiffness $k_{x,y,z}$ along tool frame axes

$$\Delta p_i(F_i) = \frac{1}{k_i} F_i, i = x, y, z \tag{5.14}$$

is sufficient (Table 5.5 lists the measured stiffness factors). These results were corroborated by a more extensive experiment setup, testing $n = 162$ non-orthogonal and non-axis-aligned directions of deforming contacts (Figure 5.39, [Harrer07]). This supported the initial assumption that total deformation can be approximated by a linear combination of axis-aligned deformations.

The remaining localization error (after deformation estimation) is below $0.25mm$ for all axes, while the worst non-corrected localization was over $1.75mm$ off for the force range under scrutiny, as seen from Figure 5.38.

5.5.3.4.2 Deformation Estimation The measured stiffness is used to estimate the deformation error for the tool tip. A milling procedure has been executed for an actual implant-based excavation volume (generic path for Vibrant Soundbridge, Siemens/Symphonix) in samples of oak wood, which in former experiments [Waringo01] has been shown to exhibit similar properties to human skull bone, except for hardness. The used miller head was a $4.5mm$ diameter spherical drill.

The occurring forces are highest along the Z axis, i.e. the miller's rotatory axis. The vertically-held miller has only little removal capability at its apex, which is desirable in everyday use because of the limited damage to the dura mater that can be inflicted this way. However, this also implies high reactive forces up and out of the milling excavation volume. The effects can be seen in Figure 5.40, showing an enlarged side view of the recorded milling path, which consists of four distinct sinking iterations. The effective position – under estimated deformation – consistently lies above the planned position during planar (X/Y) milling. The vertical (Z) feed motions show almost no lateral deformation. The effect becomes more pronounced with increasing milling depth (negative Z), where the miller head operates at higher removal rates.

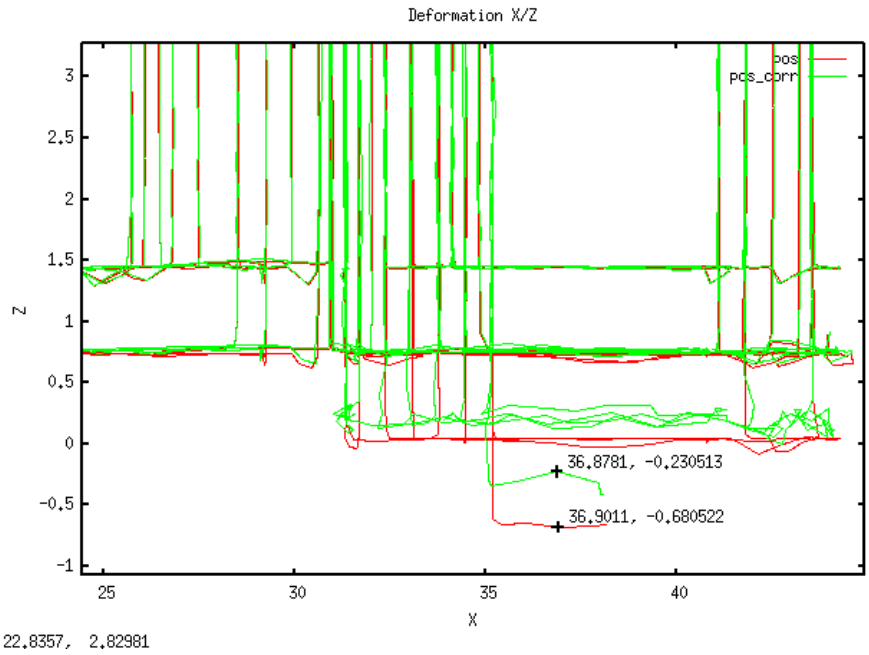


Figure 5.40: Enlarged side view (X/Z plane) of recorded milling motion (Soundbridge implant, four sinking iterations top-to-bottom, in wooden block); showing the difference between commanded milling path (red line) and effective path to due to deformation (estimated; bright green line). Forces pointing away from the bottom miller pole are pushing the tool upwards. Execution was aborted at the position marked "+" due to excessive force (adapted from [Stolka06b]).

The maximum deformation observed during this experiment was $\sim 0.45mm$.

5.6 Discussion

In this chapter, all the steps and algorithms necessary to couple the local sensor data with position information to produce maps from local sensors (MLS) were presented. The original Hypothesis 2 underlying the work in this chapter was:

> *"Faithful and current maps of relevant features in the intervention area of surgical milling interventions can be built concurrently with certain local sensors and additional localizers."*

As summarized in the following paragraphs, the separate steps to achieve this goal – building of maps from local sensors, registration of the robot to the operation environment, and precise localization – have been investigated successfully. This allows to conclude that it is possible to create maps in the described way, and that these maps can support navigation algorithms, some examples of which are investigated in the following chapter.

Maps from Local Sensors

Two complementary types of MLS were introduced – concurrently-built MLS (conMLS) and independently-built MLS (indepMLS). Both rely on the incoming local sensor data as described in the previous chapter (Chapter 4). Localization is provided by the robot itself through its own encoders. However, the necessary precision is only achievable after proper tool (and thus also sensor) calibration has been performed, as presented here.

Concurrently-Built Maps from Local Sensors

It was shown how continuous streams of local sensor data can be combined with localization information to be entered into maps in real-time. For the investigated sensors (and in fact all physical ones) the actual size is non-zero, which leads to severe restrictions on the achievable precision of data entry into the map. Two optimizations were presented that allow to partially overcome blurring due to the tool's point spread function. In particular, these optimizations improve the map quality with each pass (i.e. each visit to the position in question).

The complexity of the proposed data entry functions is linear in the size of the tool generating the local sensor data and in the resolution of the map itself. Therefore, the current real-time capability of the simulation prototypes can be guaranteed in real setups as well.

In future work, it might be worth investigating how data from multiple sensors can improve or impair map quality. In particular, it may be possible to perform optimal 2nd-level classification within map entries: When different, possibly contradicting sensor data classifications are entered into a single map entry, the decision on a single environment state might be informed by a learning algorithm. Such an approach might compare simulation-based entries with actual ground truth data, learn relations between the different sensors' classification dominance relationships, and accordingly shift *a-priori* probabilities for the final decision.

Independently-Built, Robot-Based 3D Ultrasound Maps

A process was described for bone surface and ultrasound scan path input using an IR pointer registered with a robot, resulting in *1)* a navigation system-based surface map and *2)* a robot-based 3D ultrasound map with extracted skull boundaries. These maps are registered non-invasively, thus representing an advance in RAS planning data acquisition.

With respect to the initial design requirements, it can be stated that the 3D ultrasound and surface maps created can be used for CT-free interventions. Bone thickness can be measured with a precision of $0.3mm \pm 1.0mm$, comparable to CT. These measurements are compiled into a reconstruction volume based on the spatial information supplied by robot encoders, resulting in a maximum expected $0.85mm$ imprecision in the reconstruction when adding in the $0.35mm$ robot repeatability (which effectively describes registration error between reconstruction and robot). The precision of the 3D US volume reconstruction (robot-based or manual) compared to CT data must still be experimentally verified, although initial data suggests that the disagreement is negligible for the algorithms used here.

The volume reconstruction algorithm runtime is negligible; only the manual path delineation and the ultrasound scanning steps represent a deviation from the conventional procedure, but the additional time during the intervention (at less than five minutes overall) presumably does not pose a large obstacle.

Registration by Force-Following

A novel controller (Hybrid N/P control) for force following tasks was described, based on the concept of hybrid control. Using P control for slow, small scale motions together with N control for fast, large scale motions, the goals of accuracy and speed were combined in one hybrid controller. Switching between them was accomplished through linear combination of their outputs, governed by a sliding, absolute speed-based variable.

Experiments yielded a reasonably useful set of parameters (N control parameters like mass and friction, and Hybrid N/P control parameters like threshold speed and interpolation type). Furthermore, it was proved experimentally that the Hybrid N/P controller provided the advantages of both primitive controllers (accuracy, speed) without introducing their drawbacks (high force requirements, instability). Singularities of the robot kinematics are detected with

an *ex post* strategy of checking joint speeds and are handled by scaling the Cartesian tool tip speed down until rotational speed constraints are satisfied (not discussed here). Collisions that can lead to unstable behavior are detected with a simple force- and speed-based filter, thus avoiding jerky reactions (also not discussed).

Overall, this force-following approach is less than optimal for several reasons, however – it requires quite intimate knowledge of the system behaviour, the surgeon needs to interact physically with the robot, and it is not feasible to define many points due to time requirements ($15...25s$/point on average in earlier experiments). On the other hand, its accuracy for single point-to-point registration is around $0.5...1mm$, which is at the limit of the human ability to position the tool without additional (e.g. optical magnification) assistance, and close to the robot's precision.

Registration by Optical Tracking

The errors related to registration by optical tracking, tool calibration, and deformation during robotic milling operations have been investigated and quantified.

It has been shown that with the selected algorithms for OTS-based calibration and point definition (by sphere fitting), the achievable precision can be as high as $0.33mm$ for single markers (on top of the $0.05mm$ OTS repeatability accuracy), marking a lower limit for precision of OTS-based registration. When the robot is introduced into the loop, the errors grow, but remain small enough for the OTS to serve as dynamic registration insurance. For only three-point registration, the registration error of about $2...4mm$ may be sufficient only when interventions are supported by intraoperative sensors. In general, three-point registration is not a suitable approach with the OTS, as point definition with pointers is the single largest source of localization error in the system, but surface registration may be a viable alternative (which has not been explored in this work). Still, three-point registration may be sufficient as a first guess for maps from local sensors with more precise robot-based mapping.[13]

Calibration and Deformation

For the investigated system, an additional mechanical tool calibration step resulted in the detection and correction of translational offsets as high as $2.1mm$ from an uncalibrated initial state that was previously deemed to be sufficiently precise.

For 3D deformation, estimation functions have been determined, and experiments showed their relevance for high-precision milling, as deformations in real use were shown to become as large as $0.45mm$. Further experiments showed good linearity and independence of the single axis deformations. The actual deformation (or position error between actual and desired tool tip position) in the experimental stage reached up to $2.0mm$, whereas after subtraction of the estimated model-based deformation the remaining error dropped to a maximum of approx. $0.2mm$, due to a small remaining non-linearity of the deformation.

[13] Unfortunately, this finding – that stable localizers like robots are necessary for precise localization, especially under mechanical load – extends to the generalization of the local sensors approach to other, non-robot based systems. The most important of these approaches might be hand-held tools with OTS localization for the local sensor data provided by them (a generalization of the Navigated Control approach). Here, more research into viable localization is necessary.

Chapter 6

Navigation on Maps from Local Sensors

The basic applicability of local sensors – in terms of their ability to sense intervention-relevant environment features – has been shown in Chapter 4. Together with localizers, they serve to build maps from local sensors (MLS) as described in Chapter 5. The final step to close the navigation cycles based on these two building blocks is to perform navigation tasks on those maps (Figure 6.1), which are presented in this chapter.

In global sensing, maps are always complete when they enter the navigation processes; partial maps usually need not be considered. With local sensing, however, the situation changes and breaks down into two distinct cases: Global navigation on independently-built maps from local sensors (*indepMLS*), and local navigation on concurrently-built maps from local sensors (*conMLS*)[1]. In the first case, the indepMLS has been assembled completely before it is used, and the whole environment is modeled (Section 6.1). This allows the use of navigation algorithms already investigated with global sensors: In the presented system, 3D ultrasound maps can be part of the input for implant position optimization and/or subsequent steps like milling path planning. In the second case, the conMLS is being built concurrent to the intervention execution and can be used for intra-operative re-planning to increase safety (Section 6.2). Here, no equivalent precedent for navigation exists with maps from global sensors, so new navigation algorithms have to be developed to take advantage of the concurrent MLS' dynamic nature.

[1]Naturally, the two complementary cases – global navigation on *conMLS*, and local navigation on *indepMLS* – are possible as well, but do not take advantage of the relative benefits of both types of MLS (complete availability of *indepMLS*, and concurrent availability of *conMLS*). Therefore, they are not considered within the scope of this work.

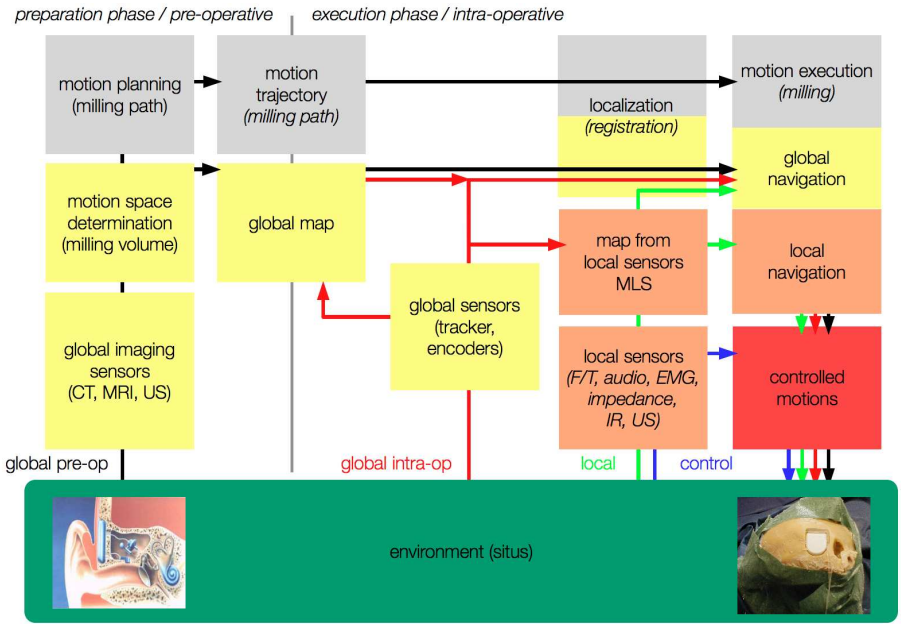


Figure 6.1: Relevance of navigation on maps from local sensors within the general navigation cycles framework (modified from [Henrich04]). Intra-operative global and local sensors together enable two new navigation cycles: global and local intra-operative navigation, partly on MLS.

6.1 Global Navigation on Maps from Local Sensors

According to the definition in Section 3.2, in global navigation motions are planned on pre-existing global maps. This principle has already been implemented for the CT-based individual milling path planning in previous work for the presented system [Waringo03a]. From that CT image, skull bone is segmented via a threshold operation, its surfaces are reconstructed along binary boundaries (where x-ray transmissive air or brain regions are adjacent to the bone) and used for implant position optimization [Waringo03b]. The bone volume itself is used for intersection determination and path computation [Waringo03c].

In this section, the same algorithms are investigated for their use with independently-built maps from local sensors (*indepMLS*).

6.1.1 Problem Formulation

With local sensors performing non-destructive sensing – like ultrasound – it is possible to construct maps exhibiting qualities which support the procedures described above during or shortly before the intervention. With only minimal changes to the algorithms whose practicality and feasibility has already been shown, the same procedures can be performed with 3D ultrasound maps, as will be described in the following.

Thus, the inputs to the global navigation on MLS problem are one or more global maps from local sensors[2] and a task description containing conditions and optimality criteria for successful completion of the given task. In the presented application, task completion is characterized by an optimal implant fit into the skull bone (i.e. only small protrusions out of the skull, little to no transgressions into the skull) and a complete-coverage path (*sweep*) within the implant or bone-implant intersection volume for the milling intervention itself.

The desired output consists of exactly this milling path, parametrized and located within the patient and within the robot workspace in such a way as to allow successful and safe execution.

[2]In this work, only the case for 3D ultrasound maps is considered. Global navigation on CT data was already investigated in [Waringo03a].

