Implementation of a 3D TRUS acquisition system for robotized focal biopsies of the prostate

An approach using 3D integrated US transducers

João Daniel Marques Ramalhinho

Thesis to obtain the Master of Science Degree in Biomedical Engineering

Supervisors: Professor Jorge Manuel Mateus Martins
Doctor Jorge Manuel Trindade Rebola

Examination Committee

Chairperson: Professor Ana Teresa Correia de Freitas
Supervisor: Professor Jorge Manuel Mateus Martins
Members of the Committee: Professor João Miguel Raposo Sanches
Doctor Arlindo Jorge de Abreu Fonseca

December 2014
Acknowledgments

I would like to express here my most sincere thanks to everyone that helped and supported me during the elaboration of this work.

Firstly, I would like to thank my advisor professor Jorge Martins for giving priceless feedback and guidance that were crucial in the development of this thesis.

I also thank the staff of the Medical Robotics Lab, Cláudio Brito, Sérgio Afonso, Tiago Salgueiro and Pedro Teodoro not only for the brainstorming discussions about work, but also for the companionship feeling demonstrated.

To my dearest masters colleagues Catarina Farinha, David Fernandes, Hugo Paixão, Ricardo Filipe and Tiago Chaves I would like to say how grateful I am for having you by my side during all this academic journey in IST.

To my long date friends from high school, I express my thanks for being always there for me in all kinds of situations. Without their friendship this work would not exist.

Finally and above all, I acknowledge my family for the unconditional support, love and care not only during my academic journey but during all my life.
Abstract

Prostate cancer is the most common neoplasm among men in developed countries and lacks well-established preventable risk factors. In case of suspicion, the standard technique used to diagnose this pathology is the TRUS-guided biopsy. In the last two decades several improvements have been introduced in this technique in order to improve its results. Taking into account the currently most researched adaptations, in this work a hardware setup prepared for 3D TRUS acquisition is developed, which final objective is to be integrated in a robotized system guided by MRI/3D TRUS data fusion. The setup comprises an ultrasound acquisition tool that consists of a 3D integrated transducer and its manipulation hardware, a lightweight robot that holds and moves the transducer and a host computer that commands the acquisition remotely. Due to the lack of information on 3D integrated transducers, a complete analysis to the manipulation software used, Propello SDK is performed. Once the setup is operational, the host acquires kinematic data from the lightweight robot and ultrasound images from the transducer synchronously at a frequency of 14.476 fps. Evaluation trials with a prostate phantom are performed to test the performance of the acquisition system in correlating robot movements to geometrical transformations in 3D TRUS image data sets. The overall results are coherent, making this setup a solid basis for further development of the biopsy robotized system.

Keywords

3D TRUS, Prostate biopsies, Manipulation of 3D Integrated Transducers, Manipulation of Lightweight Robots, Synchronous Data Acquisition.
Resumo

O cancro da próstata é o mais comum entre homens nos países desenvolvidos e não possui factores de risco evitáveis. Em caso de suspeição, a técnica utilizada para diagnosticar esta patologia é a biópsia guiada por TRUS. Nas últimas duas décadas várias melhorias tem vindo a ser introduzidas nesta técnica de forma a melhorar a sua eficiência. Tendo em conta as adaptações mais recentes, neste trabalho é desenvolvida uma montagem de hardware preparada para aquisição de 3D TRUS cujo objectivo é ser integrada num sistema robotizado guiado por fusão MRI/3D TRUS para biópsias à próstata. Esta montagem contém uma ferramenta de aquisição de ultrassons composta por uma sonda 3D e o respectivo equipamento de manipulação, um robot de baixo peso que segura e move a sonda e um computador host que coordena o processo de aquisição remotamente. Devido à falta de informação acerca de sondas 3D, é realizada uma análise extensa ao software de manipulação utilizado, o Propello SDK. Após a montagem estar operacional, o host adquire dados cinemáticos do robot e imagens da sonda 3D sincronizadamente a uma frequência de 14.476 fps. Testes com um phantom de próstata são realizados para analisar o desempenho do sistema de aquisição na tarefa de correlacionar movimentos do robot com transformações geométricas ocorridas nos volumes 3D TRUS adquiridos. Os resultados são coerentes e levam a concluir que a montagem constitui uma base sólida para o posterior desenvolvimento do sistema de biópsias robotizado.

Palavras Chave

Ultrassonografia Transrectal 3D, Biópsias à próstata, Manipulação de Sondas 3D, Manipulação de Robots de baixo peso, Aquisição sincronizada de dados.
# Contents

1 Introduction ......................................................... 1
   1.1 Prostate Cancer-The Pathology ................................. 2
   1.2 Diagnosis .................................................... 4
   1.3 TRUS-guided Prostate Biopsy Methodologies ................. 5
      1.3.1 Systematic Prostate Biopsy ............................... 5
      1.3.2 MRI Based Approaches .................................. 7
         1.3.2.A Multiparametric Magnetic Resonance Imaging .... 8
         1.3.2.B In-bore MRI Guided Biopsy ......................... 8
         1.3.2.C Cognitive Registration .............................. 9
         1.3.2.D MRI-TRUS Data Fusion ............................... 9
   1.4 3D TRUS Imaging ............................................... 11
      1.4.1 Free-hand Approaches .................................. 12
      1.4.2 Mechanical Scanning ................................... 12
   1.5 Role of Robotics .............................................. 14
   1.6 Proposed Implementation .................................... 15
   1.7 Thesis Outline ............................................... 17

2 3D Integrated Transducer Control ................................. 19
   2.1 The SonixTABLET™ ............................................ 20
   2.2 The Propello Software ........................................ 21
   2.3 Remote Wobbling Control using Propello .................... 22
      2.3.1 The C++ Environment of Propello ....................... 22
      2.3.2 Automatic Mode Imaging ................................ 23
      2.3.3 Manual Mode Imaging .................................. 25
   2.4 Propello based software Implementations .................. 27
      2.4.1 MATLAB-Propello Communication ....................... 27
      2.4.2 Manual Mode Optimized routine ........................ 29
      2.4.3 External Triggering in Propello ....................... 31
      2.4.4 External Triggering Optimized routine ................. 34
      2.4.5 Triggered versus Manual ................................ 36
# Robotic Coupling

- **3.1 The KUKA Lightweight Robot** ........................................ 40
- **3.2 The MATLAB-FRI Communication** .................................. 41
  - 3.2.1 Triggered Data Requesting ..................................... 42
  - 3.2.2 TCP/IP Data Retrieval ........................................ 43
- **3.3 Mechanical Coupling** ............................................. 43

# Registration Software Implications

- **4.1 Registration Algorithm Overview** ................................. 48
- **4.2 Semi-Automatic Segmentation** .................................... 49
  - 4.2.1 ASM components ................................................ 49
  - 4.2.2 The Segmentation procedure ................................... 50
- **4.3 The shape parametrization problem** ................................. 51
- **4.4 Probe orientation adjustment Strategy** ............................. 53
- **4.5 Robotic probe movement Implementation** ........................ 56

# Final Application and System Evaluation

- **5.1 Final Application GUI** ........................................... 60
- **5.2 Evaluation Trials** .................................................. 63
  - 5.2.1 Trial results discussion ....................................... 68

# Conclusions and Future Work

- **6.1 Conclusions** ......................................................... 70
- **6.2 Future Work** .......................................................... 71

# Bibliography

- **Bibliography** ............................................................. 73
List of Figures

1.1 Illustration of the prostate anatomy. ......................................................... 2
1.2 Age-adjusted prostate cancer incidence and mortality rates for Black and White Amer-
  icans. ........................................................................................................ 3
1.3 TRUS probe setup. ..................................................................................... 5
1.4 Sextant division of the prostate. ................................................................. 6
1.5 Standard 12-core prostate biopsy template. ............................................. 7
1.6 Elastic and Rigid Registration of MRI to TRUS. ...................................... 10
1.7 Mechanical scanning methods and their respective 2D image set geometry. 13
1.8 Wobbling-based scan in terms of image geometry. .................................. 14
1.9 4DEC9-10/5 3D Integrated transducer and wobbler scheme. .................. 15
1.10 Simplistic description over the control scheme of the intended robotized system. 16

2.1 The SonixTABLET hardware. .................................................................... 20
2.2 Propello software application interface. .................................................... 21
2.3 Mechanical wobbling routine during automatic mode imaging of Propello. 23
2.4 Mechanical range description scheme of the 4DEC9-5/10 wobbler. ........... 26
2.5 TCP/IP connection graphical windows implemented in Propello. ............... 28
2.6 Optimized volume covering with Propello Manual mode imaging. ............ 30
2.7 SonixTABLET set of connectors. ............................................................... 31
2.8 NI USB-6008 acquisition board insertion in acquisition setup. ................ 32
2.9 NI USB-6008 ideal output trigger. ........................................................... 33
2.10 Imaging pipeline in Propello triggered acquisition. .................................... 34
2.11 Optimized volume covering with external triggering. .............................. 35

3.1 The KUKA Lightweight Robot system. ..................................................... 40
3.2 KUKA LWR control architecture. ............................................................ 41
3.3 Second NI USB-6008 acquisition board insertion in acquisition setup. .... 42
3.4 Assembly of transducer mechanical coupler with CAD models. ............... 44
3.5 Mechanical coupling between KUKA LWR and 4DEC9-5/10 transducer. 45

4.1 Workflow of the MRI/3D TRUS data fusion algorithm. ......................... 48
4.2 SSM axial shape parametrization. ............................................................. 51
4.3 Graphical representation of the image acquisition geometry of the 4DEC9-5/10 transducer. .................................................. 52
4.4 Graphical representation of oblique slice prostate sampling with transducer rotation around y<sub>p</sub> axis. .................................................. 54
4.5 Graphical representation of oblique slice prostate sampling with transducer rotation around z<sub>p</sub> axis. .................................................. 54
4.6 Graphical representation of oblique slice prostate sampling with transducer rotation around x<sub>p</sub> axis. .................................................. 55
4.7 List of keyboard controls and respective output rotations in the probe reference tool point. 57
5.1 Final Application GUI. .............................................................. 61
5.2 3D TRUS acquisition system flowchart of volume covering. ................................................................. 62
5.3 Experimental setup of the 3D TRUS acquisition system. ................................................................. 64
5.4 Slice matching between reference images and set of images acquired in first experimental trial. ................................................................. 65
5.5 Slice matching between set of reference images and set of images acquired in second experimental trial. ................................................................. 66
5.6 Slice matching between reference images and set of images acquired in third experimental trial. ................................................................. 67
5.7 Slice matching between reference images and set of images acquired in fourth experimental trial. ................................................................. 67
List of Tables

2.1 Most relevant basic commands of Propello ........................................... 22
2.2 List of automatic mode parameter allowed values .................................... 24
2.3 List of portaStepMotor() output values .................................................. 26
2.4 List of strings sent from MATLAB to Propello ........................................ 29
2.5 Time expense of triggered and manual routines during coverage of a volume with 51 images ................................................................. 37
3.1 Obtained calibration values for the probe plus coupler tool ...................... 45
5.1 Probe orientation adjustment data for each experimental trial ................... 64
Abbreviations

**PCa**  Prostate Cancer

**TURP**  Transurethral Resection of the Prostate

**BPH**  Benign Prostatic Hyperplasia

**PSA**  Prostate Specific Antigen

**DRE**  Digital Rectal Examination

**TRUS**  Transrectal Ultrasound

**fPSA**  Free-Total PSA Ratio

**EAU**  European Association of Urology

**NCCN**  National Comprehensive Cancer Network

**MRI**  Magnetic Resonance Imaging

**MP-MRI**  Multiparametric Magnetic Resonance Imaging

**T2W**  T2-Weighted

**DWI**  Diffusion Weighted Imaging

**DCE**  Dynamic Contrast Enhanced

**FEM**  Finite Element Method

**DOF**  degrees of freedom

**TPS**  Thin Plate Splines

**SNR**  Signal-to-Noise Ratio

**SSM**  Statistical Shape Model

**SDK**  Software Development Kits

**GUI**  Graphical User Interface

**TCP/IP**  Transmission Control Protocol/Internet Protocol
FOV  Field of View

KUKA LWR  KUKA Lightweight Robot

UDP  User Datagram Protocol

KRC  KUKA Robot Controller

KRL  KUKA Robot Language

KRP  KUKA Robot Panel

FRI  Fast Research Interface

RBF  Radial Basis Functions

ASM  Active Shape Models

ICP  Iterative Closest Point

PCA  Principal Component Analysis
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Prostate Cancer-The Pathology</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Diagnosis</td>
<td>4</td>
</tr>
<tr>
<td>1.3 TRUS-guided Prostate Biopsy Methodologies</td>
<td>5</td>
</tr>
<tr>
<td>1.4 3D TRUS Imaging</td>
<td>11</td>
</tr>
<tr>
<td>1.5 Role of Robotics</td>
<td>14</td>
</tr>
<tr>
<td>1.6 Proposed Implementation</td>
<td>15</td>
</tr>
<tr>
<td>1.7 Thesis Outline</td>
<td>17</td>
</tr>
</tbody>
</table>
The objective of the present work is to implement a hardware setup capable of 3D TRUS acquisition to be used subsequently in robotized focal biopsies of the prostate. This setup comprises a 3D integrated transducer, a robotic system and specialized manipulation hardware and software tools. Once operational, the data acquired by the system allows for implementation of targeted sampling of the prostate using a real time registration algorithm.

In this first chapter is supplied information on the background of TRUS guided prostate biopsy. In sections 1.1 and 1.2 data is presented about prostate cancer statistics and its diagnosis methods which include the TRUS-guided prostate biopsy. Then in section 1.3 the methodologies of this diagnosis tool are widely described. In section 1.4 the methods of 3D TRUS acquisition are described and their role in the biopsy procedure. In the next section, 1.5 the role of robotics is also pointed out. The overall functioning and purpose of the hardware setup developed in this work is presented in section 1.6. Finally, section 1.7 presents an outline of the thesis structure.