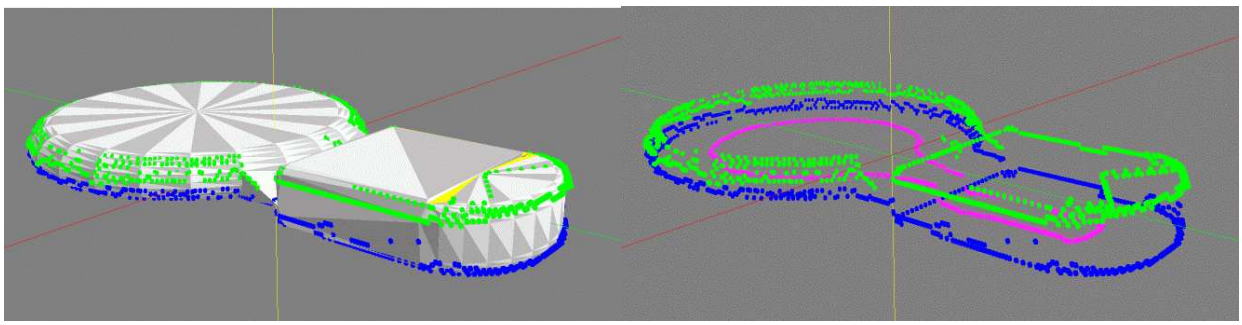


Figure 6.2: Implant volume (Soundbridge), with its three sets of optimization points overlaid onto the CAD surface model (left; right: only optimization points).

The robot thus needs to be registered with both the planning data and the patient in the OR. This can also be expressed as searching a path ${}^{rob}P$ (in robot coordinates, eventually) depending on the indepMLS M_{LS} acquired previously in tool coordinates ${}^{rob}T$ and the tool shape S:

$$ {}^{rob}P = f_{globalNav}({}^{rob}T, {}^{T}M_{LS}, S) $$

This corresponds to the R_{once} strategy in Section 1.3 (i.e. modification of the trajectory occurs only once at the beginning of the intervention in an independent step).

6.1.2 Global Navigation on indepMLS: 3D Ultrasound Maps

Based on the result of the map-building process with A-mode ultrasound probes (Section 5.3.1), several more navigation steps are performed on these maps now. These encompass localization (of the milling volume through an implant position optimization, in Section 6.1.2.1), possibly planning (of milling paths, in Section 6.1.2.2), and path execution (i.e. actual milling, in Section 6.1.2.3). The latter includes several safety-relevant control aspects which are discussed in this context here as well.

6.1.2.1 Implant Position Optimization

The implant first needs to be positioned within the reconstructed bone volume (cf. Section 5.3). In spite (or because) of the limited size of the scanned region and several functional constraints on suitable implant positions (regarding orientation and distance w.r.t. the ear canal etc.), determining a reasonably good 6D transformation to position the implant is non-trivial [Waringo03b]. The implant position optimization effectively constitutes the determination of the registration between bone (environment) and implant volume I, which in turn is defined in local "implant coordinates" which are positioned in the robot workspace via the detour of their location within the bone representation in ${}^{T}M_{LS}$, which has been sampled by the robot in its own tool reference coordinate system defined by T before.

One investigated optimization criterion is the proximity of "optimization points" (a geometric set describing salient locations with strong curvature, Figure 6.2) on the implant's upper side to the upper bone profile, i.e. how well it blends into the skull surface. One constraint is to accept only small transgressions of the lower profile to prevent breaking through the skull. This results in positions mostly oriented along the upper boundary (Figure 6.3). The localization precision of the lower boundary points, e.g. when generated by ultrasound, might be a problem for very thin bones; for future work it might become necessary to adapt the optimization's constraint criterion respecting the stochastic nature of the lower profile.

Depending on convergence criteria and optimization parameters (number of optimization points etc.), the runtime of this step is on the order of less than one minute. Since implant

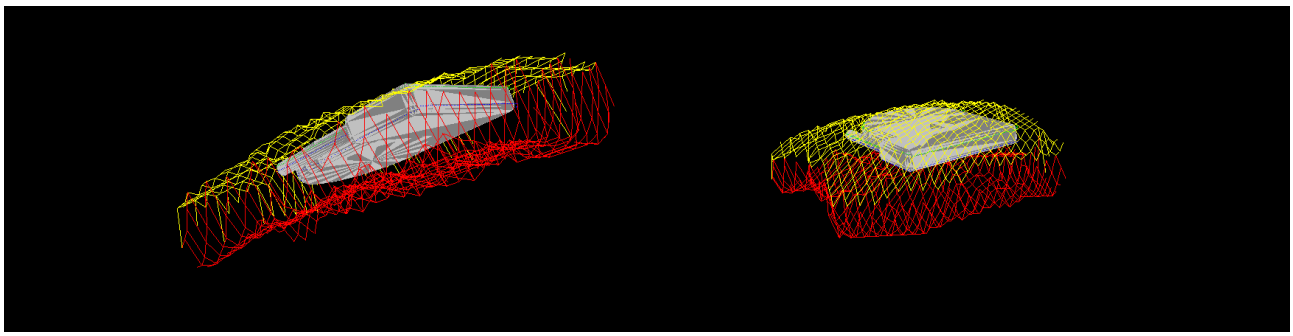


Figure 6.3: Implant volume (Combi40+), with position optimized within a reconstructed 3D ultrasound skull volume (left: seen from below, with no implant parts protruding below the lower skull surface; right: seen from above, with the implant top approximating the outer skull surface).

position planning takes place in the reconstructed US volume in an essentially continuous fashion, and the resulting implant position is transferred to the robot in a floating point format, there are no significant errors introduced here.

6.1.2.2 Milling Path Planning

Having determined the implant position within the bone volume, the last planning step is the computation of corresponding milling paths (investigated in [Waringo03c]). These are determined in the voxel representation and can be either generic (and thus pre-computable) or individual (with generally shorter path length based on reaming out only the intersection volume, Figure 6.5) and be computed using a multitude of different planning algorithms (Figure 6.4).

With path computation times on the order of a few minutes, it would be possible to compute individual milling paths in every case where imaging data (i.e. global maps) is available. However, for safety reasons it may be advisable to use pre-computed (and pre-approved) paths unless necessary to do otherwise, since visual inspection and validation of the paths by the operator under OR conditions and time pressure may be difficult, unsafe, or undesirable. Therefore, the presented system uses stored generic paths computed according to the parametrization detailed in Section 1.4.2. Furthermore, it is advisable to compute aggressive paths, i.e. such that ream out at least the intersection volume of implant volume and bone – this is non-trivial due to the spherical miller head shape of non-negligible diameter (both shape and diameter are path parameters).

Irrespective of the actual path planning algorithm and the decision on generic or individual paths, all computed milling paths have the property of "any-time capability". With the local sensing approach (and in general any real-time supervision of the system), it is possible to detect critical situations during milling plan execution, at which time it may become necessary to abort the robot-based intervention. Nevertheless, for a potential salvage operation it is important to leave behind a usable partial implant bed. This can be achieved by shaping the milling path so that the lower boundary of the milled implant bed corresponds exactly to the lower boundary of the implant at "sufficiently many" points in time during milling. Following this approach, the implant volume is effectively "sinking into" the skull bone (Figure 6.6).

6.1.2.3 Milling Intervention Execution

For the robot-based milling intervention starting now, the path executability is tested (similar to the ultrasound scan paths) before launch. This test consists of discretization of the vector-based cartesian path ${}^{rob}P$ (with empirically determined $\Delta p = 2mm$ translational and $\Delta\varphi = 7°$

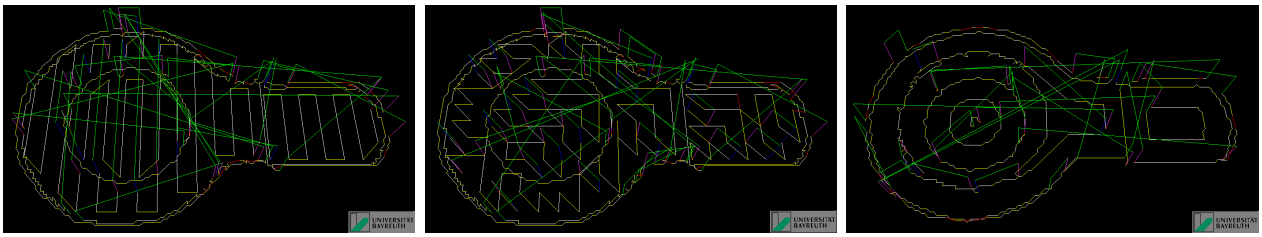


Figure 6.4: Generic milling paths planned for the Soundbridge implantable hearing aid using three different planning algorithms: parallel (left), potential-field-based, and concentric (right) paths. Different segment colors represent different segment types, associated with appropriate speeds (cf. Table 6.1).

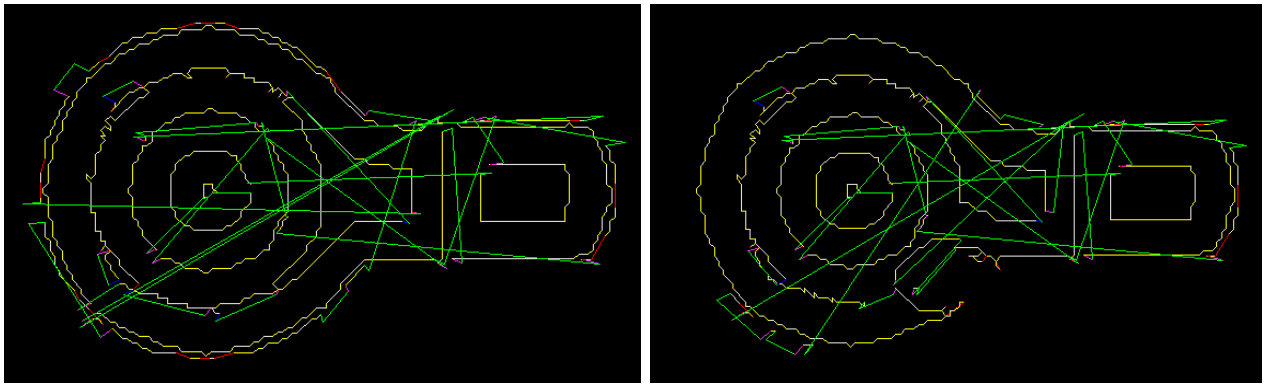


Figure 6.5: Individual concentric milling paths planned for the Soundbridge implantable hearing aid, with one sinking iteration shown (left: generic path; right: patient-individual path, with small circular region omitted at center-bottom, which corresponds to an avoided intrusion into the intracranial space).

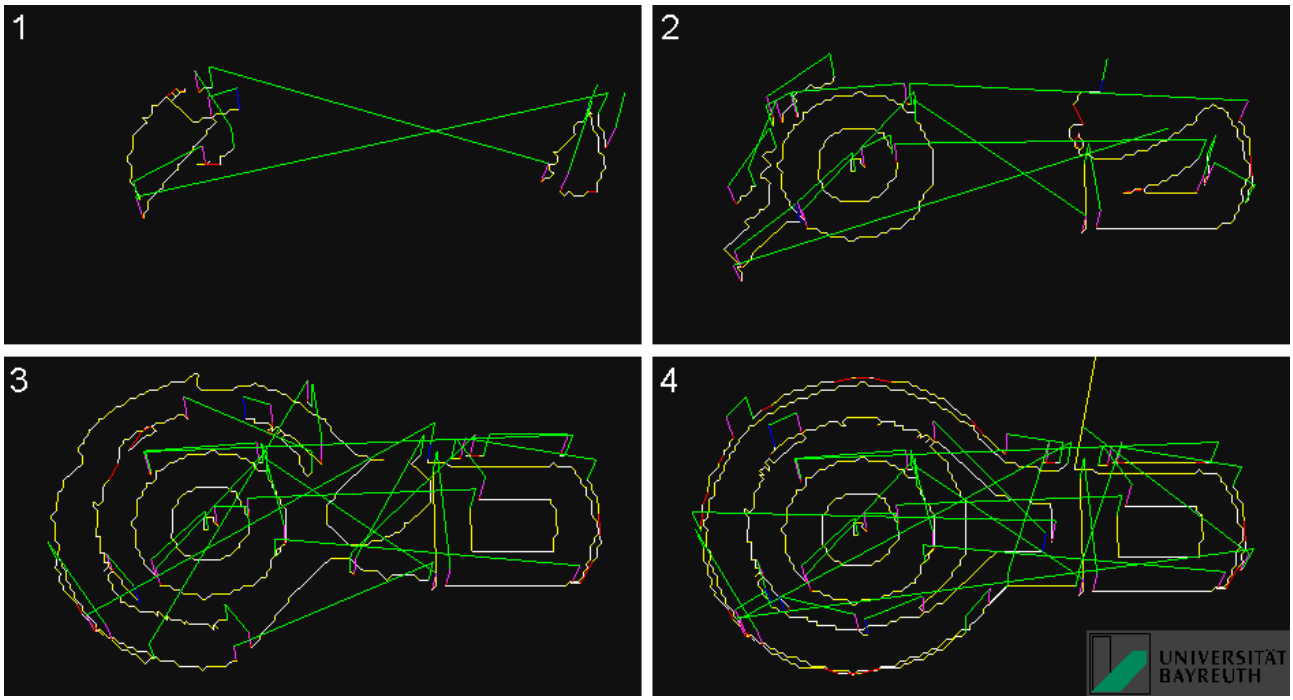


Figure 6.6: Successive individual concentric milling path iterations planned for the Soundbridge implantable hearing aid, with sinking iterations shown starting from first implant volume contact with the convex skull bone downwards (1 through 4). The "any-time" property ensures that at each step the implant is properly (if not fully) insertable.

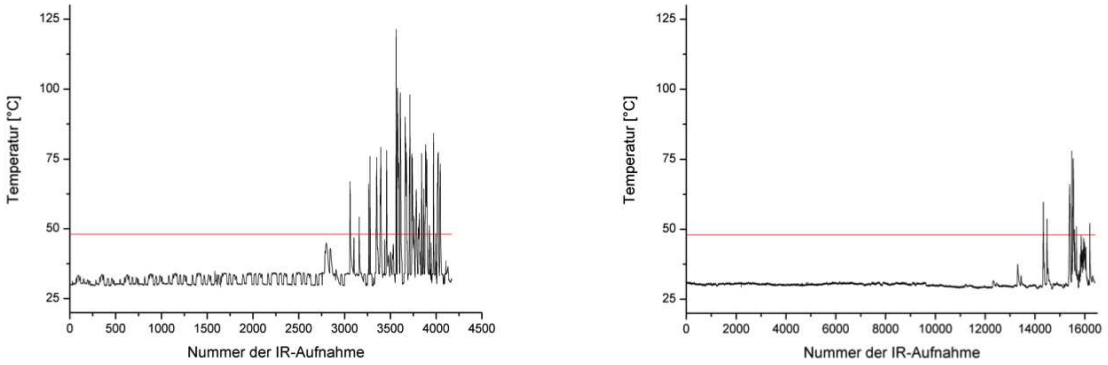


Figure 6.7: Graphs of maximum temperature spikes in each infrared (IR) frame occurring during bone milling (at $30,000rpm$ *rotation speed,* $24.9°C$ *specimen and coolant temperature; left: with constant speed of* $5mm/s$*; right: under force* P*-control at* $10N$*).*

rotational discretization)[3] into a higher-resolution path ${}^{rob}P'$ and inverse kinematic transformation according to the robot controller's internal kinematic model. This approximates the later execution joint trajectory $J_{({}^{rob}P)} = f_{IK}({}^{rob}P)$. This is the last instant where the system can reject paths which would exceed joint limits or enter kinematic problems, in particular the α_5 ("wrist-type") singularity (based on simple step-by-step limit testing).

The path execution with the miller as an ablative tool is supervised by two additional security measures aimed at increasing execution dependability and safety for the patient – force-based milling speed control (Section 6.1.2.3.1) and joint speed control (Section 6.1.2.3.2).

6.1.2.3.1 Force-Based Milling Speed Control Earlier research into bone response to milling and drilling operations in [Fuchsberger86] indicated a temperature/duration relationship governing the maximum duration of thermal exposition of vital bone before irreversible necrosis effects take place. Therefore, the milling process itself is closed-loop force-controlled to keep thermal trauma to the bone at a minimum [Stolka02]. The efficacy of force control was shown in earlier experiments by the author [Federspil03b] (Figure 6.8), where force spikes occurred more rarely, were of shorter duration, and were lower as opposed to fixed-speed milling (Figure 6.7).

The absolute force $\|F(t)\|$ measured by the force/torque sensor at the robot flange after subtraction of tool holder weight is used as control input for a proportional (P) control scheme of the tool speed (more precisely, its milling feed).[4] The control disturbance is computed as that force's deviation from the set point of $\|F\|_{target} = 10N$, with an empirically determined control gain $K_P = 1.4$. The effective tool speed factor[5] is then determined according to

[3]The rotational interpolation is based on quaternions $q = (s, \mathbf{v}) = \left(s, \begin{pmatrix} x & y & z \end{pmatrix}^T\right)$, which allow easy linear interpolation of rotations, unlike matrix, Euler angle, and many other rotation representations. The used Slerp interpolation ("spherical linear interpolation") allows equidistant spacing of rotations $q(t) = \frac{sin(1-t)\alpha}{sin\alpha} q_0 + \frac{sint\alpha}{sin\alpha} q_1$ between two quaternions q_0, q_1 with enclosed angle α [Shoemaker85].

[4]In the author's previous work [Stolka01], a "windowed" controller was used to react to forces deviating from a preset target value window (cf. Figure 3.2.4). In that setup, when absolute force exceeded an F_{high} threshold, the tool speed factor was reduced by a fixed multiplicative factor in each control cycle, and vice versa when absolute force fell below an F_{low} threshold, the tool speed factor was increased by the same factor. Although this scheme was successful in keeping forces (and resulting thermal necroses) within the defined window, the system behavior was very non-smooth, i.e. milling speed oscillated wildly between the extrema. A similar approach had been investigated by [Engel01] as well, with presumably similar results.

[5]A dimensionless value, also called the *monitor speed* in the robot control language V+. This value is limited to the range [0...100] by the controller system. Furthermore, the system programming clips it to the range between 2 (lower limit) and the initial user-defined preset at system startup (upper limit).

Figure 6.8: Experiment setup for comparison of constant-speed vs. force-controlled milling strategies. An IR temperature camera is aimed at the intervention region from below the skull preparation, as direct measurement of temperatures next to the miller tip is not possible due to the irrigation process.

Path Segment Type	$v_{segType}(p_i)[mm/s]$
bone, horizontal	4
bone, vertical	2
area change, upwards	50
area change, horizontal	50
area change, downwards	10
ultrasound scan	10

Table 6.1: Default milling speeds $v_{segType}(p_i)$ according to path segment type of point p_i.

$$v_P(t) = v_{global} - K_P(\|F(t)\| - \|F\|_{target})$$

The cartesian final tool speed $v_{tool}(t)$ corresponds to the default current path segment type speed $v_{segType}(p_i)$ of the current path point p_i (cf. Table 6.1) as given by the path definition, scaled with the scalar $v_P(t)$ factor:[6]

$$v_{tool}(t) = v_{segType}(p_i) \cdot v_P(t)$$

6.1.2.3.2 Joint Speed Control Furthermore, the tool speed is subjected to a second closed-loop controller based on robot joint speeds. Since the robot kinematic is a general serial chain, there exists no simple linear relationship between the cartesian tool speed $v_{tool}(t)$ and the resulting joint speeds $v_j(t)$ (for joints $j = 1, ..., 6$ in the presented system). However, to provide some safety to the operator and the patient during the robot-based autonomous intervention, it is necessary to limit the robot joint speeds. (Milling paths are checked for validity in advance, after registration with the patient, but unforeseen small-scale path modifications like approach/depart motions for milling pauses or local-navigation-induced evasion motions can still result in unchecked-for situations.) Although there are no linear $f_j()$ for which $v_j(t) = f_j(v_{tool}(t))$ would hold in general, such a simple relationship can still be assumed for small-scale linearizations. Then, both sides of the equation can be multiplied with

[6]Technically, this is implemented through concurrent adaptation of the monitor speed setting in the robot controller in a separate real-time task. In this high-frequency force reading task loop, a short-time absolute force average is computed which governs the P control. This is necessary to achieve a semi-continuous real-time speed change, unlike the discrete per-motion-segment speed change set by the $v_{segType}$ table entries.

Joint j	$\Delta\Theta_{jMax}[deg/s]$
1	25
2	25
3	25
4	45
5	45
6	45

Table 6.2: Maximum (empirical) joint speeds $\Delta\Theta_{jMax}[deg/s]$ for joints $j = 1, ..., 6$.

a scalar $s_{scale} \in \mathbb{R}$, leading to $s_{scale}v_j(t) = s_{scale}f_j(v_{tool}(t))$. This is exploited to superimpose a pre-emptive joint speed controller on top of the inverse kinematics.

Based on the assumption that the single joint trajectories are linear on a sufficiently small time scale (i.e. a linearization for small cartesian motions, on the scale of Δp and $\Delta\varphi$ as described above), each cartesian move is checked for a variety of error conditions (reachability, configuration changes, dead-man switch etc.) and its resulting joint speeds. In case of error, the move is aborted before execution; in case of excessive joint speeds (empirically determined values in Table 6.2), the cartesian speed is scaled down (by a scaling factor $0 < s_{scale} < 1.0$ determined according to Algorithm 6.1):

$$v_{tool}(t) = v_{segType}(p_i) \cdot v_P(t) \cdot s_{scale}$$

Algorithm 6.1 Move along a straight cartesian path between two points with joint-speed control

procedure MoveStraightCartesian($p_{start}, p_{goal}, v_{req}$)
  if p_{goal} NOT in range **then** ▷ goal is not within workspace
    error("out of range")
  end if
  $\Theta_{start} \leftarrow$ current joint values of p_{start}
  $\Theta_{goal} \leftarrow ({}^0J_6)^{-1}(p_{goal})$ ▷ joint angles of goal position
  if config(Θ_{start}) $\neq$ config(Θ_{goal}) **then**
    error("configuration change")
  end if
  $\Delta\Theta_j = |\Theta_{j,goal} - \Theta_{j,start}|$
  for $j \in \{1, ..., 6\}$ **do** ▷ calculate each joint's relative speed
    $s_{scale,j} \leftarrow \frac{\Delta\Theta_j}{\Delta\Theta_{jMax}}$
  end for
  $s_{scale} = min(1.0, \frac{1}{max_j(s_{scale,j})})$ ▷ find the maximum relative joint speed
  return s_{scale}
end procedure

6.2 Local Navigation on Maps from Local Sensors

During intervention execution, an incrementally growing representation of visited regions is kept in a concurrently-built map from local sensors (*conMLS*). This map changes over time to reflect the newly incoming local sensor data. To increase the safety of the complete system, it is indicated to use this information for navigation purposes. Algorithms for navigation on this data structure need to respect its volatility, which means it would be unreasonable to

compute complete paths in any time-consuming way based on such a changing map. Since the initial paths are any-time capable, i.e. their execution can in principle be aborted at any point of time, leaving a valid system state behind (i.e. a usable cavity in the investigated system), any replanning algorithm outputs are to conform to this requirement as well. In light of the inherent incompleteness of the map underlying navigation decisions, it is justified to classify such algorithms as local navigation procedures (cf. Figure 2.2).

In the following, the problem of local navigation on MLS will be formulated first (Section 6.2.1) and then examined in detail for the specific instantiation in the presented system (Section 6.2.2).

6.2.1 Problem Formulation

The challenge is to determine a valid milling path which at all times respects critical regions which might have been detected during intervention execution. However, since appropriately precise mapping with tool-based local sensors requires some overlap to perform optimal data entry localization (cf. Section 5.2.3), immediate consideration of new critical region information would result in permanent re-planning, low path quality due to insufficient data localization, and inferior real-time suitability of this navigation algorithm (this approach would be $R_{concurrent}$ in Section 1.3 – *concurrent evasion*, i.e. with trajectory modifications occurring continuously after each detection of dangerous regions). Therefore, one reasonable compromise between immediate system response, acceptable run-time, and safety is to trigger the re-planning navigation algorithms only upon completion of discrete sinking iterations (whose succession makes up the complete milling path, cf. Section 1.4.2; this corresponds to the $R_{iteration-wise}$ strategy in Section 1.3 – *iteration-wise evasion*, i.e. synchronized at several discrete points in time after every completion of one milling sinking iteration). Assuming perfect detection of critical contact states, this guarantees limits on their transgression in one dimension ("depth" along the Z axis of the local path coordinate system), determined by the sinking depth feed between single sinking iterations).

The given local navigation input is thus comprised of a map from local sensors M_{LS} which changes from call to call of the local navigation algorithms, and of a milling path $P = \langle p_0, ..., p_{n-1} \rangle, n \in \mathbb{N}$ where each path point p_i carries a point index $i \in \mathbb{N}$ and an iteration index $k \in \mathbb{N}$ which allow the identification of complete iterations and single points for navigational purposes. Path $P_k = \langle p_{k_0}, ..., p_{k_{n-1}} \rangle \subseteq P$ shall denote the sub-path of P containing all the (consecutive) path points p_{k_0} through $p_{k_{n-1}}$ corresponding to iteration k.

The sought output is a modified sub-path

$$^{T}P'_k = f_{localNav}(P, S, {}^{T}M_{LS}, k)$$

which is built by the local navigation function $f_{localNav}()$ off the original path P, but respects the critical regions detected in the passed conMLS M_{LS} according to that map's coverage by a tool shape S along the path in the iteration $k \in \mathbb{N}$. This modified sub-path is then merged back into P.[7]

[7]Note that – unlike for the global path ^{rob}P sought in the preceding section – the sought sub-path P'_k does not need to be checked for kinematic validity as stringently before execution (because it generally follows the original, pre-checked path, and the small deviations are unlikely to cause large changes to the trajectory beyond what was admitted by the initial checking). Therefore, it need not be computed or converted in robot coordinates directly, but the changes can be determined in local milling volume (or "implant") coordinates.

Note, however, that "unlikely" does not mean "impossible", i.e. the introduced modifications may well lead the robot into e.g. kinematic singularities or joint limits. This issue might be addressed in two ways in future work: one, pre-operatively by improving the global path checking procedure as described in Section 5.3.1.6 by maximizing the path's initial distance to said work space regions; and two, in real time by ensuring through joint speed control that the changes do not result in high-speed motions.

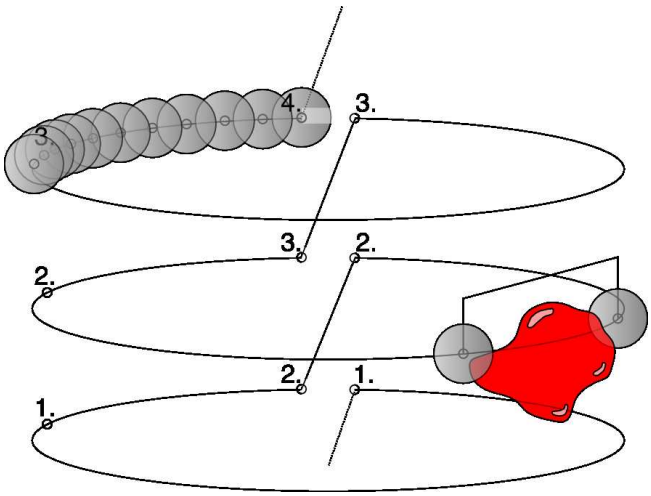


Figure 6.9: Pre-computed milling path, with sinking iteration indices (here numbers in decreasing order) appended to path points (only a selection is shown). In sinking iteration 2., a critical region (red) is intersected by the original path. The last and first non-intersection path point before and after the critical region are shown, with a linear evasion path indicated which replaces the offending segment.

In principle, two approaches to define $f_{localNav}$ are possible. The first would reframe the problem as a path planning task from the ground up, i.e. discard the input path P (or at least the iteration sub-path $P_k \subseteq P$) and reconstruct it on the limited free space as held in the map M_{LS} on each completion of one sinking iteration. While such iteration-wise path replacement would be possible in the framework of the presented system, it would imply costly replanning and possibly reduced safety (compared to paths which were pre-approved by the operator before the robot-based intervention). The second approach honors the existing path by only modifying relevant segments. The output is thus not a completely replanned path (not even on the constrained input space), but one where only segments (which are spanning critical regions in map) are modified according to a simple, explicit rule (Figures 6.9, 6.10).

6.2.2 Local Navigation on conMLS

To use concurrently-built maps from local sensors (MLS) for navigation, they must be read at discrete points of time and the planned trajectory needs to be related to them. To this end, the space covered by the tool (or its tool shape S) following the robot path is projected into the map M_{LS} in chunks according to the path points' iteration information. When the concurrent MLS M_{LS} is being filled during the intervention, this covered space and the map can be compared for overlap with critical regions.

In a first step, the robot path P needs to be discretized into a path P' so that long path sections between distant successive path points are broken up into smaller parts, which then serve as atomic navigational entities which are either traversed (for non-critical sections) or skipped (for critical ones). This step solves the problem of identifying the path exit and path re-entry points corresponding to a single evasion sequence (a segment of the modified output path due to critical region avoidance) by creating candidates for these points (path points *a)* and *d)* belonging to the discretized path P', both of which may or may not be part of the original path P). This discretization procedure is already done in the robot controller (for path reachability checking) before, so the same discretized path P' can be used for this current purpose, too.

In the presented system, since the ablative tool is a miller with a spherical head, this discretization can be based solely on cartesian distance between successive points, and it is

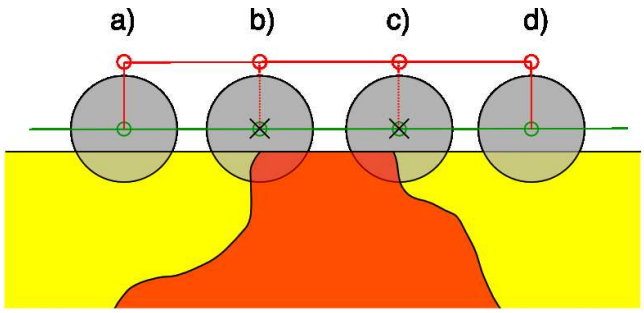


Figure 6.10: Local path modification based on path point lifting. The initial path P (green line) follows the bone volume below (yellow). When encountering critical regions (red, center), the corresponding path points are modified by lifting them (points b, c*). The nearest path points completely preceding and following such a modified section (points* a, d*) need to be modified accordingly to create a safe evasion path (red line).*

not necessary to interpolate the tool orientation.[8] During later execution of the robot path, navigation on the MLS M_{LS} is performed based on the critical regions contained therein. While these can be arbitrarily defined in general, for the investigated application they are determined by Terminal Fate Map (TFM) entries of type C_D (i.e. dura detections) and are to be avoided [Stolka07b]. When a tool shape positioned at a path point intersects with a critical region, that path point is to be excluded from traversal. The registration of the path P' and the MLS M_{LS} (necessary to project the pre-planned vector-based path into the occupancy grid map) is trivial (i.e. the identity transform), since both are defined in local coordinates of the milling volume.

The critical region avoidance is achieved by path point lifting in the presented Algorithm 6.2, according to the constraint introduced in Section 3.3.3 (not to re-plan robot paths, but to modify pre-computed ones to simplify the real-time intra-operative computations and to stick to the pre-operatively accepted path as closely as possible). This procedure is executed every time a new sinking iteration k of the pre-planned robot path P is entered, thereby satisfying the any-time property of the milling procedure at those instants of time.

This algorithmic description only holds for the idealized case of unconstrained accessibility of point p_j with the free-flying tool shape S. Although valid in the experiments described in this chapter, this is not true in a real system. Also, the algorithm includes the assumption of an obstacle-free retraction path up to a height of Δh above each path point p_j. To solve this, the map area checked for critical region intersection has to be extended, from covering just S at the position p_j to including the complete retraction path. With the tool head being at least the same width as the tool shaft, this can be accomplished by extending p_j to the line $l_j = \{p | p.x = p_j.x, p.y = p_j.y, p.z \geq p_j.z\}$ reaching upwards starting at p_j. In this case, the contact state check term in line 3 of Algorithm 6.2 simply becomes $(C_D \in M_{LS}|_{S*l_j})$, and intersections with critical regions along a generalized-cylinder-shaped volume are considered.

Another aspect to be taken into account is the absolute value of Δh. As the milling paths in the presented application are located in a bony volume approximately covering a (the lower) half-space of the path coordinate system, a Δh that reaches above the highest sinking iteration is sufficient to avoid collisions with other hard-obstacle areas for point modifications at all depths of the original path (obviously in special circumstances much smaller values can be

[8] This requirement differs from the discretization necessary for path traversability testing e.g. for milling path execution – as described in Section 6.1.2.3 – which respects the orientation as well in order to determine kinematic accessibility. Nevertheless, the same procedure can be used in each case.

However, there is a substantial restriction present in this formulation – the ablative tool consists only of its spherical head. This will rarely ever hold for any real tool. For milling tools, the rigid shaft extends upwards from the head, where the orientation of "upwards" depends on the tool orientation, determining tool accessibility. Thus, the following descriptions hold for an idealized tool, unless noted otherwise.