### 1.1 Prostate Cancer-The Pathology

Prostate Cancer (PCA) is the pathology where cancer, the uncontrolled proliferation of abnormal cells is located in the prostate [1]. The prostate is a golfball-sized gland located inferior to the urinary bladder that surrounds the prostatic urethra. It is an exocrine gland whose secretion is a fluid that grants mobility and viability to the sperm and represents about 25% of the total volume of the semen [2]. In terms of structure, this organ is divided according to a concept of zonal anatomy and each zone has a different susceptibility to PCA. There are four of them: the central zone, the transition zone, the peripheral zone and the anterior fibromuscular stroma [3].

![Figure 1.1: Illustration of the prostate anatomy. In the left, the location of the prostate, image from www.pcf.org. In the right, the prostate zonal anatomy with a closer view, image from www.wholelifeprostate.com.](image)

The peripheral zone comprises around 65% of the prostate mass and is the most common origin of PCAs: 70%-80% of cases against 20% in the transition zone and 10% in the central zone [4]. The fibromuscular stroma is not considered in this statistic since it is a smooth muscle layer without presence of prostatic glandular tissue.
As of 2008, in developed countries PCa was the most commonly diagnosed cancer and showed the third highest value of cancer deaths. In opposition to lung cancer that has the highest incidence and death rate, there have not been established preventable risk factors for PCa [5]. The only well-established risk factors for this disease are increased age, ethnicity and heredity [5–7].

According to the American Cancer Society statistics, in the USA the probability of contracting PCa rises greatly with age: for the males younger than 49 the value is 0.3%; for males older than 70 it is 11.2% [8]. These values are of great concern specially in developed countries taking into account their rise of life expectancy. Concerning ethnicity, black people show the highest incidence followed second by white [7, 8]. This incidence relation is also corroborated by figure 1.2. The heredity factor also shows great impact on PCa development: the European Association of Urology (EAU) guidelines point out that if one first-line relative has the disease, the risk of contraction is at least doubled [6]; general studies suggest that family history can be responsible for 5%-10% of the total cases [7].

Figure 1.2: Age-adjusted prostate cancer incidence and mortality rates for Black and White Americans. Data are age adjusted to the year 2000 population standard [7].

In figure 1.2 the variation through the years of both incidence and mortality of PCa in the USA are presented, where it is possible to associate specific time intervals to PCa diagnosis and therapy development. In the period of 1975-1990 the incidence rose 2% per year, a phenomenon that may be explained by the increased use of Transurethral Resection of the Prostate (TURP) for Benign Prostatic Hyperplasia (BPH) since half of the PCa were incidentally detected by this technique [7]. Around 1992-1995, a huge rise in incidence was caused by the adoption of the Prostate Specific Antigen (PSA) test. Since a large number of cases were diagnosed and a great amount of prevalence cleared, in further years the incidence lowered [7, 9]. In posterior years, from 1995 to 2010, it is registered a mean annual percent decrease of 2% in incidence and 3.1% in deaths. Decrease in death rate can be associated to the progress in screening, diagnosis and therapy techniques. For 2014, it is estimated that PCa will account for 27% of the newly diagnosed cancers and lead to 10% of cancer deaths [8].

All this data points to the fact that PCa represents a major health concern, lacks well-established preventable risk factors and therefore requires improvement in its early diagnosis.
1.2 Diagnosis

Currently, the main methods for the diagnosis of PCa are Digital Rectal Examination (DRE), serum concentration of PSA test and Transrectal Ultrasound (TRUS)-guided biopsy. Regardless of the strategy adopted, the final diagnosis depends always on histopathological confirmation [6].

DRE is a screening test where the clinician palpates the rectal wall with his finger in order to sense masses or abnormalities in the prostate. By looking to figure 1.1 it can be seen that the peripheral zone is adjacent to the rectal wall. Since this zone has the highest rate of PCa, this simple test may gather valuable evidence in early diagnosis. Before PSA implementation, DRE was the only tool for this purpose. It is estimated that 18% of PCa is tracked with DRE alone regardless of PSA levels [6]. Still, it is highly subjective and a patient may develop PCa without palpable abnormalities [10].

The PSA test consists of the serum concentration measurement of this antigen in the blood. PSA is a protein encoded by the prostate-specific gene kallikrein 3 that is secreted to the semen when it matures in its final form. Since in a healthy patient this concentration is low, an increase can be an indicative of a prostate malfunction in its structure or vascularization [11]. Currently it is considered that the higher this value, the higher the probability of the patient having PCa [6]. Still, this test is not specific as there are positively diagnosed patients with low PCa levels [9, 10]. In a recent study it has been estimated that patients with low PSA values showed considerable risk of contracting PCa (6.6% for 0-0.5 ng/mL) [6]. This exam has also risk of overdiagnosis because of the possibility of PSA levels being elevated in men with normal prostate or BPH [10]. In order to improve specificity, adaptations to the PSA test such as inclusion of PSA density, PSA density in transition zone, PSA velocity, Free-Total PSA Ratio (fPSA) and age-specific adjustments are being studied [6]. Nevertheless, PSA by itself is a better predictor than DRE.

TRUS-guided biopsy is the standard technique used for histopathological confirmation of PCa [12]. It is a procedure in which the clinician collects prostate tissue samples while real-time TRUS images are displayed. These images are continuously acquired by an ultrasound probe inserted in the rectum with its transducer facing the prostate as depicted in figure 1.3. During the acquisition, the biopsy needle is aligned with the probe through its attached needle guide. This way a coupling between probe and biopsy gun is ensured which allows the clinician to both move the probe and direct the needle to the area intended.
In the majority of situations, indications for prostate biopsy are an abnormal DRE or a high PSA value \[6, 13\]. Due to the lack of specificity of these exams, the indication for this biopsy is not universally established \[14\]. The cut-off PSA value alone where biopsy is considered is currently still undefined and a subject of debate in the major urology associations. Different recommendation criteria regarding PSA are currently adopted: the National Comprehensive Cancer Network (NCCN) considers both PSA velocity and fPSA; the EAU a cut-off of 2-3 ng/mL in younger men; the American Cancer Society a cut-off of 4 ng/mL for normal DRE and 2.5-4 ng/mL for abnormal \[13, 14\]. The decision of performing biopsy must also take into account potential comorbidities and therapeutical consequences \[6\]. Detailed information on development and improvements of this technique is presented in the section below.

### 1.3 TRUS-guided Prostate Biopsy Methodologies

The TRUS imaging technique was first developed in 1968 and applied later to prostate biopsy in 1988 \[15\]. From this time onward, TRUS-guided biopsy suffered various alterations and improvements in its methodology in order to rise its efficiency in PCa detection.

#### 1.3.1 Systematic Prostate Biopsy

In the first stages of TRUS development, cancer was thought to be identified by hyperechoic lesions visualized in the ultrasound prostate image. It was verified later that PCa was harboured most commonly in hypoechoic lesions and up to 25% of isoechoic lesions. This uncertainty in characterization of malignancy through ultrasound images led to the introduction of systematic biopsies - instead of targeting hypoechoic lesions, an acquisition in predefined anatomic regions of the prostate is performed \[15\]. With this approach, TRUS just gives feedback to the clinician about the anatomy of the prostate and the procedure stops being targeted \[12\].

The major concern in systematic biopsy is establishing the number of cores and the areas from which they are collected. The classic approach consists in a sextant division of the prostate: two
cores from the apex; two from the midgland; two from the base. This structure is described in figure 1.4.

Figure 1.4: Sextant division of the prostate: in the left, coronal view of the core location; in the right, axial view of three slices representing from top to bottom the base, midgland and apex. PZ=Peripheral Zone, TZ=Transition Zone, CZ=Central Zone, AFS=Anterior Fibromuscular Stroma, ED=Ejaculatory Ducts, PUT=Periurethral Tissue. Image from [12].

With the purpose of improving diagnostic yield, the midgland cores were adjusted to a more lateral position superimposed to the peripheral zone [12]. Although this approach granted better results than TRUS lesion targeting, the number of false negatives could reach 40% [15]. To overcome this, the concept of extended prostate biopsies where additional cores are sampled was explored. The current standard is the 12-core biopsy that consists of an extra of six more laterally directed cores in the peripheral zone besides the sextant template [13, 14]. A study claims that besides having a higher detection rate, this model diagnoses a greater amount of significant curable cancers [16]. The 12-core prostate biopsy template is represented in figure 1.5. In this mapping, there are four cores per each of the three axial prostate divisions and all of them are localized in the peripheral zone.

The number of cores taken is a current matter of discussion. Since the accuracy of this procedure decreases significantly with increasing prostate volume, it is currently debated the value of a volume or age adjustment to the core number [13, 14]. Although it may be thought that raising the core number above twelve would optimize the procedure, such strategy does not seem to present considerable benefit [6, 14]. In a study over a group of autopsies where a 18-core biopsy was performed (six central zone cores added to the already described twelve), it was observed that the majority of significant cancers were found with the 12-core technique. The authors suggested that core site location contributes more to detection rate than core number [14].

Despite the standard being the 12-core scheme, there is not an established universal recommendation for core number and location. The majority of urology guidelines consider a systemic 10-12
biopsy approach with addition of cores in suspicious areas detected either in DRE or in TRUS images. Extra cores in TRUS suspicious lesions can play a major role since in spite of not being widely used in targeting procedures, they currently represent the most important predictor of PCa [14].

All the paradigm information presented above regards mostly biopsies in patients that have never undergone it before. Due to the considerable probability of the procedure failing to detect small cancers, a repeat biopsy may have to be performed. The EAU indicates this repetition if the first is negative and subsequent DRE and PSA results persist suspicious [6]. It is estimated that initial biopsy misses cancer in around 30% of the patients [13], fact that can be explained by undersampling of specific prostate areas. This problem occurs most likely in the transition zone, the mid apex of the peripheral zone and the anterior horn of the peripheral zone [12, 14]. Sampling of the transition zone is currently thought as non-beneficial in first biopsy as the prevalence of PCa there is much lower [6, 13]. The templates recommended for repeat biopsy commonly include transition zone and anterior cores, but there is not a standard [14].

If even repeat biopsies do not track PCa and suspicion remains, a preoperative Magnetic Resonance Imaging (MRI) to the prostate becomes the best option for an accurate diagnosis [6].

1.3.2 MRI Based Approaches

Despite all the improvements described in the subsection above, the TRUS-guided biopsy still shows relevant drawbacks. On the one hand, systematic biopsies use no feedback about where the lesion might be located, making it a random sampling procedure. On the other hand, although TRUS-identified lesions are strong predictors of PCa, this image modality does not give enough coherent information for a targeted biopsy. Furthermore, PCa is currently the only cancer to be diagnosed by
an untargeted procedure [17]. These limitations point to the need of an image modality that both detects and localizes cancer suspicious sites accurately allowing posterior targeted sampling. Such requirement has motivated the implementation of methods using MRI, a technique that produces images where it is possible to depict clearly cancer lesions [12].

1.3.2.A Multiparametric Magnetic Resonance Imaging

In the last two decades MRI modality has gone through several upgrades until a point where it allows not only a precise anatomical description but also functional mapping of some molecular processes. Multiparametric Magnetic Resonance Imaging (MP-MRI) takes advantage of both of these features by evaluating different contrasts which offer a more explicit characterization of focal lesions. The contrasts typically included are T2-Weighted (T2WI), Diffusion Weighted Imaging (DWI) and Dynamic Contrast Enhanced (DCE). T2W images supply high spatial resolution data about the anatomy while the other two measure functional processes: DWI contrasts are based in the variation of water molecules diffusion through different tissues; DCE examines pharmacokinetic of the prostate after intravenous administration of a contrast agent. The use of this strategy is promising in distinguishing benign and malignant tissue [18][19]. As a predictor of PCa, MP-MRI is efficient in aggressive tumour detection and can work as a triage for biopsy, therefore avoiding unnecessary procedures. Still, its false positive rate is high and further development in lesion classification is required [20].

In spite of showing promise in being a PCa diagnosis tool by itself, MP-MRI is nowadays complementary to targeted biopsies as histopathological confirmation is still required. Presently, there are three main methods of integrating MP-MRI in the biopsy procedure: in-bore MRI-guided biopsy, cognitive registration and MRI-TRUS data fusion based software registration [18].

1.3.2.B In-bore MRI Guided Biopsy

In-bore MRI guided biopsy is a technique which uses real time MRI images for targeted sampling. During the procedure the patient is positioned inside the magnet bore and the operator performs needle navigation whilst receiving feedback about lesion and needle position from MRI images. In addition, the patient undergoes a previous MP-MRI exam in order to have more knowledge on the suspicion location [18]. This technique presents considerably better results than the standard systematic TRUS biopsy as several studies have obtained PCa detection rates ranging from 30% to 59% in patients with negative prior TRUS biopsy [19]. Also, it presents a reduced detection rate of insignificant lesions becoming a better option than repeat biopsies in terms of diagnostic yield. Still, it has several disadvantages associated: It is economically and time costly as each session may take up to 90 minutes due to the long duration of each MRI scan; it requires operator training as it depends greatly on his performance [18].
1.3.2.C Cognitive Registration

Cognitive registration is the simplest approach as it does not require any hardware or software alteration to the TRUS guided biopsy and its only MRI intervention is a previous MP-MRI exam. Registration can be defined as the estimation of the coordinate transformation that aligns two point data sets or images \[21\]. In this case, a mental registration between the preoperative MP-MRI and real time TRUS images is performed by the operator: after the lesion is identified in the MRI data set, he estimates the position to target in TRUS by knowing the relative position between the two image spaces \[18\]. The major limitation of this technique is the fact that its success relies greatly in experience, knowledge and performance of the medical practitioner. Furthermore, since the lesion may not be detected in TRUS, there is a considerable probability of missing small cancers \[19\]. Still, it shows higher detection rate than systematic biopsies and remains as an alternative that requires an experienced operator \[14\].

1.3.2.D MRI-TRUS Data Fusion

MRI-TRUS data fusion is a process where preoperative MRI and real time TRUS images and their features are assembled in a single one \[22\]. It is a software based approach which has been developed in the last decade and presents itself as a potential solution for overcoming problems faced by the two previously described methods: the biopsy procedure is maintained similar to the regular TRUS and therefore does not require additional training as the in-bore guidance \[18\]; the displayed guidance images are a fusion result where it is possible to visualize and target the lesions detected by the previous MP-MRI exam, something that shows a considerable accuracy improvement over cognitive registration alone \[19\]. Nowadays there exist several devices that are prepared for this type of intervention. The main differences between them regard the registration algorithms used and live needle position tracking techniques.