Algorithm 6.2 Modify motion path of the next iteration k based on MLS information

procedure MODIFYNEXTITERATIONPATH(P', S, M_{LS})
  for $p_j \in P_k$ **do** ▷ Determine critical path points in iteration k
    if ($p_j.segType \in \{bone\_milling\_segments\}$) && ($C_D \in M_{LS}|_{S*p_j}$) **then**
      $p_j.status \leftarrow$ "critical"
    end if
  end for
  for $p_j \in P_k$ **do** ▷ Modify/create evasion points
    if ($p_j.status ==$ "critical") **then** ▷ Situation b), c)
      $p_j.z \leftarrow p_j.z + \Delta h$
      if ($p_{j+1}.status \neq$ "critical") **then** ▷ Situation d)
        insert ($p'_{j+1} \leftarrow p_{j+1}$) into P' with ($p'_{j+1}.z \leftarrow p_{j+1}.z + \Delta h$)
      end if
    else
      if ($p_{j+1}.status ==$ "critical") **then** ▷ Situation a)
        insert ($p'_j \leftarrow p_j$) into P' with ($p'_j.z \leftarrow p_j.z + \Delta h$)
      end if
    end if
  end for
end procedure

chosen, but the proposed magnitude of evasion height is appropriate in the general case).

6.3 Experiments

In the following experiments, overall system performance was tested, i.e. based on the previously introduced local sensors and maps from local sensors (MLS), both global and local navigation algorithms $f_{globalNav}$ and $f_{localNav}$ were implemented either on the robotic system or in simulations, and the functional requirements were validated.

6.3.1 Validation of Global Navigation on 3D Ultrasound Maps

Milling based on 3D ultrasound maps (i.e. execution of the global navigation plan) was performed in human cadaver skull preparations and synthetic skull dummies, using the RX90/CASPAR setup (cf. Section A.2).

Milling duration for a typical implant volume is about 2...5min, depending on miller size, force-based speed control effects, and the sinking depth of the implant-bone intersection volume. Milling the implant cavity in the bone introduces errors through the robot's tool positioning, tool interaction (cutting/tearing bone; for an extreme example from an early system version cf. Figure 6.11), and deformations under load.

The latter have been determined in earlier experiments [Stolka06b] to stay below approx. 1.75mm for the force range dictated by thermal considerations and permitted by the force-based speed control. In actual operation, this value is smaller due to lower maximum forces (approx. 0.45mm, cf. Section 5.5.3.4.2). The final implant bed fits the implant to within approx. 1mm around its edges, which is most probably sufficient to allow complication-free healing. This error is a combination of tool deformation and motion of the region of interest due to "patient shift" from external forces on the skull preparation.

Furthermore, direct comparison of milling interventions based either on no preoperative planning data (Figure 6.12) or on robot-based 3D-US maps (Figure 6.13 on a synthetic dummy,

Figure 6.11: Dummy volumes from robot-based milling in a human skull preparation with an early system version (based on the RX130 setup). With constant-speed miller feed (without speed control), the variations in forces led to highly imprecise results, visible in particular along the outer ridge, where irregular chips broke off the bone surface. Also, it is apparent that force-following three-point registration is not sufficient to define a milling volume that does not cause breakthroughs into the skull interior (left volume).

Figure 6.14 on a human skull preparation) indicates a better implant bed fit in the skull for the latter, image-guided approach.[9]

6.3.2 Validation of Local Navigation on conMLS

The algorithm proposed in the preceding section has been implemented in a simulation[10] to validate its correctness independent of local sensor classification rates and robot positioning accuracy considerations.

An environment consisting of a thin "bone" (C_B) layer with "dura" (C_D) underneath and "air" (C_0) above was created (similar to Figure 5.32) and a milling path for an implantable hearing aid registered to it. The path was discretized and miller tool interaction was simulated along the sequence of path points, i.e. the plan was executed. This included contact state classification based on the classification dominance relation set D_C as described in Section 4.2.2, and entering this data into a concurrently-built MLS (*conMLS*). Upon finishing each sinking iteration, the tool shape was successively projected into the conMLS at the path point locations for the next iteration (which itself may consist of path segments at different depths), and the expected contact state was returned according to the dominance relation D_Σ as in Section 5.2.5 for each path point. For those points with critical contacts, their Z-coordinate was modified to avoid them in the upcoming iteration as described before.[11] The next iteration was then performed along the modified sub-path. During the whole path execution, overall contacts, critical contacts, and modified points were counted.

One remaining algorithmic parameter (apart from the process parameters like discretizations etc.) is the choice of the conMLS projection to be used in the map-reading step. From the three choices discussed in Section 5.2.1, the Earliest Information Map (*EIM*) would seem to be not a good choice, as it did not yet enjoy the benefits of the "Static Environment As-

[9] Note that due to limitations of the specimen fixation mechanism and the robot kinematics, it was not possible to create the implant cavities in anatomically correct positions (as seen in the figures). Nevertheless, the presented results can be transferred to interventions in the proper calvarial bone locations.

[10] In the *MapSpaceTest* software module.

[11] For the simulation it is not necessary to consider situations a) and d) as in Figure 6.10, since the tool does not actually traverse the space in between points. However, for a real robot-controlling implementation, this would need to be done.

Figure 6.12: Robot-based milling of a generic implant bed (left) in a human skull preparation for the Combi40+ geometry model without planning data, with the position defined exclusively by force-following registration (performed with the RX130 robot). With force-based speed control in place, the implant fits the later cavity within very tight limits (right). However, it is apparent that the implant position is suboptimal with a slight tilt to one side due to the lack of planning image data.

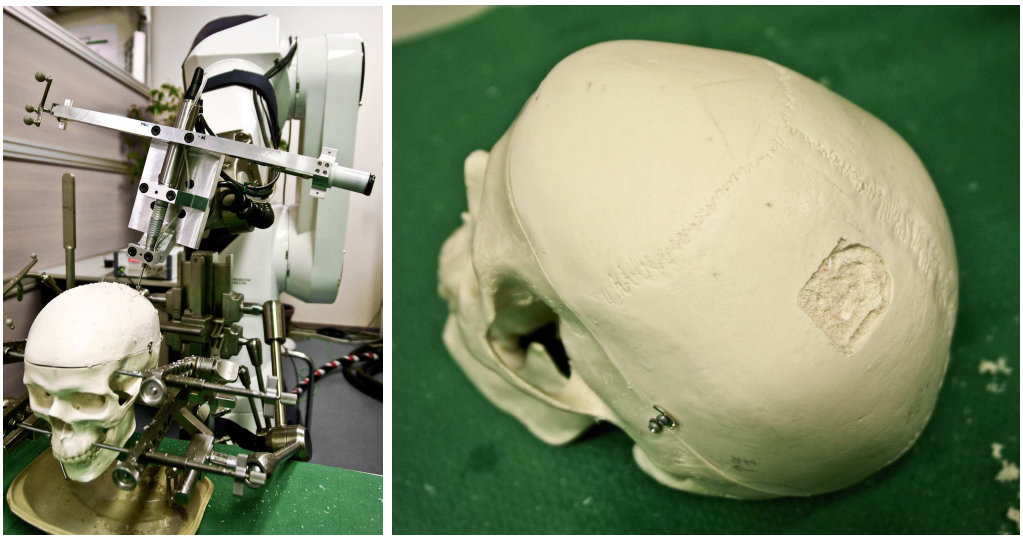


Figure 6.13: Robot-based milling of a generic implant bed for the Combi40+ geometry model (top left) based on a 3D ultrasound map (performed with the CASPAR/RX90 robot) in a synthetic skull dummy. Although the implant cavity is well-aligned with the upper skull surface and does not exhibit any breakthrough into the skull interior, milling in the plastic material produces detritus in the cavity making objective quality assessment and implant insertion difficult.

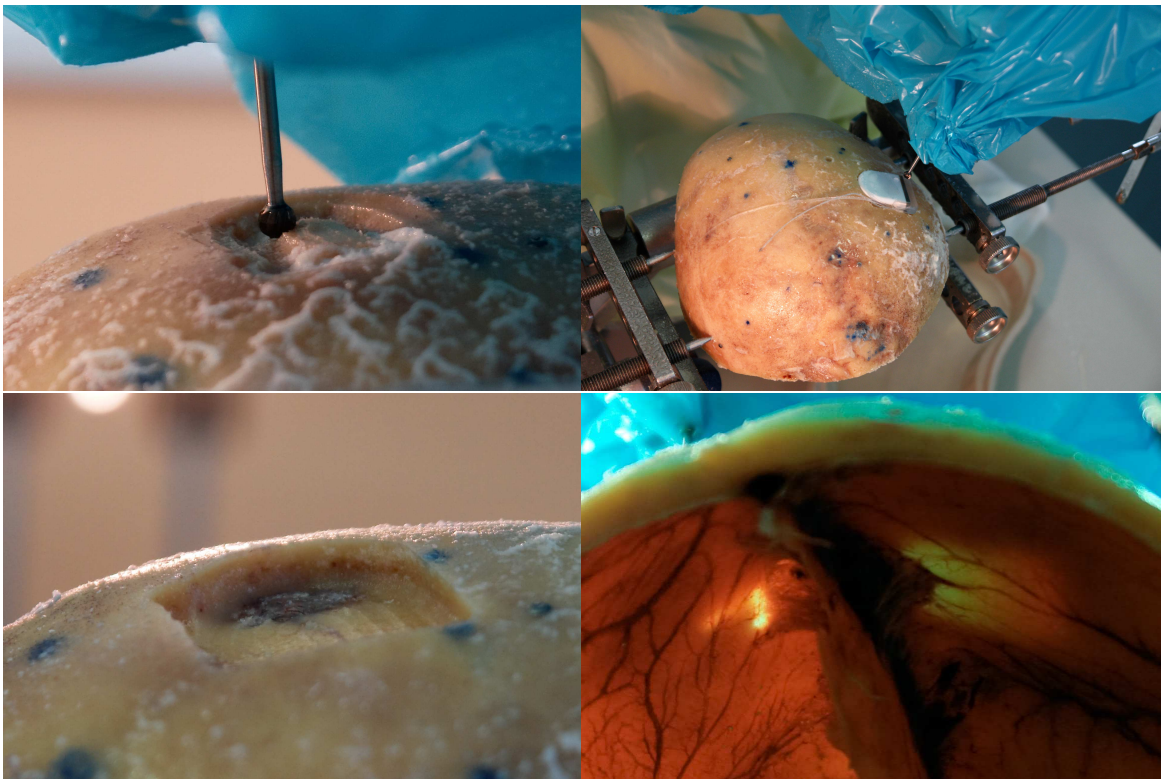


Figure 6.14: Robot-based milling of a generic implant bed in a human skull preparation for the Combi40+ geometry model (top left) based on a 3D ultrasound map (performed with the CASPAR/RX90 robot). The implant fits the later cavity within very tight limits (top right). After milling, the precise pre-operative positioning of the milling volume becomes apparent as the dura mater is just visible (bottom left). Viewed from below, the thin remaining bone layer lets light shine through, with the sensitive dura mater-perfusing blood vessels intact (bottom right).

conMLS	# contacts (critical)	# mod. points	iteration	removed C_B, C_D (missed%)
none	4440 (729)	0	n/a	39995, 5892
environment	3404 (0)	4042	4...5	39434, 0 (-1.4%, -100%)
EIM	4134 (476)	1137	5	39898, 2813 (-0.2%, -52%)
PTM	4134 (476)	1137	5	39898, 2813 (-0.2%, -52%)
TFM	4414 (703)	79	5	39995, 5792 (0%, -1.7%)

Table 6.3: Efficacy metrics for local navigation on conMLS for the described "well-registered" simulation setup; showing overall number of contacts (and critical contacts therein), modified points per sinking iteration, indices of modified sinking iterations, and number of removed voxels from "bony" and "dura" regions (with percentage changes compared to non-navigated milling).

sumption" (Section 5.2.3.2) from multiple passes through mapped locations. Then, deciding between Pre-Terminal Fate Maps (PTM) and Terminal Fate Maps (TFM) is less straightforward. The additional information still present in a PTM as compared to the TFM may allow better avoidance of critical regions, while a TFM may contain a more up-to-date view of the environment. Figure 5.31 is actually corroborating the choice of TFMs, given their generally lower rates of classification errors.

The simulation environment was of size $(50mm)^3$, with the bony layer extending from $z_{bone} = 25mm$ downwards for $d_{bone} = 5mm$. A Soundbridge milling path was computed for a miller diameter $d_{miller} = 4.5mm$ (z-feed $0.75mm$, resulting in six sinking iterations 0...5) and was then "well-registered" so its upper path points coincided almost perfectly with the bone surface (at $z_{upper} = 24.5mm$). With the lowest path points at $z_{min} = -3.76mm$ in path coordinates, the milling volume thus extended down to $z = z_{min} - \frac{d_{miller}}{2} = -6.01mm$ in path coordinates. The environment and the conMLS were discretized at $\Delta_{map} = 0.5mm$ and the path at $\Delta_{path} = 0.125mm$ (increasing the size from $|P| = 1401$ to $|P'| = 43500$ points).

As a "baseline" measurement for the upper (worst-case) limit, the procedure outlined above was performed without path modifications to determine critical region transgressions without local navigation, resulting in $n_C = 4440$ contacts, from which $n_D = 729$ were critical contacts (the latter of which should of course be minimized later under the influence of local navigation).

As a second baseline for the lower (best-case) limit, the procedure was performed with perfect environment knowledge (unrealistically using the actual simulated environment as the MLS) and path modification at each new iteration ($R_{iteration-wise}$, i.e. synchronized with the beginning of each new sinking iteration, cf. Section 1.3). As should be expected, the offending path points were modified, resulting in $n_C = 3404$ contacts with $n_D = 0$ critical contacts (1131 and 2911 modified points in iterations 4 and 5, respectively – i.e. all the previous transgressions have occurred only in those two iterations).

Finally, the actually constructed conMLS was used with its PTM and its TFM projections in two separate runs. Using the PTM projection resulted in $n_C = 4134$ contacts with $n_D = 476$ critical contacts (1137 modified points, all in iteration 5, i.e. transgressions actually occurred in iteration 4), whereas the TFM ended up with $n_C = 4414$ contacts with $n_D = 703$ critical contacts (only 79 modified points, also all in iteration 5) (cf. Table 6.3).

The runtimes of the additional path discretization and of each iteration-wise local navigation step were on the order of one second each (2.66Ghz Intel Core 2 Duo processor with 6GB RAM under Mac OS X 10.6, using around 330MB RAM as a 32bit executable).

Obviously, the use of a realistic conMLS with the proposed locNav algorithm improved the transgressions as compared with the no-mapping baseline situation. Surprisingly however, the use of the not-as-current (and therefore also less-optimized) conMLS PTM projection incurred substantially fewer critical region transgressions than the TFM, as the locNav algorithm had to rely on a more conservative estimation (with more critical region entries) of the actual

conMLS	# contacts (critical)	# mod. points	iteration	removed C_B, C_D (missed%)
none	5440 (2644)	0	n/a	39375, 19913
environment	2126 (8)	16074	1...5	36929, 1119 (-6.2%, -94.4%)
EIM	3490 (760)	8314	2...5	39271, 6133 (-0.3%, -69.2%)
PTM	3445 (726)	8575	1...5	39245, 6036 (-0.3%, -69.7%)
TFM	4928 (2158)	1875	3...5	39333, 18166 (-0.1%, -8.8%)

Table 6.4: Efficacy metrics for local navigation on conMLS for a simulation setup with $\Delta z = -2mm$ offset (cf. Table 6.3 for detailed field explanations). Note that even with perfect knowledge of the environment (second row), there are still trangressions into critical regions due to the timing of the path projection into the MLS (happening after the end of each sinking iteration, which might be too late to avoid transgressions in the very first sinking iteration 0).

environment. The TFM projection on the other hand contained fewer false-positive critical regions, but also prevented fewer path points from transgressing into as-yet-unknown parts of the environment. Also, use of the EIM projection in the locNav algorithm eventually proved to be in general almost as good as the PTM, indicating that multiple passes through the same map locations with their concomitant map updates did not improve the navigation outcome significantly.[12] To verify these findings, a second round of simulations was based on a mis-registration of $\Delta z = -2mm$, i.e. the milling volume protruding strongly into the critical region. Overall the above interpretation holds up in this case as well (cf. Table 6.4). Neither projection (EIM, PTM, or TFM) was able to completely avoid milling into dura regions.