Generally, an overall registration software comprises several important steps: MP-MRI segmentation, real-time TRUS segmentation and registration. Segmentation is the process through which the prostate surface or contours are acquired from the images that contain it. Segmentation allows the fusion software to store prostate contour data of both MRI and TRUS images for a posterior 3D volume reconstruction. Registration then fuses the two volume data sets leading to the estimation of the position of the lesion \[18\].

The success of this procedure depends mainly in the quality of the software: the higher the accuracy of the registration, the higher the value of the fused images that aid the targeting \[18\]. The major reason of concern in registration of prostate images is the non-rigid mechanical behavior of the prostate. During the targeted sampling, the physical contact of the ultrasound transducer with the rectal wall induces a deformation in the prostate. Since this condition is not met in the MP-MRI acquisition, TRUS and MRI data sets may perceive the organ with slightly different shapes. An adaptation that several studies such as the ones performed by Xu et al \[23\] \[24\] use in order to minimize such effect is the addition of an endorectal coil in the MP-MRI exam, which not only simulates the mechanical stress induced by the ultrasound probe in the prostate but also grants higher image qual-
ity. Still, software fusion should take this deformation variable into account which leads to the need of elastic registration algorithms \cite{14,18}. In opposition to rigid registration that assumes the prostate as a rigid object only susceptible to rotation and translation, elastic registration considers this organ as a 3D incompressible object that has an elastic behavior \cite{25}. In figure 1.6 is presented a scheme that depicts the overlaying of MRI in TRUS using elastic or rigid transformations: the former can achieve an accuracy of less than 1 mm; the latter may have an error that surpasses 5 mm \cite{26}.

Besides the requirement of accuracy, the software implementation should also have both the lowest time complexity and user interaction possible \cite{27}. These factors are directly related: reducing the user interaction with the software implies the need for automatic routines to be implemented which in turn can require a much higher computational power expense.

Several approaches for fusion algorithms have been studied. They can be organized in three distinct groups taking into account the registration method used: fiducial based, surface based and model based \cite{28}. Fiducial based is the simplest approach as the registration is estimated through the correspondence of manually defined landmarks in both the MRI and TRUS images. The surface based methodology performs registration by minimizing the distance between MRI and TRUS prostate surfaces. In these two previous approaches, the local deformations of the prostate are commonly adjusted with Thin Plate Splines (TPS) or polynomials whose coefficients are calculated with specific cost functions \cite{25}. Model based estimates the transformation between data sets as the solution of a physical model equations set. Various recent models have been proposed in the literature: Risholm \textit{et al} \cite{29} and Hu \textit{et al} \cite{27} estimate prostate deformation with FEM analysis; Zhang \textit{et al} \cite{25} has reported a model with strain energy minimization; Sparks \textit{et al} \cite{28} reported a fully automatic process which uses a probablistic model for prostate location determination in TRUS.

Since registration depends on segmented prostate boundaries, efforts have been made to build
automatic and precise segmentation algorithms. TRUS segmentation is more regarded in this matter than MP-MRI segmentation since the latter is previous to the biopsy procedure and manual in most cases. TRUS images have several limitations which pose a considerable problem in their segmentation: the TRUS low Signal-to-Noise Ratio (SNR) makes it impossible to use pixel based algorithms; the difficulty in identifying pixelwise intensity patterns in different tissues disallows the differentiation of prostate and surrounding tissue through region characterization. Currently, the majority of the proposed approaches are semi-automatic, needing at least manual correction of the obtained boundaries for better results or identification of specific points for purposes of initialization [30]. Depending on the information used, these approaches can be classified as: shape based, where boundaries are the core information of segmentation; region based, where local intensity or statistics are considered in an energy minimization; supervised and unsupervised classification based whose principle is the classification of regions as prostate or background using image feature vectors [31].

Live tracking of the biopsy needle is needed in order to register the position and trajectory of the performed biopsies in the fusion environment. Three main strategies to accomplish this purpose can be highlighted due to their current use in clinically accepted state of the art MRI-TRUS fusion devices: external magnetic field generators, used in the Uronav system (Philips-Invivo, Gainesville, Florida); 3D TRUS image registration used in the Urostation device (Koelis, La Tronche, France); angle sensing encoders used in the Artemis device (Eigen, Grass Valley, California). In Uronav, the generated magnetic fields create a positioning system where the TRUS probe is localized with an attached tracker. In Urostation, the tracking is based in image registration alone but requires a specialized 3D TRUS integrated transducer. In Artemis, the probe is held by a robotic arm with angle sensors integrated in its joints, allowing for real time localization [18, 19].

Several studies on MRI-TRUS fusion guided biopsies concern these devices and report an overall improvement in the detection rate over the classical systematic approach. Using Uronav, Vourganti et al. [32] obtained 37% of PCa detection rate in patients with prior negative biopsy and Rastinehad et al. [17] found an increase of 12.4% relative to systematic biopsies with 86.7% of significant findings. Sonn et al. [33] obtained a detection rate of 34% with 72% of significant cancers using the Artemis while Fiard et al. [34] achieved 55% with 91% significant using the Urostation system. Although these results show promise on the MRI-TRUS fusion as a follow-up to prior negative biopsy, further trials and development on this technology are required.

1.4 3D TRUS Imaging

3D ultrasound is an imaging modality developed in the last two decades in which conventional 2D ultrasound images are assembled with appropriate software in order to create a 3D volume [35]. 3D TRUS can be defined as the application of this technique to the imaging of the prostate. The possibility of visualizing 3D TRUS volumes instead of 2D TRUS images during biopsy offers several advantages such as: traceability of both biopsy core locations and prostate shape; lower mental effort to the operator since the whole anatomy of the prostate can be perceived [36]. For an accurate
3D ultrasound image reconstruction of the prostate it is required not only 2D images but also the position and orientation at which they have been acquired. Therefore, in 3D TRUS methods it must be assured a precise measurement of the probe location in each acquisition. These methods can be separated in two distinct groups whose differences regard the strategy for enabling this feature: free-hand approaches and mechanical scanning [35].

1.4.1 Free-hand Approaches

In free-hand approaches position sensing is achieved through the attachment of a tracker to the probe: while the physician holds and moves the probe manually, the tracker is detected and positioned in a virtual coordinate system with appropriate equipment. Methods for this tracking process include: angle sensing encoders, where the probe is held by a mechanical arm whose joint angles are measured in real time; magnetic field sensors, where the probe tracker is a magnetic sensor which is localized by an external magnetic field generator; acoustic sensors, where the probe tracker is composed by a set of microphones that are localized through emission of sound pulses. Free-hand 3D TRUS may also be performed without position sensing but only if it is assured a more restricted protocol in which the physician must move the probe within a predefined range of motion at an approximately constant velocity, granting a set of regularly spaced images. Still, the reconstruction obtained should not be used for measurements such as prostate volume since no geometrical landmarks are established [35].

1.4.2 Mechanical Scanning

In opposition to free-hand, in mechanical scanning probe movements are executed by motorized mechanisms. Since the movements performed by these motors are precise and follow a well defined protocol, the set of 2D TRUS images obtained has also a well defined set of positions and orientations associated [37]. 3D TRUS mechanical scanning systems differ in two major aspects: the scanning method used and the location of their motors.

The scanning method determines the motion routine performed by the transducer and hence the geometrical relation between the 2D images obtained. In 3D TRUS two main scan types can be highlighted: rotational and tilt. In rotational scanning the transducer is rotated along its longitudinal axis creating a "propeller-like" set of 2D images. These images intersect in the rotation axis and have a regular angular spacing. In tilt scanning the transducer is slightly tilted in a way that the imaging plane is progressively rotated. In this case the set of images resembles a fan of equally angular spaced images that do not intersect as in the rotational approach [35]. The motor location defines the ultrasound transducer and the mechanical devices are coupled. Regarding this factor, two categories of 3D TRUS mechanical systems have been developed: systems where a 2D ultrasound probe is held and moved by external mechanical fixtures; systems where the motorized mechanisms are allocated inside the probe housing, called 3D integrated ultrasound transducers [37].

3D TRUS with rotational scanning is commonly used in prostate biopsy and typically takes advantage of external mechanical fixtures to rotate an end-fire probe as depicted in figure 1.7 (b). The
major drawback in this method is the significant artifacts that are generated when there is motion in the probe rotation axis. In order to avoid such phenomenon these mechanical systems are usually supported by a stabilization apparatus [37]. In opposition to rotational scanning, tilt approaches in prostate biopsy have a wider range of options not only in terms of motor location but also probe configuration.

In terms of probe configuration, although figure 1.7 (c) illustrates a tilt scan with a linear probe, in TRUS only endocavitial end-fire and side-fire probes are considered. Side-fire probes have transducers placed along its longitudinal axis while end-fire ones have a curved one dimensional array of transducers in their extremity. Although 3D TRUS rotational scanning uses an end-fire probe, the tilt method can use both probe types (side-fire tilted scan is described in figure 1.7 (a)). For the systematic biopsy procedure itself, none of these types is considered as the universal standard since their comparison has brought mixed results: one study of 2008 points to a higher PCa detection rate with end-fire which would be justified by an easier sampling of the peripheral zone [38]; a later study of 2012 suggests that there is no statistically significant difference in detection rate [39]. In 3D TRUS mechanical tilt scanning, it can be argued that due to their round-shaped tip end-fire probes may grant a higher movement flexibility and ease in the subsequent needle direction adjustments [38].

3D TRUS tilt scanning also uses both of the above mentioned motor-transducer coupling strategies. Currently, several ultrasound systems manufacturers are developing end-fire 3D integrated transducers whose tilt scanning is achieved through an internal motor whose function is to mechanically sweep back and forth the 1D transducers array at the tip of the probe. This mechanical process is called "wobbling" and is represented in figure 1.7 (c). The wobbling process has a discrete stepping: there is a predefined set of orientations at which the imaging array can be placed by the motor, and a 2D image is acquired at each of them. Therefore, in 3D TRUS acquisition these probes must be well stabilized while the wobbler covers a 3D volume and its positions are registered [37]. A more detailed
visualization of the wobbling-based imaging is shown in figure 1.8.

Nowadays the majority of ultrasound systems use external motors to perform the tilt scanning: although these systems are bulkier than 3D integrated transducers, the latter requires specialized software and hardware tools to interface with their internal motors [18, 35, 37]. Furthermore, external motors have great versatility since they can work with a wide variety of ultrasound transducers [37].

![Figure 1.8: Wobbling-based scan in terms of image geometry. In the left is represented the transducer curved array in the imaging plane. In the center, the field of view with a fixed wobber orientation and respective plane of imaging. In the right, the volume sampled with all wobbler positions. \( \gamma \) is the image angle in relation to the probe longitudinal axis; \( O(\gamma) \) and \( r_1 \) are center and radius of the curved array; \( O \) and \( r_2 \) the center and radius of wobbling. Image from [40].](image)

In terms of clinically approved state-of-art devices, Urostation uses a 3D integrated transducer in 3D TRUS reconstruction not only for lesion targeting but also biopsy needle tracking [18]. A recent study by Peltier et al using this device found a higher PCa detection rate and higher percentage of significant findings of 3D TRUS over conventional 2D TRUS in systematic biopsies [41]. Such results point to the superiority of the former in TRUS-guided biopsies.

### 1.5 Role of Robotics

Although targeted biopsy procedures have undergone several improvements regarding image quality, the accuracy of the process is still considerably influenced by the performance of the operator. The task in question is mentally challenging since a great hand-eye coordination is demanded: while one hand engages the biopsy needle and the other both holds and moves the probe, the TRUS images displayed for feedback are hard to interpret as they are 2D and have inverted motion [42]. Robotic devices are a recent trend used for image guided interventions that can overcome the accuracy problem due to their highly controlled motion: not only the probe can be held in a static position but also the needle insertion is much more precise [43]. Besides granting accuracy in targeting, these devices may operate as 3D TRUS acquisition tools.

The function of robotic systems in 3D TRUS acquisition is to perform the movements required in mechanical scanning with very high precision and real time position sensing [43]. Such is the case of the state-of-art Artemis device, where an end-fire probe is allocated in the end-effector of a robotic arm with four degrees of freedom (DOF) not only for needle tracking as mentioned in subsection 1.3.2.D but also for 3D TRUS imaging. The automated mechanical scanning used consists on the rotation of the probe about the anus of the patient within a defined angular range [36].
This potential of robotic systems in acquiring 3D TRUS, registering accurately needle positioning and perform accurate lesion targeting makes them a promising approach in TRUS-guided biopsies.

## 1.6 Proposed Implementation

In this work is presented the development of a hardware implementation that performs automated 3D TRUS acquisition using a 3D integrated transducer such as the ones described in section 1.4.2. The purpose of such tool is its further integration in a prostate biopsy directed robotized system. This robotized system can be structured in three main subsystems that intend to take into account the state of the art prostate biopsies developments described in the previous sections: an ultrasound system that is composed by a 3D integrated transducer and its manipulation tools 1.4, a robotic arm 1.5, a MRI/3D TRUS data fusion software tool 1.3.2.

The hardware used in the ultrasound acquisition will be a 4D endocavital transducer 4DEC9-5/10 1.9 and its manipulation hardware system, the SonixTABLET developed by Ultrasonix. The objective of this system is to collect 3D TRUS prostate volumes and synchronize the images that compose them with their respective wobbling angles. As in the conventional TRUS-guided biopsy, the transducer has a needle guide attached along its longitudinal axis.

![4DEC-9/5 3D Integrated transducer](image)

**Figure 1.9:** 4DEC-9/5 3D Integrated transducer: in the left a photography of the transducer with needle guide attached, image from www.analogicultrasound.com; in the right, a zoomed view of the wobbler encased in the probe extremity, image from the probe datasheet supplied by the manufacturers.

The robotic arm to be used is the KUKA Lightweight Robot (KUKA LWR). By coupling its end-effector with the 4DEC9-5/10 transducer, it is allowed not only transducer movement in the cartesian space but also synchronization between the acquired volumes and the transducer cartesian position. Regarding the software, a MRI/3D TRUS data fusion algorithm developed in [44] will segment the acquired prostate volumes and register them elastically in a preoperative MRI. This way, the position of the biopsy target is estimated and used to guide the robotic movement hence needle guide trajectory adjustment. A simplistic overview of the total system in a control perspective is described in the diagram of figure 1.10.
Figure 1.10: Simplistic description over the control scheme of the intended robotized system: the gray arrows between the three main subsystems depict TCP/IP communications; the dashed arrow depicts the allocation of the transducer in a prostate phantom; \( R \) and \( t \) represent the rotation matrix and translation vector of the robot end-effector respectively; \( k \) is the number of images acquired per 3D TRUS volume. The 3D TRUS acquisition system developed in this thesis is marked by the dashed line area.

In this implementation, the ultrasound acquisition system and the robotic arm are connected via TCP/IP to the computer that possesses the registration software tool. This computer can be perceived as the host of the system which will both command the other two subsystems and receive valuable data from them. In every iteration \( i \), the system follows a sequence of steps: firstly, the host computer manipulates the SonixTABLET to collect a volume of \( k \) images with a set of \( \theta_k \) wobbler angles associated; at the same time, the host also retrieves the end-effector position data \((R_i, t_i)\) of the robotic arm and subtracts it to the initial reference position \((R_0, t_0)\); once all the data is integrated, the software tool reconstructs a prostate surface from the acquired images, angles, and end-effector position variation data \((\delta R_i, \delta t_i)\); with this ultrasound surface and the preoperative MRI, a trajectory is planned and a new end-effector position \((R_{i+1}, t_{i+1})\). This process would be repeated until the needle trajectory is aligned with the biopsy target. During each cycle step the robot does not perform...
any movement, ensuring probe stability.

In the diagram the dashed line area borders the 3D TRUS acquisition setup that was developed in this thesis. In this setup the two first steps described above were implemented: synchronous acquisition of ultrasound images with robot position data. Also, the complete hardware setup required by the total implementation which includes synchronous communication between the subsystems and additional data acquisition components was assembled, thus leaving only space for further software development. Furthermore, with the purpose of setting a bridge to the reconstruction software, a methodology of probe orientation adjustment with robotic movement was studied. The softwares used in each subsystem were: Propello SDK 6.1.0 Beta version in the SonixTABLET; Fast Research Interface (FRI) for interfacing with KUKA LWR; MATLAB R2011b in the host computer for manipulation of the other systems and integration of received data.

To evaluate the developed acquisition system, the acquisition frequency was tested and several prostate phantom volumes were acquired with different probe orientations. With these volumes, it was possible to acknowledge how well the system identified the probe orientation variation in the acquired images.

1.7 Thesis Outline

The present thesis is structured in five chapters: chapter 2 is dedicated to the ultrasound acquisition subsystem and presents a study over the control of 3D integrated transducers using Propello. This includes the operation methods, the underlying computational processes and limitations. Also, the development of optimized acquisition routines based in this software is presented. Then, in chapter 3 all the implementations regarding the robotic subsystem are described. After, in chapter 4 are discussed the implications of inserting the registration tool software in the the subsystems developed in the previous two chapters. An initial approach for robotic movement is proposed after acknowledging such implications. The final application that comprises all the developed implementations is presented and tested in chapter 5. In these tests it is evaluated both the overall functioning of the system and the results of the robotic movement approach. Finally, in chapter 5 not only final conclusions on the acquisition system performance are drawn out but also guidelines for future work are presented.
2

3D Integrated Transducer Control

Contents

2.1 The SonixTABLET™ .................................................. 20
2.2 The Propello Software .............................................. 21
2.3 Remote Wobbling Control using Propello ....................... 22
2.4 Propello based software Implementations ....................... 27
In this second chapter all the software implementations concerning the 3D TRUS acquisition subsystem are described. Since there is currently lack of information on softwares that manipulate 3D integrated transducers, these sections also explain the overall functioning of the Propello software, its limitations and various other aspects. The presented conclusions are supported mainly by extensive experimental testing to the machine and information supplied by the Ultrasonix research wiki page [45]. Valuable feedback was also obtained through manufacturers support and their research forum.

In the first two sections 2.1 and 2.2 is presented an overall introductory description of the hardware and software used which are the SonixTABLET and the Propello SDK respectively. Then, in section 2.3 is presented a review over the Propello operation methodologies and all their computational underlying processes. Finally, in section 2.4 are developed two acquisition routines that explore all the capabilities and features described in 2.3 and are to be used in the final setup of this work.

2.1 The SonixTABLET™

As already mentioned in section 1.6 the system used for 3D TRUS acquisition was the SonixTABLET developed by Analogic Ultrasound for the manufacturer Ultrasonix. It is a tool that consists of a computer with two ports dedicated to the insertion of US transducers and software prepared for visualization of the images acquired through them. This computer can be operated in two different ways: Clinical Mode, where the user has only access to the main application of the software that is called Exam and has no control over the imaging parameters; Research Mode, where not only the user is allowed to change parameters, as the computer can be managed with Windows environment. In the latter, for purposes of further research and development Ultrasonix provides a wide variety of Software Development Kits (SDK) that can be configured and reprogrammed by the user.

![Figure 2.1: The SonixTABLET hardware: in the left, the system mounted in a wheeled cart displaying the Exam application; in the center the back of the computer with the two transducer entries; in the right the ports of the system. Images from [45].](image)

Although these SDK offer various special features in terms of imaging, only one of them is fit for the objective of the present work: Propello, a software derived from Porta SDK designed to manipulate 3D integrated transducers by communicating with their wobbling mechanism while ultrasound images
2.2 The *Propello* Software

In order to develop the automated 3D TRUS acquisition routines mentioned in section 1.6, several code changes and additions have been implemented in *Propello*. Even though this software comes as an executable application when obtained from *Porta* SDK download, its source codes can be rewritten and compiled in C++ environment by using the *CMake* tool. In figure 2.2 is represented a print screen of the application GUI whilst acquiring ultrasound images.

Graphically the only difference in this GUI from the original downloaded demo is the presence of the TCP/IP button in the menu bar. In terms of functionality, imaging with *Propello* demo comprises several steps. Firstly, a system initialization which establishes connection between the ultrasound ports and the software must be performed by pressing the left button of the set marked by number (2). Once there is connection, the button (3) is enabled and must be pressed to verify the presence of a 3D integrated transducer in the port. About 10 seconds later, the caption of the same button changes from "Detect" to the probe name and imaging can be performed by pressing the middle button of (2). The acquired images are displayed in (8) at a predefined frame rate and in order to stop the process the right button of (2) must be pressed. While imaging is inactive, various configurations can take place: in (4) automatic or manual mode wobbling can be selected and their parameters adjusted in (5) and (6) respectively using "Increment", "Decrement" and "Apply" buttons; in (7) parameters that regard image properties can also be adjusted using "Increment" and "Decrement" buttons. In the next sections a much more profound analysis over these commands with main focus on automatic
and manual mode operations is presented. Such study also allowed for comprehension over the mechanical wobbling specifications of the 4DEC9-5/10 end-fire transducer.

2.3 Remote Wobbling Control using Propello

In the development of 3D integrated transducer control through Propello, two major aspects should be taken into account: the position of the wobbler must be precisely registered and controlled during 3D TRUS acquisition since registration algorithms are to be applied afterwards; the overall acquisition should be the least time-costly possible since further integration in a real time robotic system is intended. These requirements led to the need of exploring the lower level functions of the software instead of just relying on the GUI functionalities.

2.3.1 The C++ Environment of Propello

Access to the lower level operations of Propello is granted by the C++ source codes generated by Cmake. In this C++ environment, it is possible to acknowledge not only which are the routines that each graphical event calls but also the lowest level functions used. After performing a considerable amount of tests, the most relevant functions for the purpose of this work were identified. Information regarding their input arguments, operation mode and executed action in the acquisition system are summarized in table 2.1.

<table>
<thead>
<tr>
<th>Function</th>
<th>Operation Mode</th>
<th>Arguments</th>
<th>Output Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>portaSetParamI()</td>
<td>General</td>
<td>prm parameter; int prmvalue</td>
<td>Sets the value of imaging parameter parameter to prm-value.</td>
</tr>
<tr>
<td>portaRunImage()</td>
<td>Automatic</td>
<td>None</td>
<td>Performs continuous ultrasound image acquisition using automatic wobbling.</td>
</tr>
<tr>
<td>portaStopImage()</td>
<td>Automatic</td>
<td>None</td>
<td>Stops image acquisition initiated by portaRunImage().</td>
</tr>
<tr>
<td>portaStepMotor()</td>
<td>Manual</td>
<td>bool sweep_direction; int step</td>
<td>Increments the wobbling motor position with a value step in the sweep_direction direction.</td>
</tr>
<tr>
<td>portaGotoPosition()</td>
<td>Manual</td>
<td>double pos</td>
<td>Moves the wobbler motor to position pos.</td>
</tr>
<tr>
<td>portaGetBwImage()</td>
<td>General</td>
<td>const char* buffer</td>
<td>Writes in buffer the data of the last image acquired in grayscale format.</td>
</tr>
</tbody>
</table>

Table 2.1: Most relevant basic commands of Propello and their operation mode, input arguments and action executed in the acquisition system. In the input arguments of portaSetParamI(), prm represents a specific data type of Porta SDK.

It can be observed that all the functions present in this table have the prefix “porta” in their name. This is explained by the fact that such functions belong to Porta SDK, implying that their
source code is not accessible. Since their executed actions are not elementary, there was a significant amount of inflexibility in the development of remote wobbling control strategies. In terms of operation mode, the functions marked with "General" are used in both manual and automatic. Although portaRunImage() and portaStopImage() are specially designed for automatic mode, they are also required for manual. Detailed discussion on the operation of all these functions and their relevance in the imaging process is presented in the subsections below.

2.3.2 Automatic Mode Imaging

Besides being the default imaging process used by Propello, automatic mode is the simplest wobbling method in terms of C++ code routine. The complexity of the process resides solely in its core function, portaRunImage(). In the GUI, achieving automatic mode imaging requires only pressing the center button of (2) whereas in C++ environment only portaRunImage() must be called. In this acquisition mode the motorized wobbler of the probe sweeps the transducer back and forward in a cyclical fashion whilst acquiring images as depicted in the scheme of figure 2.3.

![Figure 2.3: Mechanical wobbling routine during automatic mode imaging of Propello](image)

This cyclical mechanical process is ruled by three main parameters that can be adjusted in the GUI with Increment and Decrement buttons of area (6) or in the C++ layer with the function portaSetParamI(): Deg/Frame, the angular resolution of the motor movement, hence the angular step at which each image will be acquired; Frames/Volume, the number of steps taken by the motor in order to go from one edge of its own field of view to the other; \[\text{FOV}\] the field of view of the acquisition that corresponds to the maximum angular amplitude covered. These variables have a well defined relation expressed in equation 2.1 in which one of them is always determined by the remaining two.

\[
\frac{\text{FOV}}{\text{Degrees/Frame}} = \frac{\text{Frames/Volume}}{} \tag{2.1}
\]
The values that can be assigned to these parameters are specific of the transducer used: since Propello possesses this data for a defined set of transducers, the concerned values are loaded after the transducer detection performed during initialization. In table 2.2 all the allowed parameter combinations for 4DEC9-5/10 automatic mode are listed.

**Table 2.2:** List of automatic mode parameter allowed values. In each column are displayed all the possible parameter value combinations for each of the three permitted angular resolutions.

<table>
<thead>
<tr>
<th>Deg/Frame (%)</th>
<th>0.7317</th>
<th>1.4634</th>
<th>2.1951</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames/Volume</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOV (°)</td>
<td>13</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>9.5122</td>
<td>10.2439</td>
<td>10.9756</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>21</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>15.3658</td>
<td>16.0975</td>
<td>15.3658</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>27</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>19.7561</td>
<td>19.0243</td>
<td>19.7561</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>35</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>25.6097</td>
<td>24.8780</td>
<td>24.1463</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>41</td>
<td>23</td>
<td>15</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>30.0000</td>
<td>30.7316</td>
<td>28.5366</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>47</td>
<td>33.6584</td>
<td>32.9268</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>34.3902</td>
<td>33.6584</td>
<td>32.9268</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>55</td>
<td>27</td>
<td>19</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>40.2439</td>
<td>39.5121</td>
<td>41.7073</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>61</td>
<td>31</td>
<td>21</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>44.6341</td>
<td>45.3657</td>
<td>46.0975</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>69</td>
<td>35</td>
<td>23</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>50.4878</td>
<td>51.2194</td>
<td>50.4878</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>75</td>
<td>37</td>
<td>25</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>54.8780</td>
<td>54.1462</td>
<td>54.8780</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>83</td>
<td>41</td>
<td>27</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>60.7317</td>
<td>59.9998</td>
<td>59.2682</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>89</td>
<td>41</td>
<td>29</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>65.1219</td>
<td>65.8535</td>
<td>63.6585</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>95</td>
<td>47</td>
<td>31</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>69.5122</td>
<td>68.7803</td>
<td>68.0487</td>
</tr>
<tr>
<td>Frames/Volume</td>
<td>103</td>
<td>51</td>
<td>35</td>
</tr>
<tr>
<td>FOV (°)</td>
<td>75.3658</td>
<td>74.6339</td>
<td>76.8292</td>
</tr>
</tbody>
</table>

In order to have a deeper understanding over how these parameters affect the procedure in figure 2.3 the internal computational sequence of `portaRunImage()` must be disclosed. When this function is called, three main wobbler movements can be observed. Since the system must acknowledge in which angle the wobbler is positioned before starting, the first movement consists on setting it to its “zero” reference angle. Immediately after, the wobbler is moved to the angle coincident with the edge of the field of view that is farther from the reference position. Finally, with the field of view edges set, back and forth sweeping between them occurs until the operation is stopped. The main role of the automatic mode parameters is the calculation of these edges through the two expressions presented below in 2.2. During imaging, the wobbler position $\theta$ never surpasses the upper edge of the first expression nor the lower edge of the second.
\[
\theta \leq \text{rightcenter} + \text{Deg/Frame} \left( \frac{\text{Frames/Volume} - 1}{2} \right)
\]

\[
\theta \geq \text{leftcenter} - \text{Deg/Frame} \left( \frac{\text{Frames/Volume} - 1}{2} \right)
\]

(2.2)

Here, the center of the field of view is expressed as two different values instead of one. Such is explained by the fact that automatic mode collects each image along an angular transition between two discrete positions rather than in a static one. The red arrows in figure 2.3 represent these transitions in a theoretical case in which the value of Frames/Volume would be 5. By observation, it is possible to conclude that the number of positions is always equal to the Frames/Volume value plus one (there are 6 gray dashed lines). This implies that there is not a measurable field of view central position but a central frame instead, justifying why Frames/Volume values in table 2.2 are all odd numbers. The variables rightcenter and leftcenter refer to the upper and lower edge of this frame respectively and are dependent on the value of Deg/Frame. It is important to note that although the field of view should be and apparently is bisected by the probe axis and needle guide, it was not possible to confirm this fact since no strategy for determining accurately the central frame angular position was found.