However, this is a natural limitation of conMLS – they contain only information of the environment in regions that have already been passed, and provide very limited knowledge of regions beyond, depending on the actual local sensing principle used in building that map. For the "well-registered" case, from the six iterations, only the last one could be modified in the runs using the conMLS, i.e. all potential transgressions in iteration 4 actually had to occur to fill the conMLS with critical region information (cf. Figure 6.15).

As the goal of local navigation based on conMLS should be the maximization of avoided critical region segments under simultaneous minimization of missed bony regions, the best choice of MLS projection for navigation on the investigated volumes is the PTM (*Pre-Terminal Map*), closely followed by the EIM (*Earliest Information Map*).

6.4 Discussion

The main Hypothesis 3 in this chapter was

> *"The current maps, created according to the procedure laid out in the previous hypothesis, are valid input for the used path planning algorithms, and guarantee safe avoidance of forbidden regions under these algorithms during planning and execution with a robot-based system."*

Two approach examples to support this hypothesis were investigated: global navigation on tool-independent maps from local sensors (i.e. 3D ultrasound), and local navigation on concurrent maps from local sensors (i.e. maps using state classifications).

The results of global navigation based on 3D ultrasound maps show the feasibility of planning and executing cochlea implant bed preparations in the skull without either prior CT imaging or any additional registration procedures, representing a big step forward in image-based

[12] It is possible that this artifact is due to the properties of the milling path with stacked horizontal-parallel layers of path segments, which – together with the consequences of the static environment assumption (Section 5.2.3.2) – causes little actual difference between the EIM and PTM projections in this particular case.

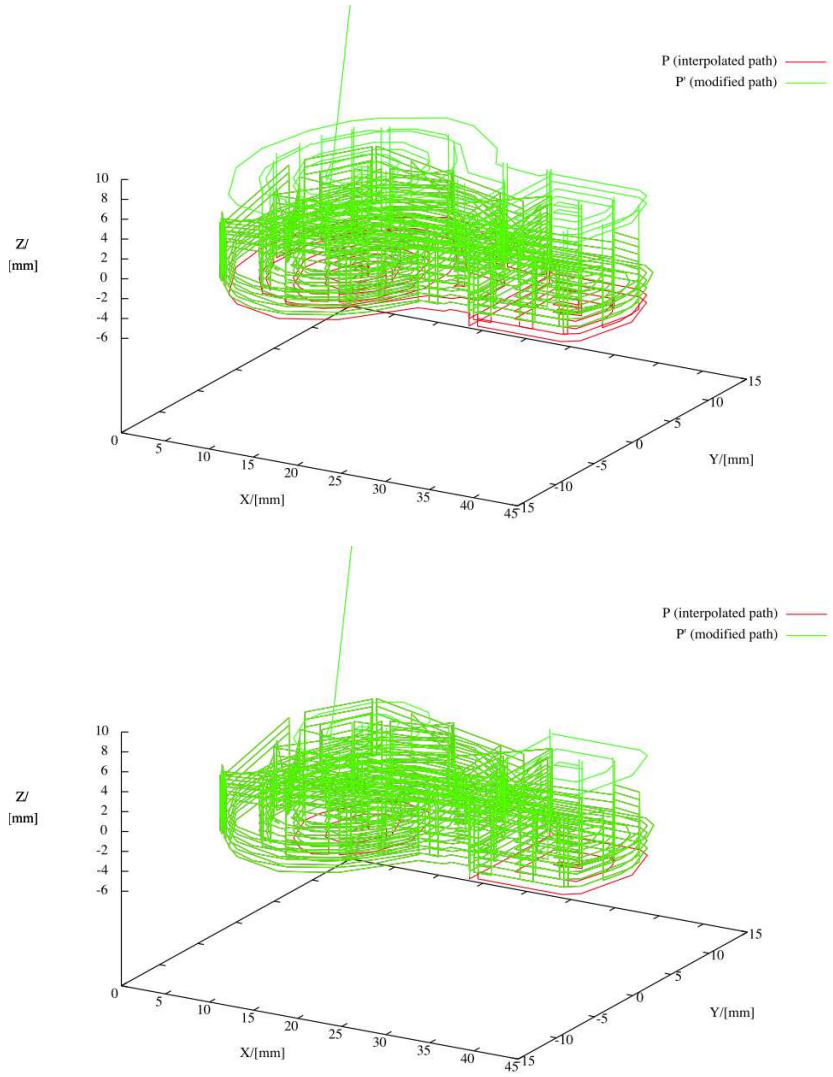


Figure 6.15: Generic milling paths for the Soundbridge implantable hearing aid as entered into a "well-registered" milling simulation (red) and as modified by the proposed local navigation algorithm (green). When the algorithm (unrealistically) has full knowledge of the environment, it can completely prevent transgressions into critical regions (left; critical regions below $z = -5mm$, with practically completely skipped iterations 4 and 5). For actual concurrently-built MLS, the limited knowledge allows fewer evading modifications (right; using a PTM projection).

ENT surgery. The 3D ultrasound map has thus been shown to be usable input for navigation purposes, especially when supported by control loops (during initial map building and later milling execution), meaning the overall precision of the complete intervention is sufficient to repeatably place an implant into an exactly-fitting bed in the thin skull bone.

The positioning of the intervention area within the robot workspace – i.e. the implant-to-patient registration – has been performed using two previously discussed approaches (cf. Figure 3.10): image-free intervention with three-point force-following registration, and planned intervention with a robot-based 3D-US map and optimization-based registration. It was shown that the image-based workflow resulted in safe and optimal implant bed creation, while the image-free approach suffered from positioning imprecisions.

To improve plan execution, further work might concentrate on optimization of the miller inclination based on achievable feed rates (to speed up the milling while at the same time reducing forces to generate less heat, deformation, and tear and wear) and other criteria like singularity avoidance, joint speed minimization, and configuration space limits (to increase the safety of the robot-based system w.r.t. its environment and reduce the probability of kinematics-related path abortions).

Regarding local navigation (locNav) on concurrent maps from local sensors (conMLS), it was possible to show that the use of conMLS reduced the amount of miller transgressions into critical regions by almost 70% compared to a "blind" approach, depending on the quality of the initial registration situation (or rather lack thereof, as worse starting conditions have more potential for locNav benefits). Although naturally no realistic projection of the conMLS allowed the local navigation algorithm to completely avoid transgressions, these results would merit inclusion of local navigation procedures in a future phantom study. The neglible computational overhead, together with the concurrent nature of both the proposed local sensing and the building of MLS, makes addition of these components and algorithms into research and clinical practice seem very promising.

Chapter 7

Conclusions

In this chapter, the findings of the present work are summarized (Section 7.1), their relation to the initial hypotheses and their significance are discussed (Section 7.2), and future research directions are outlined (Section 7.3).

7.1 Summary

The overarching question investigated in the present work was aimed at the utility of local sensors in medical robotics, and at the definition of suitable navigation algorithms built on top of such sensors.

As a validation demonstrator, a robotic surgical system was developed and investigated, starting from the medical application and the definition of the research question all through the implementation and experimental validation. In particular, the aim of the research presented here was the investigation of the validity of local-sensor-based navigation cycles in surgical robotics. To this end, several local sensors were introduced, then combined with localization methods to build maps from local sensors, and finally navigation algorithms were developed to leverage the particular benefits of those novel maps. All of these were validated using milling interventions at the lateral skull base, to create bone cavity beds for implantable hearing aids. The central requirement stemming from this choice of intervention was to honor the limited bone thickness to avoid intrusions into the skull interior while simultaneously creating exact cavities.

Overall, *the proposed system fulfills all the requirements* that were initially stated, and *is capable of performing safe, autonomous robotic milling interventions at the lateral skull base with the necessary accuracy by using maps from local sensors.* The three major steps along the way – towards a system that incorporates 3D ultrasound, surface maps, and concurrently-built maps and that can be used for CT-free implant position optimization and milling path planning – are summarized below.

Realizing the limitations of existing global sensors in medical imaging – first and foremost that imaging with them is usually static, intrusive, asynchronous, and/or expensive – the first hypothesis put forth maintained the existence of sensors that overcome these problems. This present work then first defined and explored the potential of a novel class of unobtrusive sensors (*local sensors*) that are limited in sensing range, but cheap enough in integration and sampling cost as to warrant investigation. Three new sensors were introduced – force/torque-based contact state classification, audio-based contact state classification, and high-precision A-mode ultrasonography – and their usefulness for subsequent navigation was established.

Second, navigation requires data structures that allow planning within environmental constraints; this being the proposal of the second hypothesis. Building on the previously investigated local sensors, this work then defined the concept of *maps from local sensors* (*MLS*)

and explored two major classes – independently-built MLS and concurrently-built MLS – based on ultrasound bone thickness measurements [Stolka07a] and contact-state classifications [Stolka05][Stolka07b].

Then, navigation algorithms benefitting from those MLS were developed to answer the third hypothesis stating that safe navigation with MLS support is possible. Leveraging the respective advantages of the two distinct kinds of MLS, two different navigation paradigms stood behind these developments. First, *global navigation on independently-built MLS* showed how the use of robot-based ultrasound helps with registration [Stolka03][Stolka07a], mapping, and planning of milling interventions when complete maps can be quickly acquired before milling execution begins [Stolka08]. Second, *local navigation on concurrently-built MLS* allowed the use of partial maps that contain only very limited, but extremely current information about the local ε-environment of the milling tool. Neither approach requires access to pre-operative imaging, although both allow high-precision and safe plan execution.

Finally, experiments – both on preparations as well as simulations, for training and testing – validated the concepts introduced in the present work.

7.2 Discussion

In summary, all three hypotheses have been answered in the affirmative:

> *"There exist local sensors, applicable for surgical milling interventions, which satisfy the evaluation criteria for sensors [as formulated [in Section 3.3.1.2]]."*
>
> *"Faithful and current maps of relevant features in the intervention area of surgical milling interventions can be built concurrently with certain local sensors and additional localizers."*
>
> *"The current maps, created according to the procedure laid out in the previous hypothesis, are valid input for the used path planning algorithms, and guarantee safe avoidance of forbidden regions under these algorithms during planning and execution with a robot-based system."*

The introduced local sensors satisfy the initial design requirements regarding discriminatory power, reliability of classification/accuracy, and data rate/latency. The system's localization is reliable (based on an industrial/medical robot) and precise enough to allow the safe creation of the desired bone cavities. (Applications with even higher precision requirements could leverage the findings about the linear deformation behavior [Stolka06b] that was used only analytically in the presented work, but could potentially inform the design of future control algorithms.) The resulting maps exhibit classification rates that are asymptotically improving with the fraction of explored space (for conMLS). In the investigated scenarios, they are precise enough to serve as input to both global and local navigation algorithms. These allow the complete avoidance of calotte breakthroughs (for global navigation) or the reduction of critical region transgressions by 50...70% (for local navigation).

The main contributions to various fields of science can be described as follows:

- to Medical Robotics:
  - *Contact states* (together with system states) relevant to milling applications were defined. Although they target robotic systems, manual interventions follow the same contact states. These states can provide e.g. a data source for surgeon skill evaluation complementary to motion-based "surgemes" [Reiley09].
  - Sensor data is returned according to *data classification dominance relations*, depending on the particular tool. These relations were formalized in the present work.

  - Local sensors provide immediate and non-intrusive sampling of relevant information from the region of intervention. Apart from the robot-based navigational uses presented in this work where a "feeling" robot is realized, they can be used in conjunction with intelligent tools (as an extension to Navigated Control) as well.
  - The concept of *navigation cycles* in surgical robotics was formalized and four distinct cycles were introduced.

- to Robotics:

  - This work introduced a new, highly reactive *local navigation* principle based on incomplete, semi-static maps (the maps from local sensors) between purely reactive control and navigation on complete static or dynamic maps.
  - To perform navigation on partial/partially annotated maps, existing complete-coverage path planning was extended with the definition of *adaptable path segments to replan path sections*.
  - A registration method was introduced to define a 6D transformation in a three-point procedure with the user guiding the tool tip to patient points in a *force-following controller* scheme. This is facilitated with a flexible "N/P" controller to allow both swift large-scale translations and very precise small-scale approaches.
  - A *distributed, extensible surgical robotics architecture* encompassing modules for sensing, robot representation and control, and user interface and flow logic control, together with a module communications infrastructure, was developed and one specific design was instantiated, the RONAF system.

- to Computing Science:

  - For map-building, a *sparse 4D map data structure* (for the MLS) was defined, allowing the semi-continuous modification of rasterized environment representations with change recording. This abstract data type has associated read/write functions $q_M()$ and $b_M()$ for map querying and map building. Writing with $b_M()$ respects both *optimized point spread functions* and *data classification dominance relations* to improve data localisation in the map. Reading with $q_M()$ on the other hand honors *critical fusion dominance relations* to return the most relevant covered map entry. Furthermore temporal projections $\tau(t)$ of the map data structure were defined to reduce the map dimensionality for planning purposes, with the three special cases *Earliest Information Map* (EIM), *Pre-Terminal Map* (PTM), and *Terminal Fate Map* (TFM) meriting special attention.

7.3 Future Work

Future work can be divided up as heading into two directions – either starting off from the proposed system and improving it, or moving the results beyond milling applications or even surgical robotics into new fields.

7.3.1 System Development

There is some room for future improvement, both within the system itself and in terms of comparative evaluation.

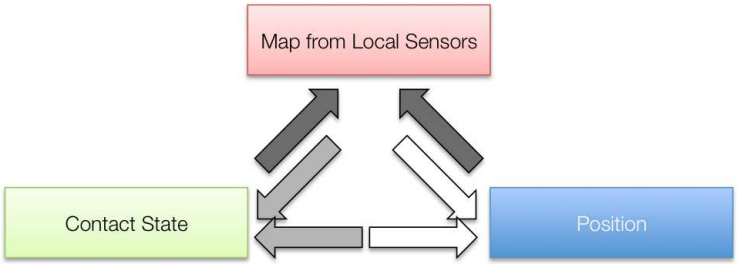


Figure 7.1: Information flow in the described system (dark arrows), alternatives (white and grey arrows).

On the most general note, the information flow in the proposed system (Figure 7.1) is strictly bottom-up, i.e. results in a uni-directional composition of information from the sensors and the tool position into the map (dark arrows), influencing the executed tool trajectory. However, there is reason to believe that e.g. the contact state classification could be improved by integrating prior knowledge about the environment. This could be done by a cooperative classification scheme, using the position information together with environment knowledge extracted from the current map, thus adapting the classifier result hypotheses (medium-gray arrows). Similarly, the currently classified contact state and the current map together can serve to verify position accuracy (white arrows) – if there is a high probability for a contact state which is impossible based on knowledge from the map, the position information should not be trusted.

Dynamic registration (tracking) between camera, robot base, and the patient would need to be integrated to cope with sudden shifts of tracked objects or the optical tracking system (*OTS*) by controlling the robot position appropriately. This would also allow comparative measurements of deformation based on both force and OTS.

It will also be necessary to ensure synchronization between sensors and localizers, as currently the system is based on the assumption that latency differences and jitter for different data streams and machines are negligible.

Furthermore, milling path optimization with respect to variable probe orientations (or other open DOFs) will be necessary for reliable and optimal milling, as there may be paths that are not traversable with constant values for the open DOFs, or which can be improved on a point-to-point basis with respect to joint velocities, collision avoidance, milling feed rate, exerted forces, thermal load, or other criteria.

7.3.2 Other Applications

The presented development can find extended use in other applications as well. First and foremost, the investigated medical application – milling of an implant bed for implantable hearing aids – consists of several steps (Section 1.4.1), only one of which was dealt with here. To insert the electrode into the cochlea or to attach the mechanical exciter to the ossicles, it is necessary to perform a mastoidectomy for access to the inner ear as well as potentially perform a cochleostomy to open up the cochlea for insertion. Both steps operate in highly critical anatomical regions (with the facial nerve and the cochlear wall itself being structures to approach cautiously) where local sensing based on electromyography (EMG) and force/torque may prove useful.

Also, automatic drilling cancellation upon calotte penetration during trepanations seems feasible with the proposed methods. Even without positional navigation, handheld tools could be augmented to control instrument power accordingly with the contact state (cf. the Navigated Control concept relying only on static pre-operative data). Robotic craniotomies (e.g. in the RobaCKa system) would benefit from depth control along pre-planned paths without the

need for an inflexible mechanical sheath preventing the milling tool from intruding into the cranial interior. Laser-based cochleostomy systems (again cf. RobaCKa) could move away from purely optical or fluorescency vision approaches and include acoustic contact state sensing with acoustics-based MLS to optimize the pulsed laser drilling procedure.

Within the field of surgical robotics, it might be possible to adapt the presented robot-based 3D ultrasound procedure and algorithms to other milling interventions. One example could be the transcutaneous 3D ultrasound acquisition of the femur for hip endoprosthesis implantations, maybe supported by X-ray images to help the surgeon judge the expected bone density before planning the implant position, but otherwise making marker implantation or CT acquisition obsolete.

While the presented system is based on a surgical robot, it can easily be used with handheld tools tracked by a localizer as well. The use of handheld ”smart tools” promises safety improvements, especially when used in conjunction with map-based tool power output control systems like Navigated Control.

Moving far beyond surgical robotics, the generalization of the local sensing paradigm to other fields like sculpting or archaeologic restoration (freeing artifacts of baked-on mud etc.) in ”automated archaeology” could be possible.

On a longer time scale, in particular local sensing and sparse 4D maps could be expected to see wider adoption in general in robotics applications.

Bibliography

[Amenta01] Amenta, N.; Choi, S.; Kolluri, R.K.: "The Power Crust", Proceedings of 6th ACM Symposium on Solid Modeling, pages 249-260, 2001

[Amstutz03] Amstutz C, Caversaccio M, Kowal J, Bächler R, Nolte LP, Häusler R, Styner M: "A-mode ultrasound-based registration in computer-aided surgery of the skull". Otolaryngology Head Neck Surg 2003; 129:1310-1316, 2003

[Arun87] Arun, K.S.; Huang, T.S.; Blostein, S.D.: "Least-Squares Fitting of Two 3-D Point Sets", IEEE Transactions on Pattern Analysis and Machine Intelligence, Sept. 1987

[Bale97] Bale RJ, Vogele M et al: "VBH head holder to improve frameless stereotactic brachytherapy of cranial tumors". Comput Aided Surg. 1997;2(5):286-91, 1997

[Baron10] S Baron, H Eilers, B Munske, J L Toennies, R Balachandran, R F Labadie, T Ortmaier, and R J Webster III: "Percutaneous inner-ear access via an image-guided industrial robot system". In: Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine, Volume 224, Number 5 / 2010

[Boctor10] E.M. Boctor, **Ph.J. Stolka** et al: "Precisely shaped acoustic ablation of tumors utilizing steerable needle and 3D ultrasound image guidance". In: Proceedings of SPIE Medical Imaging 2010 (San Diego, CA/USA)

[Boesnach04] Boesnach, I.; Hahn, M.; Moldenhauer, J.; Beth, Th.; Spetzger, U.: "Analysis of Drill Sound in Spine Surgery". In: Proceedings of MRNV 2004 – Medical Robotics, Navigation and Visualisation, Remagen. 2004

[Boisvert08] J. Boisvert, D. Gobbi, S. Vikal, R. Rohling, G. Fichtinger, P. Abolmaesumi: "An Open-Source Solution for Interactive Acquisition, Processing and Transfer of Interventional Ultrasound Images". Insight Journal, 2008

[brain-surgery.com] "History of Brain Surgery", from http://www.brain-surgery.com/history.html, accessed Dec 2009

[Brett07] Brett, P., Taylor, R., Proops, D., Coulson, C., Reid, A., Griffiths, M: "A surgical robot for cochleostomy". In: Proceedings of 29th Annual International Conference of the IEEE, Engineering in Medicine and Biology Society, EMBS 2007, Lyon, France, August 2007, pp. 1229–1232

[Brunke07] S.S. Brunke, M.F. Insana, J.J. Dahl, C. Hansen, M. Ashfaq, H. Ermert: "An Ultrasound Research Interface for a Clinical System". IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 54, no. 1, January 2007

[Bumm05] Bumm K, Wurm J, Rachinger J, et al.: "An automated robotic approach with redundant navigation for minimal invasive extended transsphenoidal skull base surgery". Journal of Minimally Invasive Neurosurgery 2005; 48(3): 159 – 164, 2005

[Caversaccio09] M. Caversaccio, C. Stieger, S. Weber, R. Häusler, L.-P. Nolte: "Navigation und Robotik an der Otobasis". HNO 2009, 57:975–982

[ClaronInc] Claron Technology Inc., http://www.clarontech.com/measurement_technology.php, accessed Jun 2010

[CISST] https://trac.lcsr.jhu.edu/cisst, accessed May 2011

[Coulson07] Coulson, C.J.; Reid, A.P.; Proops, D.W.; Brett, P.N.: "ENT challenges at the small scale". In: International Journal of Medical Robotics and Computer Assisted Surgery, 3: 91-96, 2007

[Dario03] Dario, P.; Hannaford, B.; Menciassi, A.: "Smart Surgical Tools and Augmenting Devices'. IEEE Transactions on Robotics and Automation, vol. 19, no. 5, Oct. 2003

[Davies00] Davies, B.: "A review of robotics in surgery". Proceedings / Institutional Mechanical Engineers, Vol. 214, Part H. 2000

[Davy02] M. Davy, S.J. Godsill: "Audio Information Retrieval: A Bibliographical Study". Technical Report, Cambridge University Engineering Department, 2002

[delaFuente04] M. De la Fuente, M. Hahndorff, T. Dahmen, S. Weltgen, D.C. Wirtz, F. Portheine, K. Radermacher: "Freehand Navigation using longshaft tools – comparing a force sensor based deflection compensation with an intramedular support device". In: Proceedings of CAOS International, 2004

[Denis03] K. Denis, A. Ranftl, G. v. Ham, et al: "Comparison of Registration Procedures of the Tibia in Robot-Assisted Total Knee Arthroplasty". In: Proceedings of MICCAI 2003, Montréal/Canada, November 2003

[Dragomatz97] D. Dragomatz, S. Mann: "A classified bibliography of literature on NC milling path generation". Computer-Aided Design, vol. 29, pp. 239–247, 1997

[Drouet02] P. Drouet, S. Dubowsky, C. Mavroidis: "Compensation of Geometric and Elastic Deflection Errors in Large Manipulators based on Experimental Measurements: Application to a High Accuracy Medical Manipulator". Robotica, vol 20, No. 3, 2002

[DudaHartStork02] R.O. Duda, P.E. Hart, D.G. Stork: "Pattern Classification". Wiley-Interscience; 2nd edition, October 2000

[Eggers05] G. Eggers, C. Wirtz, W. Korb, D. Engel, O. Schorr, B. Kotrikova, J. Raczkowsky, H. Wörn, J. Mühling, S. Hassfeld, R. Marmulla: "Robot−Assisted Craniotomy". Minimally Invasive Neurosurgery 2005; 48: 154–158

[Engel01] Engel, D., Raczkowsky, J., Wörn, H.: "A safe robot system for craniofacial surgery". In: Proceedings of ICRA. IEEE International Conference on Robotics and Automation, 2001

[Engel02] Engel, D.; Raczkowsky, J.; Wörn, H.: "Sensor-aided Milling with a Surgical Robot System". In: Proceedings of CARS (Computer Assisted Radiology and Surgery), Paris/France, 2002

[Engel03] D. Engel, W. Korb, J. Raczkowsky, S. Hassfeld, H. Woern: "Location Decision for a robot milling complex trajectories in craniofacial surgery". In: Proceedings of CARS 2003: Computer Assisted Radiology and Surgery, 2003

[Engelhardt06] M. Engelhardt, P. Bast, N. Jeblink, W. Lauer, A. Popovic, H. Eufinger, M. Scholz, A. Christmann, A. Harders, K. Radermacher, K. Schmieder: "Analysis of Surgical Management of Calvarial Tumours and First Results of a Newly Designed Robotic Trepanation System". Minimally Invasive Neurosurgery 2006; 49: 98–103

[Fabian01] C.M. Fabian, K.N. Ballu, J.A. Hossack, T.N. Blalock, W.F. Walker: "Development of a parallel acquisition system for ultrasound research". SPIE, Vol. 4325, 54, 2001

[Federspil03] Ph. A. Federspil, U. W. Geisthoff, D. Henrich, P. K. Plinkert, "Development of the First Force-Controlled Robot for Otoneurosurgery". The Laryngoscope, Lippincott Williams & Wilkins, Inc., Philadelphia, 2003

[Federspil03b] Federspil, P.A.; De Mola, C.; Geisthoff, U.W.; Henrich, D.; Plinkert, P.K.; **Stolka, Ph.**: "Force Feedback Control of Robot's Speed Prevents Heat Trauma in Robotic Milling at the Lateral Skull Base". In: Proceedings of CURAC 2003, Nürnberg/Germany, 2003

[Federspil03c] Ph. A. Federspil, S. H. Tretbar, U. Geisthoff, B. Plinkert, P. K. Plinkert: "Ultrasound Based Navigation of Robotic Drilling at the Lateral Skull Base". In: Proceedings of CARS 2003, 2003

[Federspil05] Federspil, P.A.; Tretbar, S.H.; Plinkert, P.K.: "Increase the Accuracy in Navigated Surgery by in site Measurement of individual Sound Velocity in Skull Bone". In: Proceedings of CURAC 2005, Berlin/Germany, September 2005

[Federspil06] Federspil, P.A.; Tretbar, S.H.; Sittel, C.; Plinkert, P.K.: "SonoPointer – a prototype system for ultrasound scanning of skull bone thickness", AORL 2006 (in press)

[Federspil10] Federspil, Philipp A.; Tretbar, Steffen H.; Böhlen, Friederike H.; Rohde, Stefan; Glaser, Simon; Plinkert, Peter K.: "Measurement of Skull Bone Thickness for Bone-Anchored Hearing Aids: An Experimental Study Comparing Both a Novel Ultrasound System (SonoPointer) and Computed Tomographic Scanning to Mechanical Measurements", Otology & Neurotology, 28 January 2010

[FFTW] "FFTW – Fastest Fourier Transform in the West". http://www.fftw.org/, accessed May 2011

[Fuchsberger86] Fuchsberger, A.: "Untersuchung der spanenden Bearbeitung von Knochen". iwb Forschungsberichte Band 2, Techn. Univ. München/Institut für Werkzeugmaschinen und Betriebswissenschaften, 1986

[Guthart00] Guthart, G.S.; Salisbury, J.K.: "The intuitive telesurgery system: overview and application". In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA2000) (San Francisco 2000) pp. 618–621

[Harrer07] Harrer, Michael: "RONAF: Gravitationskompensation und Deformationsschätzung in der Chirurgierobotik (Gravitation compensation and deformation estimation in surgical robotics)". Bachelor Thesis, Computing Science, Universität Bayreuth, 2007

[Harris78] Harris, F. J.: "On the Use of Windows for the Harmonic Analysis with the Discrete Fourier Transform". In: Proceedings of the IEEE, Vol. 66, No. 1, Jan 1978

[Hein02] Hein A., Kneissler M., Mätzig M., Lüth T.: "Navigated Control – Ein neuer Ansatz für das exakte Fräsen". In: Proceedings of CURAC 2002, Leipzig/Germany, 2002

[Henrich02] Henrich D.; Plinkert, P.K.; Federspil, Ph.A.; Plinkert, B.: "Robotergestütztes Fräsen an der lateralen Schädelbasis: Kraft-basierte lokale Navigation bei der Implantatbettanlage". VDI-Bericht 1679 – Tagungshandbuch zur Robotik 2002, Ludwigsburg/Germany, 2002

[Henrich04] Henrich D.; **Stolka Ph.**: "Navigation Principles in Surgical Robotics". In: Proceedings of MRNV 2004 – Medical Robotics, Navigation and Visualisation. Remagen/Germany, 2004

[Intuitive] Intuitive Surgical: "Frequently Asked Questions". http://www.intuitivesurgical.com/products/products_faq.html#1, accessed May 2011

[IGSTK] K. Cleary, P. Cheng, A. Enquobahrie, Z. Yaniv (eds.): "IGSTK: The Book". http://www.igstk.org, accessed May 2011

[ITK] http://www.itk.org, accessed May 2011

[Jan04] U. v. Jan, D. Sandkühler, L. Kirsch, H.-M. Overhoff: "Definition of a humerus coordinate system from semiautomatically segmented 3-D ultrasound volumes". Biomedizinische Technik (49) Ergänzungsband, Schiele&Schön, Berlin, 2004

[JR3] JR3: "90M31A Sensor Overview", "50M31A Sensor Overview". http://www.jr3.com/sensors-m-series.html, accessed May 2011

[Kahrs09] L.A. Kahrs: "Bildverarbeitungsunterstützte Laserknochenablation am humanen Felsenbein". Dissertation, Universität Karlsruhe, 2009

[Kahrs10] L.A. Kahrs, J. Burgner, T. Klenzner, J. Raczkowsky, J. Schipper, H. Wörn: "Planning and simulation of microsurgical laser bone ablation". Int J CARS (2010) 5:155–162

[Kapoor08] Kapoor, A.; Taylor, R.H.: "A constrained optimization approach to virtual fixtures for multi-handed tasks". In: Proceedings of IEEE International Conference on Robotics and Automation (ICRA) 2008

[Kazanzides95] P. Kazanzides, B.D. Mittelstadt et al: "An integrated system for cementless hip replacement". IEEE Engineering in Medicine and Biology, 14, 307–313, 1995

[Keil09] Keil, Matthias; **Stolka, Philipp**; Wiebel, Marion; Sakas, Georgios; McVeigh, Elliot; Taylor, Russell H; Boctor, Emad: "Ultrasound and CT Registration Quality: Elastography vs. Conventional B-Mode". In: Proceedings of ISBI 2009 – IEEE International Symposium on Biomedical Imaging (Boston, MA/USA), 2009

[Klenzner09a] T. Klenzner, Chiu Chun Ngan, F.B. Knapp, H. Knoop, J. Kromeier, A. Aschendorff, E. Papastathopoulos, J. Raczkowsky, H. Wörn, J. Schipper: "New strategies for high precision surgery of the temporal bone using a robotic approach for cochlear implantation". European Archives of Otorhinolaryngoly (2009) 266:955–960

[Klenzner09b] T Klenzner, FB Knapp, J Schipper, J Raczkowsky, H Woern, LA Kahrs, M Werner, P Hering: "High precision cochleostomy by use of a pulsed CO2 laser – an experimental approach". Cochlear Implants Int. 10(S1), 58–62, 2009

[Konietschke08] Konietschke, Rainer: "Planning of Workplaces with Multiple Kinematically Redundant Robots". Dissertation thesis, Technische Universität München, 2008

[Koulechov06] K. Koulechov, G. Strauss, A. Dietz, M. Strauss, M. Hofer, T.C. Lueth: "FESS control: Realization and evaluation of navigated control for functional endoscopic sinus surgery". Computer Aided Surgery, May 2006; 11(3): 147–159, 2006

[Kroeger10] Kroeger, Torsten: "On-Line Trajectory Generation in Robotic Systems. Basic Concepts for Instantaneous Reactions to Unforeseen (Sensor) Events". Springer Tracts in Advanced Robotics, 2010

[Lea95] Lea, J.T.; Santos-Munné, J.J.; Peshkin, M.A.: "Diagramming Registration Connectivity and Structure". In: Proceedings of Medicine Meets Virtual Reality (MMVR) III, San Diego, USA, 1995

[Lenze06] Carsten Lenze, Ralf Eckert, Andreas Hein: "Evaluation of dynamic performance of the Interactive Milling System MicroAssistant". In: Proceedings of CURAC 2006

[Li04] Ming Li, R.H. Taylor: "Spatial Motion Constraints in Medical Robot Using Virtual Fixtures Generated by Anatomy". In: Proceedings of the 2004 IEEE International Conference on Robotics & Automation New Orleans, LA, April 2004

[Lueth98] Tim C. Lueth, Andreas Hein, Jochen Albrecht, Mustafa Demirtas, Stefan Zachow, Ernst Heissler, Martin Klein, Horst Menneking, Guenter Hommel, Juergen Bier: "A Surgical Robot System for Maxillofacial Surgery". In: Proceedings of the 24th Annual Conference of the IEEE IECON '98