The movements performed by portaRunImage() can be visualized as transitions between the wobbler geometric references depicted in figure 2.4: the reference angle where the wobbler is initially set is depicted as the first red dashed line; the second transition can be perceived as the transition between the reference and the second purple line; the automatic sweeping is the cyclical transition between the two purple lines.

### 2.3.3 Manual Mode Imaging

Manual mode imaging differs greatly from automatic as it allows the user to step the wobbling motor manually while keeping track on its current position. For this mode, the sequence of commands used in the GUI 2.2 is more complex as it comprises more steps. Initially, the option Manual must be set in the selection box (4) and the play button in (2) pressed. For the same reason as the one described for portaRunImage(), the button Home Motor must be pressed in order to place the motor in its “zero” reference. Once the movement is finished, the two fields from (6) are set to the value 0 and manual stepping can be performed by using the Increment and Decrement buttons below. The function of these buttons is to change the position of the motor by a predefined angular step and present the updated position in the Motor Location field. By following this method it is possible to acquire images from all the possible discrete positions in which the motor can be placed.

In the C++ environment this mode relies mainly on the functions portaGoToPosition and portaStepMotor. When portaGoToPosition(pos) is called, the wobbler is placed in the input angle pos but with one contingency: since positioning in every possible angle is not allowed, the function finds the permitted angle that is closest to the input pos. This command is mainly used for purposes of position initialization: for example, setting the motor to its reference consists on running portaGoToPosition(0). The function portaStepMotor(step,sweep_direction) is the core of manual mode imaging and represents the key feature of Propello, the manual stepping of the wobbler. When ran, the position of the motor is incremented by a value proportional the input integer step in a direction defined by the
boolean sweep\_direction. The function also returns the final step value to the workspace. In table 2.3 are presented all the possible discrete manual steps in terms of amplitude and direction.

Table 2.3: List of portaStepMotor() output values in degrees (°). bool and int refer to sweep\_direction and step respectively: rows fix the input value of the former; columns fix the input value of the latter.

<table>
<thead>
<tr>
<th>bool / int</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>-0.3655</td>
<td>-0.7301</td>
<td>-1.4620</td>
<td>-2.9240</td>
<td>-5.8480</td>
</tr>
<tr>
<td>false</td>
<td>0.3655</td>
<td>0.7301</td>
<td>1.4620</td>
<td>2.9240</td>
<td>5.8480</td>
</tr>
</tbody>
</table>

In this table it can be observed that the minimum angular amplitude that can be achieved is 0.3655° with step equal to 2. The remaining amplitudes are simply obtained by the multiplication of this reference step by integer powers of 2: for example, the lowest resolution 5.8480° is the product between 0.3655° and 16. The same relation is also verified in the inputs as 32 is the product between 16 and 2. After experimental testing on the limits of this function it was observed that the motor is not moved when the two lowest amplitudes are used, limiting the real maximum angular resolution to 1.4620°. This limitation is associated to a hardware issue: since the motor stepping commands are communicated to the probe by sending an electrical impulse of duration proportional to step, it can be argued that the duration implied by the two lowest amplitudes may not be enough to trigger motor movement. In terms of direction, a negative value angle represents a movement towards the “zero” reference whereas a positive represents a movement apart from it.

Tests over these manual mode functions also granted understanding over the coordinate system and mechanical range of the 4DEC9-5/10 wobbler. Such findings are summarized in figure 2.4.

Figure 2.4: Mechanical range description scheme of the 4DEC9-5/10 wobbler. The red dashed lines depict the mechanical stops of the wobbler. The green arrow depicts the total angular amplitude of movement. The purple lines and dotted purple arc represent the field of view of automatic mode imaging. θ₁ and θ₂ represent the offset angles between the mechanical stops and the base of the transducer.

Two main conclusions are drawn from this scheme: the total mechanical range of the transducer is around 100° starting on the “zero” reference position that coincides with the first mechanical stop; the
angular position of the mechanical stops relative to the probe axis are unknown (θ₁ and θ₂). The latter
leads to an uncertainty in finding the central position of the motor: since θ₁ is unknown, it is impossible
to detect the position at which the wobbler is orthogonal to the base. Also, the fact that the angular
position is always a multiple of the values output by \textit{portaStepMotor()} implies that its sum with θ₁ may
never be equal to 90°. Still, it is known that the exact central angle is around 63°, explaining why the
maximum values of \textit{FOV} in automatic mode do not surpass numbers near 76° as they must fulfill the
criterion in (2.3)

\[
\frac{\text{FOV}}{2} + (63 \pm \text{uncertainty}) \leq 100
\]

This means that an imaging field of view centered in the the probe axis must never cross the upper
mechanical stop of 100°. This criterion will be of major importance in the optimized manual imaging
routines described in the next section.

Besides the wobbling strategy, manual mode also differs from automatic in terms of image capture.
Although \textit{portaRunImage()} performs a complex routine that both moves moves the motor and acquires
images, it is possible to disable the movements and preserve the image capture. Such feature is
achieved by changing the value of a parameter variable named \textit{prmMotorStatus} with the command
\textit{portaSetParamI(parameter,value)}: when this value is in its default, 1, \textit{portaRunImage()} runs normally;
when it is 0 automatic motor movement is blocked. This is indeed the command used by the GUI
when selection box (4) is changed to \textit{Manual}. When the wobbler position is changed either with \textit{Home
Motor} or \textit{Increment} and \textit{Decrement} buttons, \textit{portaStopImage()} and \textit{portaRunImage()} are sequentially
called in order to reinitialize the image capturing for the new updated position. Therefore, images
are acquired in a static position and not in a transition as in automatic mode, hence one image
corresponds to a single wobbler angle.

2.4 \textit{Propello} based software Implementations

Once general understanding over the underlying C++ procedures of \textit{Propello} was obtained, it was
possible to develop acquisition routines that take into account the requirements of time complexity
and control over wobbling already described. Two main strategies were implemented and optimized:
one based in pure manual mode imaging; other based in a external electrical triggering method that
enables a "pseudo-manual" mode imaging. In order to understand the latter, matters concerning the
triggering functionality and imaging pipeline of \textit{Propello} are discussed in this section.

2.4.1 \textit{MATLAB-Propello} Communication

As already mentioned in section 1.6 the implemented acquisition system is remotely manipulated
by a host computer. Therefore, before describing acquisition routines it is necessary to understand the
process of communication between \textit{Propello} and the host. The basis of this communication was the
establishment of a TCP/IP protocol connection between \textit{MATLAB} in the host computer and \textit{Propello}
in the SonixTABLET. Such was allowed through the implementation of TCP sockets in both sides of
the connection: in MATLAB, such was performed only using tcpip command whereas in the C++ code of Propello a signals and slots [46] routine based on Qt software was necessary. Qt specific libraries are included in the C++ codes of Propello due to the fact that its GUI was developed with the Qt Designer software.

After having the socket codes functioning, the connection procedure was coupled to a newly added graphical object in Propello GUI: the TCP button marked by (1) in 2.2. When this button is pressed, two options show up: Connect To Host and Disconnect. The procedure of establishing connection consists in: running TCP/IP code in MATLAB with the IP of SonixTABLET; clicking in TCP → Connect To Host and inserting the IP of the host in the window that pops up after; pressing OK button and waiting until a connection confirmation window pops up; pressing OK in this new window. After this final click a string is sent to MATLAB in order to ensure that communication exchange of data is now operational. Screenshots of these windows are presented in figure 2.5. Disconnecting just requires pressing TCP → Disconnect in the GUI and closing or clearing the MATLAB socket.

![Figure 2.5: TCP/IP connection graphical windows implemented in Propello: in the left, the host IP selection window; in the right the posterior connection confirmation window.](image)

While the connection is active, the implemented signals and slots code in Propello triggers a data reading routine whenever the socket detects data sent by the host. In this data reading routine it is possible to build functions that perform events in the Propello application. So, the method through which MATLAB commands Propello consists in sending predefined strings that trigger specific routines based in the functions described in section 2.3. The set of used strings is listed in table 2.4.
Table 2.4: List of strings sent from MATLAB to Propello with their output action, data sent to the host, and strategy in which they are inserted.

<table>
<thead>
<tr>
<th>String</th>
<th>Strategy</th>
<th>Output Action</th>
<th>Received data</th>
</tr>
</thead>
<tbody>
<tr>
<td>'detect'</td>
<td>Both</td>
<td>Initializes system and detects transducer</td>
<td>string with detected transducer name</td>
</tr>
<tr>
<td>'freedom_res_f'r</td>
<td>Manual</td>
<td>Sets the manual mode imaging parameters</td>
<td>None</td>
</tr>
<tr>
<td>'switchblock'</td>
<td>Triggered</td>
<td>Changes the motor block status</td>
<td>string with block status</td>
</tr>
<tr>
<td>'center'</td>
<td>Manual</td>
<td>Places the wobbler in its approximate central position</td>
<td>double with wobbler position</td>
</tr>
<tr>
<td>'move'</td>
<td>Manual</td>
<td>Steps the wobbler</td>
<td>double with wobbler position</td>
</tr>
<tr>
<td>'acquire'</td>
<td>Manual</td>
<td>Steps the wobbler and acquires an image in the new position</td>
<td>double with wobbler position and uint8 array with image</td>
</tr>
<tr>
<td>'nogui'</td>
<td>Triggered</td>
<td>Retrieves an image from the pipeline</td>
<td>uint8 array with image</td>
</tr>
<tr>
<td>'garbage'</td>
<td>Triggered</td>
<td>Deletes an image from the pipeline</td>
<td>string confirming image elimination</td>
</tr>
</tbody>
</table>

From these strings, only 'detect' is used in both strategies as it simply initializes the system remotely. In C++ layer this command calls the functions initHardware() and OnDetect(): the former initializes the system (same as left button of (2) in the GUI 2.2, the latter detects the transducer (same as button Detect (3)). The remaining strings are discussed in the next subsections.

2.4.2 Manual Mode Optimized routine

In spite of presenting itself as a very efficient acquisition approach in terms of time complexity, automatic mode lacks flexibility in various aspects: it is not possible to change the field of view of the covered volumes; there is no tracking of the wobbler position; there is no way to control the wobbler stepping. Taking into account these limitations, it was decided that optimization of the acquisition using manual mode would suit better the objectives of the final implementation. The flowchart of the final code is presented in figure 2.6.

Three main aspects were explored in this routine: the parameter setting, the wobbling movement and image collection. Parameter setting is the step where four parameters present in 2.6 are defined: nvol, the number of volumes collected; frames, the number of images collected in each volume; step, the angular spacing between each consecutive image; upperlimit and lowerlimit, the imaging field of view edges. All of them except the first are calculated in Propello when the string 'freedom_res_f'r is sent: the 'res' substring defines the value of step and 'fr' the value of frames. With these two variables, a field of view is constructed by calculating the two positions that the wobbler can not surpass. The expressions in 2.2 are also applied here: the first calculates upperlimit and the second lowerlimit;
Deg/Frame and Frames/Volume are the same as step and frames respectively. The only modification is the use of a single center instead of a central frame, that is calculated as the multiple of step nearest to the probe axis. All these calculations are included in the initialization block.

The wobbling movement was achieved with the strings 'move', 'center' and 'acquire'. The first two strings call `portaStepMotor(movdir,step)` and `portaGoToPosition(center)` respectively and are used solely for positioning the wobbler in one of the limits before volume covering starts: firstly, the wobbler is put in its center and the motor position variable `m.motorpos` is initialized; after, string 'move' is sent until the wobbler is placed one step before the field of view limit. It is important to note that

![Diagram](image)

**Figure 2.6:** Optimized volume covering with Propello Manual mode: green blocks are operations in the C++ environment; blue are operations in MATLAB; orange refers to initialization and termination processes; dashed lines represent TCP/IP communications.

The wobbling movement was achieved with the strings 'move', 'center' and 'acquire'. The first two strings call `portaStepMotor(movdir,step)` and `portaGoToPosition(center)` respectively and are used solely for positioning the wobbler in one of the limits before volume covering starts: firstly, the wobbler is put in its center and the motor position variable `m.motorpos` is initialized; after, string 'move' is sent until the wobbler is placed one step before the field of view limit. It is important to note that
before stepping, it is checked if the wobbler position is coincident with one of the limits: if that is the case, boolean movdir changes. The string ‘acquire’ is used to perform the unitary operation of volume covering: stepping the motor and subsequent image collection in the new position. The stepping is done the same way as in ‘move’ and the acquisition is performed in a similar fashion to the one Propello has implemented for manual mode as default: running portaRunImage() and portaStopImage() sequentially. This reinitialization is necessary due to a significant limitation present in this software, the incompatibility between portaRunImage() and portaStepMotor(): if the former is running and a step is induced with the latter, imaging is immediately stopped and a reset is demanded. After, portaGetBwImage(buffer) is called to write the image data in the variable buffer that is to be sent to MATLAB. As it can be observed in the flowchart, between the imaging restarting and stopping 50ms are awaited. Such operation is mandatory as portaRunImage() needs this time to start collecting and putting images in the imaging pipeline, allowing portaGetBwImage() to access the last of them and write its data. Although different delay values were tested to reduce time expense of the procedure it was confirmed that the default 50ms used by Propello is around the lowest possible.