[Lueth02] Lüth, T.; Hein, A.: "Navigated Control – Ein neuer Ansatz für das exakte Fräsen", VDI-Berichte 1679 – Proceedings of Tagung Robotik, 2002

[Maillet04] P Maillet, P Poignet, E Dombre: "Optimisation of the Robot Placement in the Operating Room". In: Proceedings of MRNV (Medical Robotics, Navigation and Visualisation), Remagen/Germany, 2004

[Malthan03] D. Malthan, G. Ehrlich, J. Stallkamp, F. Dammann, E. Schwaderer, M.M. Maassen: "Automated Registration of Partially Defective Surfaces by Local Landmark Identification". In: Computer Aidcd Surgery 8:300–309 (2003)

[Malthan04] D. Malthan et al.: "Improvement of Computer- and Robot-Assisted Surgery at the Lateral Skull Base by Sensory Feedback". In: Proceedings of MRNV (Medical Robotics, Navigation and Visualisation), Remagen/Germany, 2004

[Maurer99] Calvin R Maurer Jr, Ronald P Gaston et al: "AcouStick: A Tracked A-Mode Ultrasonography System for Registration in Image-Guided Surgery". Medical Image Computing and Computer-Assisted Intervention – MICCAI'99, 1999

[Megerian10] Cliff A Megerian, Gail S Murray: "Cochlear Implants, Surgical Technique". http://emedicine.medscape.com/article/857242-overview, accessed Mar 3, 2010

[Marescaux02] Jacques Marescaux; Joel Leroy; Francesco Rubino; Michelle Smith; Michel Vix; Michele Simone; Didier Mutter: "Transcontinental Robot-Assisted Remote Telesurgery: Feasibility and Potential Applications". Annals of Surgery Vol. 235, No. 4, 487–492, 2002

[Mo08] L. Mo, D. DeBusschere, G. McLaughlin, et al.: "Compact ultrasound scanner with simultaneous parallel channel data acquisition capabilities". In: Proceedings of IEEE International Ultrasonics Symposium, pp. 1342-1345, 2008

[NewRiverKinematics] New River Kinematics, “COBRA Robot Deflection Modeling”, http://www.kinematics.com/custom_projects_cobra.htm, accessed May 2006

[Nimsky04] Ch. Nimsky, J. Rachinger, H. Iro, R. Fahlbusch: ”Adaptation of a Hexapod-Based Robotic System for Extended Endoscope-Assisted Transsphenoidal Skull Base Surgery”. Minimally Invasive Neurosurgery 2004; 47: 41–46

[NIST06] National Institute of Standards and Technology, Advanced Technology Program Status Report: ”Integrated Surgical Systems, Inc.: Robodoc for Follow-Up Hip Replacement Surgery”. http://statusreports.atp.nist.gov/reports/94-01-0228.htm

[OROCOS] http://www.orocos.org/, accessed May 2011

[Park01] S. Park, R.D. Howe, D.F. Torchiana: ”Virtual fixtures for robotic cardiac surgery”. In: Proceedings of MICCAI 2001

[Peters03] Peters, Matthias: ”RONAF: Entwicklung einer ergonomischen Benutzerführung für einen Chirurgieroboter (Development of ergonomic user guidance for a surgical robot)”. Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2003

[Plinkert98] Plinkert P. K., Baumann I., Flemming E., Loewenheim H., Buess G. F.: ”The use of a vibrotactile sensor as an artificial sense of touch for tissues of the head and neck”. Minimally Invasive Therapy and Allied Technologies, 1998:7/2:111-115, 1998

[Popovic03] A. Popovic, M. Engelhardt, T. Wu, F. Portheine, K. Schmieder, K. Radermacher: ”CRANIO—computer-assisted planning for navigated and robot-assisted surgery on the skull”. In: Proceedings of the 17th International Congress and Exhibition CARS – Computer Assisted Radiology and Surgery, 2003

[Popovic05] Popovic A., Engelhardt M., Heger S., Radermacher K.: ”Efficient Non-Invasive Registration with A-mode Ultrasound in Skull Surgery: Method and First Clinical Trials”. In: Proceedings of 18th Conference on Computer Assisted Radiology and Surgery (CARS 2005), pp. 821-826, 2005

[Pott05] Pott, P.P.; Scharf, H.-P.; Schwarz, M.L.R.: ”Today’s state of the art in surgical robotics”. Computer Aided Surgery, 10(2), 101 – 132, 2005

[Raczkowsky06] J. Raczkowsky, H. Wörn: ”Hochpräzisionschirurgie mit Robotern”. In: Proceedings of CURAC 2006

[Raetzer05] A. Rätzer-Scheibe, M. Klasing, M. Werner, M. Ivanenko, P. Hering: ”Acoustic monitoring of bone ablation using pulsed CO2 lasers”. Technical Report, University Duesseldorf, Germany. 2005

[Reiley09] C.E. Reiley, G.D. Hager: ”Task versus Subtask Surgical Skill Evaluation of Robotics Minimally Invasive Surgery”. In: Proceedings of MICCAI 2009, London, UK

[Rojas93] Rojas, R.: ”Theorie der neuronalen Netze”. Springer Verlag Berlin Heidelberg, ISBN 3-540-56353-9, 1993

[ROS] http://www.ros.org/wiki/, accessed May 2011

[Rosen01] Rosen, Jacob; Hannaford, Blake; Richards, Christina G.; Sinanan, Mika N.: ”Markov Modeling of Minimally Invasive Surgery Based on Tool/Tissue Interaction and Force/Torque Signatures for Evaluating Surgical Skills”. IEEE Transactions on Biomedical Engineering, Vol. 48, No. 5, May 2001

[SAW] https://www.cisst.org/saw/Main_Page, accessed May 2011

[Seo09] Joonho Seo, Norihiro Koizumi et al.: "Development of computer controlled three dimensional HIFU focus model scanning system". In: Proceedings of 9th International Symposium on Therapeutic Ultrasound, Aix en Provence, France. 2009

[Shamdasani08] V. Shamdasani, U. Bae, S. Sikdar, et al.: "Research interface on a programmable ultrasound scanner". Ultrasonics, vol. 48, pp. 159-168, 2008

[Shoemaker85] Shoemaker, K.: "Animating rotation with quaternion curves". SIGGRAPH Computational Graphics, 1985.

[Slicer3D] http://www.slicer.org, accessed May 2011

[Speich04] Speich, J. E. ; Rosen, J.: "Medical Robotics". Encyclopedia of Biomaterials and Biomedical Engineering, 2004.

[SpringerRobotics] Springer "Handbook of Robotics". Siciliano, Khatib (ed.), Springer 2008

[Stäubli] "Stäubli RX130 series industrial robots", http://www.staubli.com/robotics/, accessed May 2011

[STL] Marshall Burns: "The StL Format", Section 6.5 of Automated Fabrication, http://www.ennex.com/~fabbers/StL.asp, accessed May 14, 2010

[Stolka01] **Stolka, Philipp**: "Robot-assisted otolaryngology surgery: Voruntersuchungen zum Sensoreinsatz (First Experiments for Sensor Application)". Projektarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2001

[Stolka02] **Stolka, Philipp**: "Robotergestützte Navigation zum Fräsen an der lateralen Schädelbasis: Auswertung von Kraftsensordaten". Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2002

[Stolka03] **Stolka, Philipp**; Henrich, Dominik: "A Hybrid Force-Following Controller for Multi-Scale Motions". In: Proceedings of SYROCO 2003

[Stolka05] **Stolka Ph.**; Henrich D.: "Building Local Maps in Surgical Robotics". In: Proceedings of IROS 2005

[Stolka06a] **Stolka, Philipp**; Tretbar, Steffen; Waringo, Michel; Federspil, Philipp; Henrich, Dominik: "Robot-Assisted 3D Ultrasound Scanning with IR Navigation Support". In: Proceedings of CURAC 2006

[Stolka06b] **Stolka, Philipp**; Henrich, Dominik: "Improving Navigation Precision of Milling Operations in Surgical Robotics". In: Proceedings of IROS 2006

[Stolka07a] **Stolka, Philipp**; Henrich, Dominik: "Robot-Based 3D Ultrasound Scanning and Registration with Infrared Navigation Support". In: Proceedings of ICRA 2007

[Stolka07b] **Stolka, Philipp**; Henrich, Dominik: "Using Maps from Local Sensors with Volume-Removing Tools". In: Proceedings of IROS 2007

[Stolka08] **Stolka, Ph. J.**; Henrich, D.; Tretbar, S. H.; Federspil, Ph. A.: "First 3D Ultrasound Scanning, Planning, and Execution of CT-free Milling Interventions with a Surgical Robot". In: Proceedings of IEEE EMBS Conference 2008, Vancouver, British Columbia, Canada

[Stolka10a] **Ph.J. Stolka**; M. Keil; G. Sakas; E.R. McVeigh; R.H. Taylor; E.M. Boctor: "A 3D-elastography-guided system for laparoscopic partial nephrectomies". In: Proceedings of SPIE Medical Imaging 2010 (San Diego, CA/USA), 2010

[Stolka10b] **Ph.J. Stolka**, H-J. Kang, M. Choti, E.M. Boctor: "Multi-DoF Probe Trajectory Reconstruction with Local Sensors for 2D-to-3D Ultrasound". In: Proceedings of ISBI 2010 - IEEE International Symposium on Biomedical Imaging (Rotterdam/The Netherlands), 2010

[Stolka10c] **Ph.J. Stolka**, H-J. Kang, E.M. Boctor: "The MUSiiC Toolkit: Modular Real-Time Toolkit for Advanced Ultrasound Research". In: International Workshop on Systems and Architectures for Computer Assisted Interventions 2010, MICCAI 2010, Beijing, China

[Strauss05] Strauss M., Olze H., Lüth T.C.: "Accuracy analysis of cavities for cochlea implants milled conventionally and manually with navigated control". In: Proceedings of CARS 2005: Computer Assisted Radiology and Surgery, vol. 1281, 553 – 558, May 2005

[Sugita08] Sugita, N.; Osa, T.; Nakajima, Y.; Mitsuishi, M.: "Deformation Analysis and Active Compensation of Surgical Milling Robot Based on System Error Evaluation". In: Proceedings of IEEE International Conference on Robotics and Automation 2008, Pasadena, CA, USA, May 19-23, 2008

[Sugita08b] Naohiko Sugita, Takayuki Osa, Yoshikazu Nakajima and Mamoru Mitsuishi: "Optimization of Toolpath Generation in Medical CAM for a Machine Tool for Orthopedic Surgery". In: Proceedings of 41st CIRP Conference on Manufacturing Systems May 26–28, 2008, Tokyo, Japan

[Sugita09] Naohiko Sugita, Taiga Nakano, Yoshikazu Nakajima, Kazuo Fujiwara, Nobuhiro Abe, Toshifumi Ozaki, Masahiko Suzuki, Mamoru Mitsuishi: "Dynamic controlled milling process for bone machining". Journal of Materials Processing Technology 209 (2009) 5777–5784, 2009

[Taylor94] R.H. Taylor et al: "An image-directed robotic system for precise orthopaedic surgery". IEEE Transactions on Robotics and Automation 10(3):261-275, June 1994.

[Taylor99] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, et al.: "A Steady-Hand Robotic System for Microsurgical Augmentation". International Journal of Robotics Research, 18(12), 1999

[Taylor03] Taylor, R. H.; Stoianovici, D.: "Medical Robotics in Computer-Integrated Surgery". IEEE Transactions on Robotics and Automation, Vol. 19, No. 5, Oct. 2003.

[Tokuda09] J. Tokuda, G.S. Fischer et al.: "OpenIGTLink: an open network protocol for image-guided therapy environment". Interational Journal of Medical Robotics and Computer-Assisted Surgery 2009; 5: 423434

[Trantakis09] Trantakis, C.; Dengl, M.; Grunert, R.; Korb, W.; Strauss, G.; Lueth, T.; Meixensberger, J.: "Evaluation of a navigated and power controlled milling system 'navigated control spine' for spinal surgery". International Journal of Computer Assisted Radiology and Surgery, Volume 4, Number 5 / September, 2009

[Tretbar03] Tretbar, S.H.; Schmitt, D.; Günther, C.; Hoß, M.; Weber P. K.; Lemor, R. M.: "Study of Soundfield Characteristics under Coded Excitation", 5th World Congress on Ultrasonics, Paris, France, 2003

[Tretbar04a] Tretbar, S.H.; Federspil, Ph.A.; Günther, C.; Plinkert, P.K.: "Ultraschall-Dickenmessung der Schädelkalotte zur Registrierung bei navigierten Eingriffen an der lateralen Schädelbasis", Biomedizinische Technik 2004, 49 Ergänzungsband 2 (858-859), 2004

[Tretbar04b] S. H. Tretbar, Ph.A. Federspil, P.K. Plinkert: "Improved ultrasound based navigation for robotic drilling at the lateral skull base". In: Proceedings of CARS 2004

[Tretbar09] Tretbar, S.H. Plinkert, P.K. Federspil, P.A.: "Accuracy of Ultrasound Measurements for Skull Bone Thickness Using Coded Signals". IEEE Transactions on Biomedical Engineering, March 2009

[Troccaz96] J. Troccaz, Y. Delnondedieu: "Semi-active guiding systems in surgery. A two-dof prototype of the passive arm with dynamic constraints (PADyC)". Mechatronics, 6(4):399—421, June 1996

[Umbach00] Umbach, D.; Jones, K.N.: "A Few Methods for Fitting Circles to Data". IEEE Transactions on Instrumentation and Measurement, 2000

[Waringo01] Waringo, Michel: „Robotergestützte HNO-Chirurgie (ROTOS – Robotics in Otolaryngology Surgery): Fräsalgorithmus". Projektarbeit, Universität Kaiserslautern, Fachbereich Informatik, AG Robotik und Eingebettete Systeme (RESY). 2001

[Waringo03a] Waringo, Michel: "Robotergestützte Navigation zum Fräsen an der lateralen Schädelbasis: Lageoptimierung und schichtweise Bahnplanung auf Basis globaler Sensordaten". Diplomarbeit, Fachbereich Informatik, Universität Kaiserslautern, 2003

[Waringo03b] Waringo, Michel; **Stolka, Philipp**; Henrich, Dominik: "First System for Interactive Position Planning of Implant Components". In: Proceedings of CURAC 2003 – Computer- und Roboter-Assistierte Chirurgie, Nürnberg/Germany. 2003

[Waringo03c] Waringo, Michel; **Stolka, Philipp**; Henrich, Dominik: "3-Dimensional Layered Path Planning for Anytime Milling Applications". In: Proceedings of CURAC 2003 – Computer- und Roboter-Assistierte Chirurgie, Nürnberg/Germany. 2003

[Waringo06] Waringo, Michel; Henrich, Dominik: "Efficient smoothing of piecewise linear paths with minimal deviation". In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2006, pp. 3867-3872

[Widmann07] G Widmann: "Image-guided surgery and medical robotics in the cranial area". Biomedical Imaging and Intervention Journal, 2007

[Wilson06] T. Wilson, J. Zagzebski, T. Varghese, Q. Chen, M. Rao: "TheUltrasonix 500RP: A Commercial Ultrasound Research Interface". IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 53, no. 10, October 2006

[Winter02] S. Winter, B. Brendel, A. Rick, M. Stockheim, K. Schmieder, H. Ermert: "Registrierung 3-dimensionaler CT- und Ultraschalldaten anhand von Knochenstrukturen". In: Proceedings of Computer- und Roboter-assistierte Radiologie und Chirurgie (CURAC), Leipzig/Germany, 2002

[Woessner04] S. Woessner, J. Huen, D. Malthan: ”Differentiating tissue by fluorescence spectroscopy”. In: Proceedings of SPIE Vol. 5261

[Xia08] Tian Xia, Clint Baird et al.: ”An integrated system for planning, navigation and robotic assistance for skull base surgery”. International Journal of Medical Robotics and Computer Assisted Surgery 2008; 4: 321–330, 2008

[Yaniv06] Ziv Yaniv, Kevin Cleary: ”Image-Guided Procedures: A Review”. Technical Report CAIMR TR-2006-3, Georgetown University, 2006

Appendix A

System Description

In the next sections, the overall system design will be described from a technical (implementation) point of view (Section A.1), then followed by hardware (Section A.2) documentations.