This routine has also great value in the matter that the center of the imaging field of view can be adjusted with some code refinements, hence allowing for a better response to the robotic movements intended by the final implementation. Still, the time cost is critically affected by the incompatibility already described and compromises the overall acquisition frequency severely. With the purpose of solving this issue, a routine involving external triggers was developed.

### 2.4.3 External Triggering in Propello

For purposes of synchronizing image acquisition with external equipment, the Porta SDK offers a synchronization signals feature: through a set of BNC entries that can be connected to the area depicted in the right image of [2.1] it is possible to both send and receive electrical triggers whenever an ultrasound image is collected by the system.

![Figure 2.7: SonixTABLET set of BNC connectors. Numbers designate the operational entries.](image)

There are three operational BNC cable entries [2.7] two dedicated to output triggers (numbers 21 and 25); one to input triggers (numbers with 26). In order to communicate with them, Propello has three specific parameters present in the area (7) of the GUI [2.2]: trigger in for cable 26; trigger out and trigger out 2 for cables 21 and 25 respectively. The values that can be assigned to these variables are 0, 1 and 2. When the value is 0, triggering operations can only be activated through implementations
using *Porta* specific functions in the C++ code. Otherwise, these operations are associated to the ultrasound image acquisition event. In terms of output, when the *trigger out* variables are set to a value different from 0, electrical triggers are sent through their respective BNC cables during image acquisition: if the value is 2, a trigger is sent per each acquired image; if the value is 1, triggers are sent per each acquired ultrasound line that composes a whole image. In terms of input, when *trigger in* is different from 0, image acquisition only occurs if the respective BNC cable receives an external triggers: if the value is 2, an image is acquired per each received trigger; if the value is 1, each trigger only activates the acquisition of a single ultrasound line.

Among these triggering operations, only one shows promise in reducing the time expense of the previous routine: the whole image acquisition triggering with *trigger in* set to 2. According to the Ultrasonix research wiki, for proper triggering the input signal should be a TTL wave with 3-5V of amplitude and length higher than 100ns. In order to produce a signal with such characteristics, a data acquisition board (NI USB-6008 model by National Instruments) was added to the acquisition system via the connections depicted in scheme 2.8.

![Figure 2.8](image)

*Figure 2.8:* NI USB-6008 acquisition board insertion in acquisition setup: the gray line depicts a coaxial cable that connects BNC 26 of SonixTABLET to the analog output channel ao0 (2) of the board; the blue line represents USB connection between the board port (1) and the host.

Once connected through USB to the host computer, NI USB-6008 input and output signals can be managed via *MATLAB* and its Session-Based Interface [47] library. In this environment, trigger generation was accomplished with a simple routine where the command `outputSingleScan(s, data)` is called in succession. When this function is called, all the analog output channels present in the variable *s* have their voltage value set to *data*: *s* is a data acquisition session object in which the input and output channels of a given device can be managed; *data* is a double that must be within the range of the concerned channels. In this case, *s* contained only one analog output (ao0) whose range is [0, 5V]. This way, signals can be output through ao0 by calling `outputSingleScan(s, data)` *n* times successively with a predefined ordered set of *n* amplitudes *data*: for the required trigger a *data* sequence `{0 3 0}` was used. In figure 2.9 is plotted the resulting pulse in ideal conditions.

The length of this pulse corresponds to the time in which NI USB-6008 outputs a single sample: \( \frac{1}{150} \) seconds since its analog output frequency is 150Hz. The fact that this duration is much higher
than the mentioned 100 ns did not compromise the process since the ultrasound system identifies the trigger by looking into their rising edge.

Initially it was expected that an external trigger would make Propello perform the simple operation of collecting an image if trigger in was set to 2. This led to the idea that the time complexity of the manual approach could be reduced just by sending a trigger instead of proceeding to a portaRunImage() reinitialization in each motor position. Still, these expectations were proven wrong since experimental trials brought evidence that the triggered image acquisition has a complex underlying process that involves portaRunImage().

Propello only activates this feature if two requirements are met: portaRunImage() must be active and trigger in must be changed to 2 either through Increment and Decrement buttons in the GUI area or by modifying the parameters xml file. After completing these two steps, Propello enters in an imaging mode which is manually controlled through external triggering and uses exactly the same wobbling parameters and image acquisition method as automatic mode. As an initialization, the system performs the same initial wobbler movements as portaRunImage(): a placement in the "zero" reference angle followed by a transition to the field of view edge calculated through the first expression of equations. At this stage, the Propello does not perform any action unless a trigger is detected. In case of detection, the system both collects an image and induces a step of amplitude equal to Deg/Frame in the wobbler. In fact, this process is precisely the unitary acquisition operation of portaRunImage() depicted by the red arrows in figure. This means that external triggering offers some control over automatic mode as it can decompose the automatic cyclical sweeping in discrete steps. Through this method not only it is possible to plan how many frames are to be acquired but also predict in which position the wobbler will be placed during the process. For example, if the wobbler is in the field of view edge and a number equal to Frames/Volume of triggers is received, a single volume is acquired and each position can be estimated as a multiple of Deg/Frame. Detailed position prediction for a given set of triggers is described in the next subsection.

Another relevant matter of discussion in triggered acquisition is the way collected images are made accessible in the imaging pipeline. Whenever a trigger is sent, the corresponding image is only placed in the imaging pipeline for posterior storage once another trigger is received. Such is explained by
the fact that this collector thread ensures that a frame has been correctly acquired by looking to the header of a newly acquired image, hence requiring a second trigger. This process is repeated along a sequence of triggers: when the trigger number \( n \) is sent, the acquired frame number \( n-1 \) becomes accessible and so on. The scheme in image 2.10 depicts this collector thread mechanism.

**Figure 2.10:** Imaging pipeline in *Propello* triggered acquisition. Numbers 0,1,2 and \( n \) identify the sequence of acquired images. Software Interrupt indicates the time at which each image becomes available for data writing. Image from [45].

The function used to get the image data from this pipeline is the same used in the previous approach, *portaBwImage(buffer)*. To use it properly, it is important to note that the images are retrieved from the pipeline as if they were in a queue: if two images were acquired and only the second is of interest, the *portaGetBwImage(buffer)* must be called twice anyway to ensure that the first is cleared.

### 2.4.4 External Triggering Optimized routine

Taking into account all these findings, it can be concluded that the imaging reinitialization issue presented by the manual approach can not be avoided through triggered acquisition since trigger reading is only activated while *portaRunImage()* is running. Still, this acquisition mode by itself has a dramatically lower time expense than manual and even though limited, it offers some control in terms of wobbler movement and position tracking. For this reason a new routine based in this acquisition was implemented.

The overall routine is depicted in the flowchart 2.11. The initialization block comprises two steps: parameter setting and trigger initialization. In the former, the imaging parameters are set using the GUI area (5) since the triggered wobbling is ruled by automatic mode parameters. After setting them, they are registered in *MATLAB* manually with the following correspondences: *frames* is *Frames/Volume* and *step* is *Deg/Frame*. Additionally, *nvol* is defined with the same purpose as in the manual routine. For trigger initialization the variables for manipulation of the ao0 channel are created. After, its output is set to 0 V by calling *outputSingleScan(s,0)*.
In the subsequent volume acquisition, two aspects must be considered: image handling and position estimation. Image handling is performed through triggering the system and sending the strings ‘nogui’ or ‘garbage’ after. When each of these two strings is received by Propello, portaGetBwImage(buffer) is called. What differs in their action is the destination of the obtained image data buffer: ‘garbage’ leads to its elimination; ‘nogui’ leads to its dispatching to the host MATLAB. It can be seen in the flowchart that the total amount of triggers sent during the procedure is the expected \( frames \times nvol \) plus an extra of \( frames + 1 \). This strategy was adopted to overcome two problems inherent to the imaging pipeline: after system initialization, the first images acquired through triggering have a much lower intensity than expected; as already mentioned, each image is only accessible after triggering.
the acquisition of the next one. In order to solve the first problem, a "dummy" sweep is performed by sending a number frames of triggers, not only ensuring that all the initial problematic images are cleared with string ‘garbage’ but also positioning the motor in the lower limit of the field of view. For the second problem, a single additional trigger is sent to clear the last image of the "dummy" sweep. Therefore, the purpose of the first frames+1 triggers is to eliminate unwanted images.

Position estimation is performed by predicting the movements that an amount of triggers will induce in the wobbler. Given a number of triggers $n$, the angle $\theta_n$ is calculated through the function (2.4)

$$\theta_n = \begin{cases} 
    \text{FOV} - \text{mod}(n - 2, \text{frames}) \times \text{step} & \text{if } \text{floor} \left( \frac{n - 2}{\text{frames}} \right) \text{ is even} \\
    \text{FOV} - (\text{frames} - \text{mod}(n - 2, \text{frames})) \times \text{step} & \text{if } \text{floor} \left( \frac{n - 2}{\text{frames}} \right) \text{ is odd}
\end{cases} \quad (2.4)$$

This equation is divided in two branches, one for each sweep direction. The direction is identified by measuring how many volumes have already been collected: if an even number has been collected, the wobbler is moving towards the lower limit; otherwise, it is moving towards the upper limit. The angles that these expressions return are within a reference in which the lower limit is the angle 0 and the upper limit angle FOV. Since the wobbler is placed in the latter after initialization, the expressions are always the result of a subtraction to FOV. It is possible to notice that the $n$ value has always a delay of 2 when used. Such modification is introduced to compensate for the image-trigger delay of the pipeline 2.10 and to associate each image to the starting position of its angular transition. For example, in the case of the first acquired image with just 2 triggers, the position value is simply FOV.

In case of repetition of the routine an additional reinitialization must be followed due to the fact that the wobbler will not be positioned in the upper limit after finishing the sweeping. In order to recover the upper limit position hence guaranteeing correctly calculated positions, the string ‘switchblock’ is used. When it is sent to Propello, the value of prmMotorStatus is changed: if the motor is currently blocked, it will be unblocked and vice versa. Apparently, such alteration would have no use for the reinitialization, but it was observed that blocking and unblocking the motor does affect the wobbler position: if the motor status is changed to blocked, the wobbler will make a movement to a position around the center; if it is changed to unblocked, the wobbler performs the initialization to the upper limit. Therefore, if ‘switchblock’ is sent twice, the initial position will be accurately reset.

2.4.5 Triggered versus Manual

Although two routines have been implemented, only one of them was selected for posterior integration in the final acquisition system. Along their description in the previous sections several pros and cons have been presented. In terms of wobbling control, the manual approach is excellent as it can explore all the mechanical capabilities of the transducer whereas the triggered is highly restrained by the automatic mode parametrization. Regarding the time complexity matter, it was thought that the triggered approach would offer the best results. In order to validate this statement, the time at which each approach covers a specific number of frames was measured five times. The results for a single volume with 51 frames are summarized in table 2.5.
Table 2.5: Time expense of triggered and manual routines during coverage of a volume with 51 images: the calculated frequencies are result of dividing each time value by the number of frames.

<table>
<thead>
<tr>
<th>Trials</th>
<th>Manual Routine</th>
<th>Triggered Routine</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Spent Time (s)</td>
<td>Frequency (fps)</td>
</tr>
<tr>
<td>1</td>
<td>11.2073</td>
<td>4.5510</td>
</tr>
<tr>
<td>2</td>
<td>11.3927</td>
<td>4.4766</td>
</tr>
<tr>
<td>3</td>
<td>11.2684</td>
<td>4.5259</td>
</tr>
<tr>
<td>4</td>
<td>11.4734</td>
<td>4.4451</td>
</tr>
<tr>
<td>5</td>
<td>11.2813</td>
<td>4.5208</td>
</tr>
<tr>
<td>Mean</td>
<td>11.3264</td>
<td>4.5039</td>
</tr>
</tbody>
</table>

These results confirm the significantly lower time complexity of triggered acquisition: its mean acquisition frequency is above the triple of the one obtained by the manual approach \( \frac{14.6760}{4.5039} = 3.259 \).

Another matter of comparison is the correlation between the acquired images and the wobbler position: in the manual approach, each image has a static angle associated; in the triggered, images are acquired in angular transitions. This transition based acquisition may pose a problem since the wobbler movement induces a slight image distortion which can compromise the posterior reconstruction results.

Despite the fact that manual is a much more consistent approach in terms of parametrization and image quality, the low time expense is a major priority in a real time application. Therefore, it was decided that the triggered approach would best fit the final implementation objective.
3 Robotic Coupling

Contents

3.1 The KUKA Lightweight Robot ........................................... 40
3.2 The *MATLAB*-FRI Communication ................................... 41
3.3 Mechanical Coupling ...................................................... 43
In the third chapter of this thesis are presented all the implementations regarding the insertion of the KUKA LWR in the 3D TRUS acquisition system. These include the communication between the robot and the host computer and the mechanical coupling between robot and 3D integrated transducer.

In the first section 3.1 an overview of the KUKA LWR system and its control architecture is described. After, the implementations concerning the communication between KUKA LWR and the system host are described in section 3.2. Finally, in section 3.3 the mechanical coupling between the 3D integrated transducer and the lightweight robot is presented.

### 3.1 The KUKA Lightweight Robot

The KUKA LWR is an articulated robotic arm developed by KUKA specifically for tasks involving human-robot interaction. For this purpose, it presents several features such as: a kinematic redundancy that resembles the human arm since it possesses seven joints with one rotational DOF each (the human arm has three DOF in the shoulder, three DOF in the wrist and one DOF in the elbow); a high power-to-weight ratio that confers ease in being moved through human touch; torque sensors integrated in all joints which establish safety limits [48].

![Figure 3.1: The KUKA Lightweight Robot system: (1) is the lightweight robot; (2) is the KUKA Robot Panel (KRP); (3) is the KUKA Robot Controller KRC. Image from [49].](image)

In figure 3.1 is presented a photo of a KUKA LWR and its manipulation hardware, the KUKA Robot Controller (KRC) and the KUKA Robot Panel (KRP). Interfacing with this robot can be achieved...
through two distinct methods: operating directly with the KRP or using the KUKA Fast Research Interface (FRI) with a remote computer. In the KRP the robot is manipulated through specific buttons and scripts written in KUKA Robot Language (KRL). By pressing the buttons it is possible to perform simple operations such as moving the end-effector or changing the angle of each joint separately. Through the KRL scripts the robot can be commanded to perform movements with predefined trajectories and control methodologies. The FRI is an interface designed with the purpose of establishing low level communication between the KRC and an external host computer. This communication is based in the UDP protocol and allows the host to access KRC and its variables at a frequency of 500Hz [48]. In the diagram of figure 3.2 is represented a simplified overview of the KUKA LWR system communication and control architecture.

![Figure 3.2: KUKA LWR control architecture: the red blocks are KUKA system components; the white block is the FRI host computer. Simplified diagram based in [49].](image)

In order to establish the FRI communication, it is required the use of FRI specific C++ libraries in the host and a KRL script with FRI specific functions in the KRP [48]. Once there is connection, the host can interact with the KUKA LWR through programs developed in C++ environment.

### 3.2 The MATLAB-FRI Communication

As already mentioned in section 1.6 it is intended the synchronous acquisition of the robot position data with the acquired ultrasound volumes. This means that the host computer must request this data at the exact same moment of ultrasound image acquisition. To achieve this goal, a communication between the host of the total system using MATLAB and the host of KUKA LWR with FRI was implemented. The communication was separated in two components: one based in electrical triggers for requesting data and another based in TCP/IP for data retrieval.
3.2.1 Triggered Data Requesting

Since the ultrasound image acquisition event is already activated through sending a trigger signal to Propello, it was decided that synchronization would be achieved by using this same signal to request the position data from the FRI host. In order to allow the FRI host to receive and process such signals, another data acquisition board NI USB-608 was added to the hardware setup via the connections represented in the scheme of figure 3.3.

![Figure 3.3: Second NI USB-608 acquisition board insertion in acquisition setup: the blue and gray lines represent the same connections as in image 2.8; the orange line depicts a connection between the analog output channel ao0 (2) of the trigger sending board and analog input channel ai1 (3) of the receiving board; the green line depicts a USB connection between the FRI host and the receiving board port (4).]

In terms of software, a thread based in functions of the NIDAQmx library [50] was implemented in the existing FRI C++ program. In a similar way to the Session-Based Interface of MATLAB, these functions allow the management of the voltage in the channels of data acquisition devices in the C++ environment. In this case, it was defined a variable to read the analog input channel ai1 which was connected to the analog output ao0 of the trigger sending board. This way, while the FRI is communicating with KUKA LWR, the values in ai1 are sampled at a rate of 10kHz. For triggering detection, the thread always compares the current with the previous channel value: if their difference is above a predefined positive threshold, the rising edge of a trigger has been received. This method is quite accurate since the acquisition frequency in ai1 is much higher than the output rate of ao0. The threshold value used was 4, requiring the pulse to have an amplitude higher than the 3V referred in the section 2.4. This choice is explained by the fact that a distinction between triggering both image
acquisition and position request and triggering image acquisition alone was intended.

If a trigger is detected, the confirmation message “Trigger detected!” shows up in the command line and specific CHAI3D functions [48] are called to measure and store the robot end-effector position data. This data consists of a $[3 \times 3]$ rotation matrix $R$ and a $[3 \times 1]$ translation vector $t$.

### 3.2.2 TCP/IP Data Retrieval

Once the FRI host has the synchronized variables $R$ and $t$, a TCP/IP communication is used to send them to the MATLAB host. Just as in subsection 2.4.1 in order to establish it, TCP sockets were implemented in both connection sides: in MATLAB a new socket was added again with the tcpip function; in the FRI C++ environment a routine based in the Windows Sockets libraries was used [51]. The TCP/IP connection must be established before starting any other operation such as trigger detection or communication with the robot. As a confirmation that this requirement has been fulfilled, the command line outputs the string “Connection has been established”.

Since it is not possible to send double type values through the sockets, the position data variables are stored and sent in a string variable. Given the rotation matrix and translation vector definitions of 3.1, this string has the form 

$$R = \begin{bmatrix}
R_{xx} & R_{xy} & R_{xz} \\
R_{yx} & R_{yy} & R_{yz} \\
R_{zx} & R_{zy} & R_{zz}
\end{bmatrix}
$$

$$t = \begin{bmatrix}
t_x \\
t_y \\
t_z
\end{bmatrix}$$

(3.1)

### 3.3 Mechanical Coupling

The mechanical coupling is the process through which the 3D integrated transducer becomes coupled to the end-effector of the KUKA LWR. To accomplish it, not only the probe must be allocated mechanically in the robot but also a software tool calibration in the KRP is required. For the allocation, a 3D CAD coupler was designed in SolidWorks and printed afterwards. This coupler was composed of two pieces that form a shell that encases the 4DEC9-5/10 transducer when assembled. The design of such shell was only possible since the CAD model of the transducer was supplied by the manufacturers. For attachment to the KUKA LWR end-effector the coupler presents a circular base with a specific arrangement of screw holes. In figure 3.4 is represented this assembly with CAD models.
The tool calibration is a procedure through which the end-effector of the robot is updated to a reference point of a tool attached to it. This means that if an end-effector position measurement is requested, the output data will be relative to the new reference point. The same will happen in terms of movement trajectories: for example, if a rotation around the end-effector is commanded, the output will be a rotation around the newly calibrated point. In this case, the tool is the probe plus coupler set of figure 3.4 and the intended reference point is the center of the wobbler rotation $O$ described in figure 1.8.

The calibration was executed in the KRP with two specific methods developed by KUKA system software [52]: firstly the location of the point is measured through the XYZ 4-Point method; secondly, the orientation of the tool is defined with ABC 2-Point method. Once these steps are completed, the KRP generates a tool file which has six variables: the three position coordinates relative to the robot end-effector; the three axial rotations relative to the end-effector local frame of reference. In table 3.1 are summarized the experimental values obtained.
Table 3.1: Obtained calibration values for the probe plus coupler tool. The maximum error estimated in the translation is 1.8 mm.

<table>
<thead>
<tr>
<th>Direction</th>
<th>Translation(mm)</th>
<th>Axis</th>
<th>Rotation (°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>205.216</td>
<td>X</td>
<td>-166.373</td>
</tr>
<tr>
<td>Y</td>
<td>191.356</td>
<td>Y</td>
<td>-0.589</td>
</tr>
<tr>
<td>Z</td>
<td>107.636</td>
<td>Z</td>
<td>44.040</td>
</tr>
</tbody>
</table>

In the photos of figure 3.5, is shown the real mechanical coupling. Also, the local frames of reference of interest are depicted: \((x_b, y_b, z_b)\) is the frame of reference of the robot base and all the translation measurements are relative to its origin; \((x_e, y_e, z_e)\) and \((x_p, y_p, z_p)\) are the end-effector and tool reference point frames of reference respectively which are related by the values of table 3.1.

Figure 3.5: Mechanical coupling between KUKA LWR and 4DEC9-5/10 transducer: the set \((x_b, y_b, z_b)\) is the robot base frame of reference; the set \((x_e, y_e, z_e)\) is the end-effector frame of reference; the set \((x_p, y_p, z_p)\) is the probe frame of reference.

In order to finalize this step, it is necessary to put the new tool file in the FRI communication KRL script. This way, the data retrieved by the method described in the previous section will be relative to the transducer position and orientation.
Registration Software Implications

Contents

4.1 Registration Algorithm Overview ........................................ 48
4.2 Semi-Automatic Segmentation ........................................... 49
4.3 The shape parametrization problem ................................. 51
4.4 Probe orientation adjustment Strategy ............................. 53
4.5 Robotic probe movement Implementation ......................... 56
This fourth chapter is focused on the influence of the chosen registration software algorithm on the previous two subsystems operation. Particular emphasis is given to the segmentation process, whose results are highly dependent on the robot position data and 3D TRUS image data set. After unraveling the nature of such dependence, a methodology of probe orientation adjustment is proposed and implemented.

The chapter starts with a summarized description over the registration algorithm to be used in section 4.1. After, special emphasis is given to the semi-automatic segmentation process in 4.2. The influence of the 3D TRUS image data set and robot position data in the results of this process is discussed in 4.3. Finally, the probe orientation strategy that takes into account this influence is proposed in section 4.4 and implemented in section 4.5.

4.1 Registration Algorithm Overview

As already mentioned in section 1.6, the registration software to be used in the MATLAB host computer is a MRI/3D TRUS data fusion based algorithm developed in [44]. The main stages of this algorithm are summarized in the scheme of figure 4.1.

![Workflow of the MRI/3D TRUS data fusion algorithm. Image from [44].](image)

Figure 4.1: Workflow of the MRI/3D TRUS data fusion algorithm. Image from [44].

Before starting this algorithm in the surgical environment, it is necessary to perform a preoperative MP-MRI examination to the concerned prostate. This exam grants not only a set of images from which a prostate 3D surface can be extracted but also the location of suspicious cancerous tissue.
The surface extraction procedure consists in segmenting manually all the 2D prostate images and interpolating the resulting contours with Radial Basis Functions (RBF). After completing it, a 3D model of the prostate is stored and ready to be used for real time registration. The landmarks of interest such as suspicious lesions or systematic biopsy cores can also be stored in this model.

In surgical environment, the reconstruction process holds several differences. Firstly, a TRUS image data set of the prostate is acquired. Then, in order to reduce greatly the time expense of the overall procedure, a semi-automatic segmentation based in Active Shape Models (ASM) takes place. As in the MP-MRI segmentation, this process also results in a set of contours that is subsequently interpolated using RBF, creating a 3D TRUS prostate surface. At this point, a registration process will estimate the coordinate transformation between this real time ultrasound surface and the preoperative MP-MRI one. The registration is separated in two parts: firstly a rigid one based in Iterative Closest Point (ICP) that only accounts for translations and rotations; secondly an elastic one based in TPS which will compensate for the prostate deformation phenomenon induced by the ultrasound probe contact.

4.2 Semi-Automatic Segmentation

From all the algorithm steps in figure 4.1, the one that is most affected by the ultrasound system acquisition process and accounts for the larger computational expense is the ASM semi-automatic segmentation. ASM are models whose objective is to guide the segmentation process using prior information about the shapes that are to be extracted. These models comprise two main components: a Statistical Shape Model (SSM) that provides the prior information; an image search procedure that adapts this information in order to fit the experimental data of the images to be segmented.

4.2.1 ASM components

The SSM is a model whose objective is to supply relevant statistical data about the prostate gland global shape. In order to have such data, this model must be built using several prostate image data sets from different patients. By manually segmenting these images and interpolating the resulting contours with RBF it is obtained a training set composed by different prostate surfaces. Since the ASM segments the prostate in 2D images, the statistical shape data must refer also to 2D slices of the prostate, requiring the reconstructed surfaces of the training set to be sampled in 2D contours. Regardless the geometrical parametrization of this sampling, a prostate shape is defined computationally in the model as a \( [m \times 3] \) matrix \( s \) which contains the 3D coordinates of all the sampled contour points. This data structure is represented in equation 4.1.

\[
\begin{bmatrix}
  s_1 \\
  s_2 \\
  \vdots \\
  s_m
\end{bmatrix} = \begin{bmatrix}
  x_1 & x_2 & \cdots & x_m \\
  y_1 & y_2 & \cdots & y_m \\
  z_1 & z_2 & \cdots & z_m
\end{bmatrix}^T
\]  

(4.1)

Once a training set of \( n \) shapes \( s \) is obtained, two statistical variables are calculated through a Principal Component Analysis (PCA), the mean shape \( \bar{s} \) and the covariance matrix \( C \) of the popula-
tion. The first is solely used for purposes of shape initialization and is defined by the expression in equation 4.2

\[
\bar{s} = \frac{1}{n} \sum_{i=1}^{n} s_i
\] (4.2)

The second variable is used to calculate the principal variation modes of the prostate shape and is given by the expression in equation 4.3. These modes are obtained through a singular value decomposition of \( C \) and can be perceived as eigenshapes, a set of orthogonal vectors that define the prostate shape space of the model. This means that any shape can be decomposed as a linear combination of these eigenshapes. Unlike \( \bar{s} \), this matrix is used throughout the whole segmentation process.

\[
C = \frac{1}{n - 1} \sum_{i=1}^{n} (s_i - \bar{s}) \cdot (s_i - \bar{s})^T
\] (4.3)

In the 2D segmentation process, the role of this variables is to supply an initial estimation of the contour to be segmented. The image search procedure is the process through which this initial estimation is adjusted to the boundaries present in the image. Its method consists in pushing iteratively each point that composes the estimated contour towards an area with higher hypoechoic-hyperechoic contrast, hence a probable boundary area.

### 4.2.2 The Segmentation procedure

Computationally, the segmentation is a procedure whose aim is to extract contours from a set of prostate ultrasound slice images. For each of them, the contour is calculated as the set of points \( c = \{p_1, p_2, \ldots, p_k\} \) that minimizes an energy function that includes three distinct terms: one that accounts for the SSM prior shape; another for the image search procedure; an additional one that introduces smoothness to the contour, ensuring the extraction of a more realistic shape without discontinuities. The generic expression is presented in equation 4.4

\[
E(p_i) = E_{shape}(p_i) + E_{search}(p_i) + E_{smooth}(p_i)
\] (4.4)

In this equation, the \( E_{search}(p_i) \) and \( E_{smooth}(p_i) \) terms adjust an initial contour that is associated to the \( E_{shape}(p_i) \) term and supplied by the covariance matrix and mean shape of the SSM. For the first image to be segmented, the initial estimation is simply the mean shape of the model. This means that without knowledge about the new prostate, it is estimated that its shape will be similar to the mean of the training set. Depending on which is the anatomical slice of the prostate that has been acquired, the SSM will superimpose the respective mean contour to the image. Once this first slice is segmented after energy function optimization, the covariance matrix makes a new estimation of how the prostate shape will be by knowing the principal modes of shape variation and the shape of the previously segmented contour. In the next slice, the SSM will make the same anatomical correlation as in the first and superimpose a contour from the newly estimated prostate shape. This initial shape estimation is repeated for all the acquired slices until the whole prostate is segmented.
The anatomical correlation of the contours performed by the SSM is directly related to the shape matrix parametrization. The parametrization used in the work in which this algorithm has been developed \cite{44} consists in defining a prostate shape as a set of 10 axial equally spaced slices from the base to the apex with 40 points each. Therefore, the number of points \( k \) that define each contour is 40 and the size \( m \) of the shape vectors \( s \) is the product between the number of slices and \( k \), 400. In figure 4.2 is represented this sampling approach.