A.1 System Design

The presented RONAF system follows a modular, distributed software architecture approach (Figure A.1). With the hardware (robot, robot controller, PCs, ultrasound hardware, other sensors etc.) naturally consisting of discrete components, the single software components closely model the corresponding hardware. Wherever possible, functional decomposition dictates that system logic is executed in separate processes or tasks to simplify the overall control flow. Furthermore, this approach allows the definition of clear and relatively simple component interfaces.

The hardware system setup consists of at least the following components:

- a control unit (on a PC) running the user interface,
- a robot controller with the connected robot (and attached peripherals like miller tool, ultrasound probe, and localizer markers), manual control pendant, and the force/torque sensor,
- an ultrasound unit (on a PC) running the ultrasound interface software with the connected ultrasound probe,
- a localizer (e.g. an optical tracking system),
- and possible further physical sensors (e.g. a room microphone).

Furthermore, the following functional units need to be present:

- logical sensors (e.g. an audio classifier, a pose-providing interface for the robot, and/or an F/T classifier etc.),
- and an NFS volume server providing file exchange services.

The non-robot-control-related software modules on the PC(s) can in principle be arbitrarily distributed on a network, e.g. on one, two, or more physical or virtual machines (for the presented system, the setup matched the graph in Figure A.1). NFS services were provided by a central, remote NFS server. The PC-based components are executables that are independent of each other (implemented in C/C++ with Qt 3.x integration for graphical user interface and

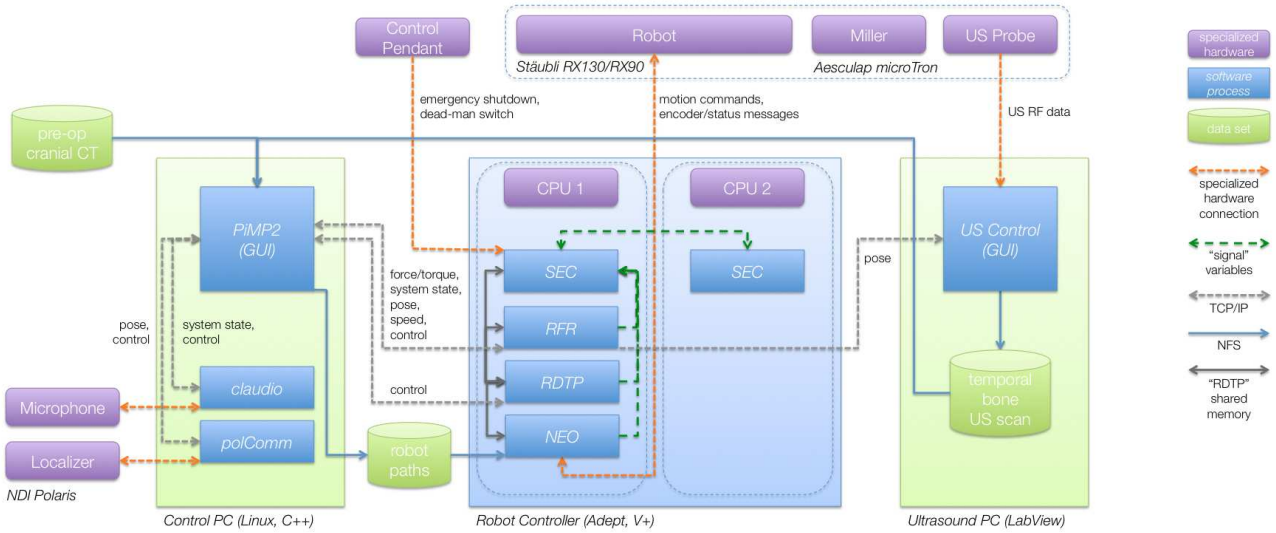


Figure A.1: RONAF software architecture graph. The system is distributed on three separate machines (left: main control PC; center: robot controller; right: ultrasound control PC), with communications performed via TCP/IP and NFS over Ethernet. The robot, miller, and ultrasound probe are physically linked.

networking functionality), while the robot-controller components are started by spawning them from a central control program (in the robot controller's standard language V+).

The inter-component communications are realized differently depending on the type of participating components:

- For the components implemented as separate processes on the control PC (the *PiMP2* control graphical user interface (GUI), the *claudio* audio classifier, and the *polComm* localizer interface), as well as for real-time communication between software components on different machines, information is exchanged via TCP/IP using proprietary, text-based protocols. For commands, status messages, and other non-sensor-data packets, this is implemented via the proprietary RDTP (*RONAF Data Transfer Protocol*) using messages of the form

  ```
  sender->receiver!variable:=value1;value2;...
  ```

  where *sender* and *receiver* are module names on the same or different machines. Those modules have to implement the RDTP interface by providing the *receiveRDTPMessage(...)* method, which gets called by a central *RDTP_ServerDaemon* instance that coordinates buffered, asynchronous network communication. For "local" modules (within the same process/task) messages are not routed via TCP/IP but rather delivered directly.

- Between components implemented as V+ tasks on the robot controller (the safety module *SEC*, the F/T sensor interface *RFR*, the network interface *RDTP*, and the robot control task *NEO*), two different communication channels exist: For watchdog, emergency signaling, and inter-CPU communication, so-called V+ "signals" (software and hardware binary flags) are used, while for general intra-CPU control the communication messages are implemented as direct procedure invocations and shared memory accesses across component boundaries.

- For the exchange of image data and path descriptions between machines, files are created on and read from NFS-mounted volumes by the participating modules, with the files either manually selected by the user (as e.g. for CT DICOM data) or identified by the

Figure A.2: Manual control pendant connected to the robot controller. The two redundant STOP buttons (red) are hardwired into the controller emergency shutdown circuit, while the two redundant ROBOT buttons (blue) and the MENU buttons can be queried by robot controller programs in software.

file names transmitted in corresponding RDTP transfer initiation messages via TCP/IP as described above.

The safety subsystem on the robot controller serves to detect safety-relevant user inputs, PiMP2 messages, error signals, and undefined or undesired system states, and to react accordingly. It consists of one instance of the SEC V+ module, run as one task each on both the robot controller's CPUs (for CPU 2, this is the only executing task). On system startup, they are executed independently on their respective CPUs, but synchronize early on via a two-way handshake with each other before any other tasks are started.

- The safety modules SEC on the robot controller perform error checking (in a watchdog function) by checking first the execution status (running/not running) of the corresponding SEC on the other CPU, then the execution status of the main robot control task NEO and the force control task RFR. If any of these are not running (for NEO, this can be triggered by a force-protect overload, i.e. excessive forces causing a robot program termination and robot servo shutdown), a fatal error is triggered.
- If the networking task RDTP is not running, a fatal error is triggered; if either the RDTP or the PiMP2 do not respond (with continuous alive messages), a warning is triggered.
- The manual control pendant's (Figure A.2) red buttons are hardwired (together with an additional emergency button on the robot controller, and a force protect overload line from the F/T sensor on the robot arm) into the emergency power cutoff circuit of the robot controller. As soon as the robot "high power" is cut, the SEC module triggers a fatal error.
- If a fatal error was triggered, the robot is shut down immediately, a message is printed to the console and an RDTP message is sent to the PiMP2. Then, the other running tasks are asked to terminate cleanly, and aborted after another five seconds.
- The manual control pendant's blue buttons serve as a dead-man switch – when pressed, the robot monitor speed it set to its default value, when depressed, it is set to zero, thus allowing immediately interruption/continuation of the current robot motion.

A.2 Hardware Documentation

A.2.1 Control PC

The main process control software (comprised of mainly the configuration, planning, and monitoring GUI) is running on a standard PC (AMD Athlon XP 2600+ with 2GB RAM under OpenSuSE Linux) with 100Mbps Ethernet to a switch connecting to the robot controller and the US sensor control computer.

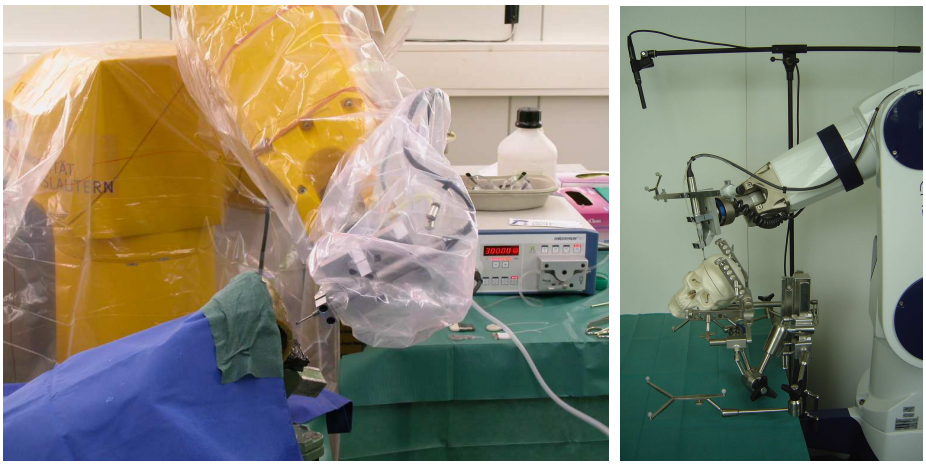


Figure A.3: System setup with the initial Stäubli RX130 (left; in a thermal imaging setup while milling on a human temporal bone preparation), later with a Stäubli RX90/CASPAR (right; with room microphone and extended tool holder with US probe and Polaris rigid body).

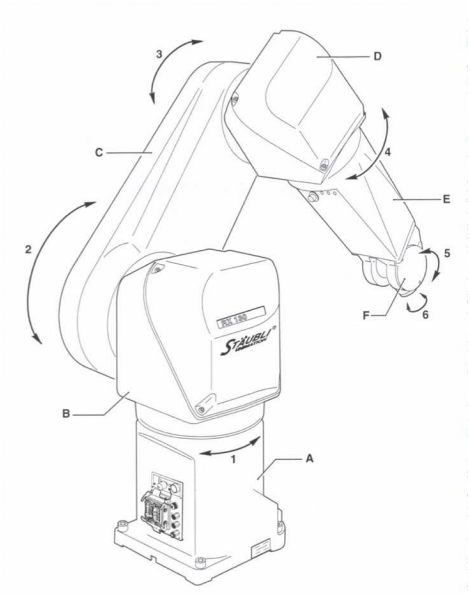


Figure A.4: Schematic sketch of the Stäubli RX130, with its six axes enumerated [Stäubli].

A.2.2 Robot

The presented methods were tested on the RONAF system („Robot-based Navigation for Milling at the lateral Skull Base“) for automated milling of cavities in the skull bone for subdermal implantation of hearing aids. The robot is an industrial model (Stäubli RX90 [Stäubli]) retrofitted with speed and safety enhancements, and that was in medical use for hip and knee endoprosthesis milling applications (the former CASPAR system's robotic component, by the former company Orto-Maquet). An initial version of the presented system was based on a slightly larger Stäubli RX130 for industrial use (Figures A.3, A.4). Both robots are serial six-axis kinematics and are controlled by Adept real-time robot controllers (40MHz 68040-based Adept CS7 with 32MB RAM) using the interpreted robot programming language V+ 12.3. Their nominal repeatability is $0.02mm$ (RX90) and $0.03mm$ (RX130), although they exhibit much (around an order of magnitude) larger motion inaccuracies in reality.

The tool is a surgical miller (electrically driven miller Aesculap microtron EC/GD622, up to $30,000rpm$) mounted perpendicularly to the robot tool flange to minimize deformation.

A.2.3 Sensors

Commercial (off-the-shelf) sensors include – amongst others – a 6D force/torque (F/T) sensor (JR3 90M31A with strain gauge bridges, max. sensing range $63N/5Nm$, resolution 1:4000; later a JR3 50M31A [JR3]), a room microphone, and an NDI Polaris infrared-optical tracking system (OTS; measured repeatability accuracy $0.05mm$ root-mean-square (RMS) after warm-up phase, specified absolute accuracy $0.35mm$ RMS, silo-shaped work volume of approx. $(1000mm)^3$, data rate $20...60Hz$).

System-specific sensors include a Transmit-Receive Module II ultrasound probe (US; by Fraunhofer IBMT, St. Ingbert/Germany). The US system control computer is indirectly connected to the robot controller for position acquisition via a local TCP/IP network. Using two ultrasound probes (center frequency $2.25MHz$, diameter $\frac{1}{4}in.$ and $1MHz$, diameter $\frac{1}{2}in.$) with flexible delay lines attached to the transducer, filtered and unfiltered radio frequency (RF) signals are available for signal processing.

A.3 Notation and Abbreviations

${}^{A}P$ – path or point P in coordinate system A

${}^{A}M_{B}$ – transformation matrix giving the position of point or coordinate system B in coordinates of A

P – robot path in milling (local, implant) volume coordinates (i.e. after complete-coverage path planning)

${}^{T}P$ – robot path in reference position T coordinates (i.e. relative to start position T, after registration in the execution site)

${}^{rob}P$ – robot path in robot (global, world) coordinates (needed e.g. for reachability checking)

S – tool shape (with S_p the passive and $S_a(v)$ the active tool shape)

$\mathbf{v}^T$ – transpose of the vector $\mathbf{v}$

F/T – Force/Torque sensing

IR – infrared

MLS – *M*ap from *L*ocal *S*ensors

indepMLS – *indep*endently-built *M*ap from *L*ocal *S*ensors

conMLS – *con*currently-built *M*ap from *L*ocal *S*ensors

OTS – optical tracking system

RONAF – Robot-based Navigation for Milling at the Lateral Skull Base; project name for the described system

US – ultrasound (mostly A-mode)

List of Figures

List of Tables

List of Algorithms

Postfix

- *When I got back to Boston I went to the library and discovered a book by Kimura on the subject, and much to my disappointment, all of our "discoveries" were covered in the first few pages. When I called back and told Richard [Feynman] what I had found, he was elated.* "Hey, we got it right!" *he said.* "Not bad for amateurs." *In retrospect I realize that in almost everything that we worked on together, we were both amateurs. In digital physics, neural networks, even parallel computing, we never really knew what we were doing. But the things that we studied were so new that no one else knew exactly what they were doing either. It was amateurs who made the progress.* – W. Daniel Hillis

About the Author

Philipp J. Stolka received the degree "*Diplom-Informatiker*" at the University of Kaiserslautern, Germany (now Technical University of Kaiserslautern) in October 2002 (grade 1.3/"*sehr gut*"). His diploma thesis was titled "*Robot-Based Navigation for Milling at the Lateral Skull Base: Evaluation of Force Sensor Data*" (grade 1.0) and supervised by Prof. Dr. Dominik Henrich (University of Kaiserslautern) and Prof. Dr. med. Peter K. Plinkert (Ear-Nose-Throat Department at the University Clinic of the University of Saarland, Homburg; now at the UniversitaetsKlinikum Heidelberg).

Initial work done in the author's earlier project thesis with the title "*Robot-assisted Otolaryngology Surgery: First Experiments for Sensor Application*" (2000) helped to secure a research grant from the Deutsche Forschungsgemeinschaft (DFG) within the Priority Programme SPP 1124 (*"Medical Navigation and Robotics"*). This program has funded the presented research at the University of Kaiserslautern (later at the University of Bayreuth) and the University of the Saarland (later at the University of Heidelberg) with three successive DFG grants during the period of years 2002–2008.

The results presented in this dissertation thesis have been published at various national and international conferences, including robotics, control, medical, and bio-medical engineering conferences throughout Europe, Northern America, and Asia, as well as at several technical workshops and invited talks in academia and industry.

The author's current affiliation (2008–) is with the Laboratory for Computational Sensing and Robotics (LCSR) under Prof. Russell H. Taylor and the Medical Ultrasound Imaging and Interventions Collaboration (MUSiiC) under Dr. Emad M. Boctor at The Johns Hopkins University, Baltimore, Maryland/USA. His research interests include ultrasound-based elasticity imaging (EI), laparoscopic navigation using 3D-EI and CT [Stolka10a], and integration of local sensors into handheld US probes to replace global tracking [Stolka10b]. He is also leading the development of a distributed real-time software toolkit for integrated ultrasound research applications [Stolka10c]. Last not least, he is co-founder and serves as Chief Technical Officer for the medical device development startup company Clear Guide Medical, LLC and is co-inventor on several patents.

40 Jahre Technische Universität Kaiserslautern · Universität Bayreuth · Johns Hopkins University · Johns Hopkins Medicine