![Figure 4.2: SSM shape parametrization used in \cite{44}: on the left, the axial slice positioning; on the right, the sampling of each axial slice.](image)

If the model defines a shape with this parametrization, the images to be segmented will also follow it. This implies that the acquired images should be approximately a set of 10 axial slices from base to apex with equal spacing. If the segmentation process initiates with the slice closest to the apex, the SSM model will superimpose the mean contour that was sampled in this precise area. By acknowledging the contour point set of the new prostate in this specific slice, the covariance matrix will estimate which will be the set of contours that define the remaining 9 slices, hence the global shape \( s \). This shape estimation is repeated throughout the whole process: the shape that supplies a prior contour to a given slice is calculated using the previously segmented ones.

Currently, the algorithm performs the initialization in a slice around the mid height of the prostate and finalizes in the apex and base areas. In each slice segmentation, the user can always adjust the calculated contours with control points, explaining why this algorithm is semi-automatic.

### 4.3 The shape parametrization problem

By understanding the role of the SSM in the ASM segmentation process, it is possible to conclude that the shape parametrization used is dependent on the geometrical parameters in which the target images will be acquired. This implies that a SSM based in 2D slices is prepared for a specific methodology of image acquisition. For example, the SSM parametrization used in \cite{44} represented in figure 4.2 assumes that the images will be acquired during a brachytherapy procedure: in this case, the prostate is visualized in axial slices using a side-fire probe. However, the geometrical parametrization of the images obtained by the system developed in this work is absolutely different since each volume
acquired by the 4DEC9-5/10 end-fire probe is composed by a set of oblique slices with equal angular spacing. Therefore, in order to have the segmentation hence registration software tool working properly with the current system, a different SSM parametrization must be adopted.

\[ x_p = \tan(k\theta)y_p, \quad k \in \left\{ -\frac{f_r-1}{2}, -\frac{f_r-1}{2} + 1, \ldots, \frac{f_r-1}{2}, \frac{f_r-1}{2} + 1 \right\} \]  

(4.5)

Here, all imaging planes are parallel to the \( z_p \) axis and have a slope in the \( x_pOy_p \) plane proportional to \( k \). The range of \( k \) values is defined in a way that the plane inclinations are measured relative to the \( x_p \) axis: there is a central value 0 for the center frame and two symmetrical sets of inclinations whose absolute maximums coincide with the field of view edges. For the example of the theoretical situation visualized in figure 4.3, where \( f_r \) is equal to 7, \( k \) is defined as the integers in the interval \([-3, 3]\). It is important to note that in this definition, the transitional acquisition between positions inherent to the automatic mode imaging is not being taken into account for purposes of simplification.
Considering the prostate as a surface in this cartesian space, the sampled contours are defined as its intersection with the set of equations 4.5. In image 4.3, the prostate surface is perfectly bisected by the central imaging plane and a set of approximately circular contours is visualized. If some translation occurs whether in the prostate or the transducer, both the position vector $CM$ and the intersection of the imaging planes with the prostate will change, generating new contours. This poses a problem to the ASM segmentation: since the algorithm uses prior knowledge based in 2D slices, a movement that leads to a different parametrization than the assumed by the SSM can influence directly the segmentation results. In order to minimize this unwanted effect, a strategy for probe orientation adjustment was developed.

4.4 Probe orientation adjustment Strategy

After completing the tool calibration process of 3.3, it is allowed not only to measure the kinematic data of the probe wobbling center but also to command movements in this new end-effector. Since it is required a match between the SSM shape parametrization chosen and the acquired images, the transducer wobbling center can only be moved with certain restrictions. Without any constraint, the probe can be moved in six distinct DOF, ($t_x$, $t_y$, $t_z$, $R_x$, $R_y$, $R_z$). Assuming that the SSM is prepared for the acquisition situation presented in figure 4.3, it can be predicted how the intersection between the imaging planes and the prostate will be affected by movements in each DOF. It is of utmost importance to refer that the prostate translation phenomenon due to patient movements or anesthesia solution pressure was not taken into account in this prediction.

The effect of performing any of the three translations is easy to predict: if the $CM$ distance vector changes, the intersection of the equations 4.5 will output different contours than the ones in the reference figure 4.3. For this reason, the movement strategy just considers rotations. In the case of the rotation $R_y$ around the $y_p$ axis, the alignment between the slices and the prostate does not suffer any change. The expected modification in the sampling can be perceived as a slight rotation of the prostate around the wobbling center, implying that the acquired contours will be approximately equal to the reference ones. In image 4.4, this effect can be observed: for $R_y$ positive, the probe perceives a prostate negative rotation around the $y_p$ axis; the opposite effect occurs with $R_y$ negative. The unexpectedly significant changes in the visualized contour intersections can be explained by errors in the CAD modeling.
Figure 4.4: Graphical description of oblique slice prostate sampling with transducer rotation around $y_p$ axis: in the left, a rotation with positive orientation; in the right, a rotation with negative orientation.

As for the $R_z$ rotation around the $z_p$ axis, a considerable modification is induced in the oblique sampling. Since the wobbler also rotates around this axis, the inclinations of the imaging planes will be directly affected: if the rotation is positive, the $k\theta$ values of equation 4.5 are incremented; otherwise they are decremented. The effects of such movement in the sampling are depicted in figure 4.5.

Figure 4.5: Graphical description of oblique slice prostate sampling with transducer rotation around $z_p$ axis: in the left, a rotation with negative orientation; in the right, a rotation with positive orientation.
In this case, it is possible to see that the prostate contours are well preserved but with one restriction: the amplitude of the rotation must be an integer around a multiple of the angular spacing between slices, Deg/Frame. In the theoretical cases of the figures in this section, the angular resolution is equal 10°. Since the amplitude of the rotations imposed in figure 4.5 is twice this resolution, 20°, it is expected not a change in the sampled contours towards the reference, but in the planes in which they are captured: for example, in case of positive rotation the contour present in the initial Reference center frame of figure 4.3 is now sampled by the plane with inclination 20°. This effect propagates in all the slices and can be perceived as a rotation of the prostate around the \( z_p \) axis.

**Figure 4.6:** Graphical representation of oblique slice prostate sampling with transducer rotation around \( x_p \) axis.

In terms of rotations \( R_x \) around the \( x_p \) axis, preservation of the reference contours is not granted. Due to the fact that \( x_t \) is indeed the probe longitudinal axis, a rotation in this direction will modify the wobbler alignment towards the initial Reference center frame which induces a new inclination term in the imaging planes equations. As it can be seen in figure 4.6, the obtained sampling is quite different than the reference of figure 4.3.

With this prediction information, it can be devised a orientation adjustment strategy in which all the acquired volumes will approximately follow a given shape parametrization. At the beginning, before starting any movement, a volume must be collected in order to define the reference parametrization. This initialization would involve a new SSM training set surfaces sampling that takes into account the \( CM \) vector and the oblique slice equations, generating a shape prior directed for the current biopsy. From this stage onward, the only probe orientation adjustments allowed would be two: slight rotations around the \( z_p \) axis with an multiple amplitude of the Deg/Frame parameter value used during the procedure; slight rotations around the \( y_p \) axis. This way, since the 3D TRUS acquisition system
synchronizes the KUKA-LWR end-effector position with the acquired images, it is possible to calculate
the transformations that the reference SSM contours will require in order to fit the new volume data.

4.5 Robotic probe movement Implementation

In order to perform the movements planned in the previous section, a set of new routines based
in the Reflexxes Motion libraries was implemented in the FRI host C++ program used in section 3.2.
With these functions it is possible to set the robot end-effector hence tool reference point to a target
translation \( t_f \) and orientation \( R_f \) with a given control methodology: in this case, Cartesian Impedance
Control [48] was used. To perform such movement, they use a cubic interpolation to generate a
trajectory between the current position data and the target one. In the case of this work, since only
two distinct rotations are allowed the target \( t_f \) was always equal to the measured tool translation,
guaranteeing that the wobbler center remains static during the procedure. To induce the rotations,
the input \( R_f \) was the product between the current tool rotation and one from the two expressions in
equation 4.6

\[
R_z(\theta_z) = \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad R_y(\theta_y) = \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix}
\] (4.6)

With the purpose of creating a manual method that allows the user to command the robot to
perform these rotations, a function based in keyboard keys was implemented in the FRI program. In
this program, four specific events were assigned to four different keys: two keys for each rotation, one
for the positive orientation and the other for the negative. The list of keys and graphical representation
of their induced movements is depicted in figure 4.7 Here, it is also shown the amplitude values \( \theta_z \) and \( \theta_y \) of the rotations. These values are always a multiple of a predefined angular resolution:
since the Deg/Frame value used to evaluate the system was 1.4634°, the angular resolution of \( \theta_z \) was
rounded up to 1.5°; for \( \theta_y \) the value 1° was used as there are no special requirements to be fulfilled.
The multiplier constant \( c \) is an integer value defined by the user.
This movement feature routine is only operational when the TCP/IP connection procedure implemented in 3.2 is completed. At this point, the FRI program is ready for both trigger detection and tool rotation events. To initiate the tool rotation event, one of four keys must be pressed in the command line. After, a message shows up asking for an amount of steps to define the amplitude of the rotation: this is where the user inputs the value \( c \) that calculates the rotation amplitude. Finally, the resulting value is displayed and the robot performs the rotation following its planned trajectory.
Final Application and System Evaluation
In this chapter is presented the final product of this work that is an interface developed in the host computer which uses all the processes and communications implemented in the chapters 2 and 3. Also, a set experimental volume acquisition trials and their results is presented in order to discuss the performance of the system. These trials take into account the proposed probe orientation strategy described in chapter 4.

The final application GUI is presented in section 5.1 along with its detailed operation protocol. In section 5.2 is presented the volume acquisition trial procedures, their results and discussion.

## 5.1 Final Application GUI

In the final part of this work a GUI was built to command the 3D TRUS data acquisition system from the host computer with MATLAB. In figure 5.1 it is a shown a screenshot of this interface with a set of numbers that mark procedural areas of the implemented synchronized images and robot position data acquisition.

The area (1) contains the first buttons to be used and is dedicated to establishment of communication between the three subsystems. The *Connect to Ultrasonix/Propello* button initiates the process of TCP/IP connection described in section 2.4.1 when pressed: the IP of the SonixTABLET must be input in the *Ultrasonix IP* edit text and the connection confirmation must be performed directly in the *Propello* GUI. The *Connect to CHAI3D/KUKA LWR* starts the TCP/IP connection described in section 3.2.2 as in the previous button, FRI host IP must be input in the respective edit text and the connection confirmation visualized in the FRI command line. Finally, the button *Initialize NI USB-6008* prepares the data acquisition board to send signals through the ao0 analog channel. In order to shutdown all these three connections, the button *Shutdown connections* must be pressed.

The initialization of *Propello* triggered image acquisition is done in the buttons of area (2). By pressing *Initialize Propello* the initialization process based in the 'detect' string of subsection 2.4.1 is executed. Once completed, the caption of button *Detected 3D transducer* is set to the name of the transducer used. At this point, in the *Propello* GUI the automatic mode imaging parameters must be selected and the *trigger* in parameter set to 2. To assure the correct wobbler position initialization the button *Change motor status* is used to perform the 'switchblock' trick initialization of subsection 2.4.1. The caption of button *Motor status* displays the block status of the wobbler during the procedure.

Area (3) is reserved to calibration of the robot reference position. When the FRI host program is initialized, the KUKA LWR enters in Gravity Compensation control mode which allows the user to adjust its position manually without exerting high forces. This way, the probe can be set to an intended reference 3D position which is measured and stored with button *Set Reference Position*. The result is displayed in the *Position/meters and Orientation/Degrees* boxes. For the orientation, the measured rotation matrix is converted to Yaw ($R_z$), Pitch ($R_y$) and Roll ($R_x$) angles.
Figure 5.1: Final Application GUI
In area (4) is performed the acquisition of synchronized data: by pressing the button *Acquire volume* the acquisition routine depicted in the flowchart of figure 5.2 is ran.

**Figure 5.2:** 3D TRUS acquisition system flowchart of volume covering: green blocks are operations in the C++ environment of Propello; blue are operations in the host computer MATLAB; red are operations in the C++ environment of the FRI host program; orange refers to initialization and termination processes; dashed lines represent TCP/IP communications; dotted lines represent electrical communication.

This flowchart presents only one modification towards the optimized triggered routine of figure 2.11.
the triggering of the FRI host and subsequent position data acquisition. Whenever a trigger is sent for image acquisition, a rotation matrix and translation vector are also obtained from the FRI host. The button Acquire single image serves only the purpose of testing the functionality of the system: it retrieves a single image and one robot position when pressed. Regardless the acquisition button pressed, the acquired images are displayed successively in area (5).

After performing the synchronized acquisition, the resulting data can be managed and observed in area (6). The data output by the acquisition routine is composed of three structures: an image array of dimension \([436 \times 480 \times fr]\), where \(fr\) is Frames/Volume; a vector of \(fr\) angular positions; a robot position data matrix \([3 \times 4]\). The entries of the position vector are calculated with the equations 2.4 and synchronized with the image array. By managing the buttons Previous and Next the user can visualize each image acquired separately in the area (5) and the corresponding wobbler angle in Image angle. The position matrix contains the mean of all the measured rotations and translations of the probe wobbling center during the acquisition. This data is displayed graphically the same way as the reference data position. The button Store Data allows to create a structure in MATLAB with all these three data sets and the reference position.

In order to acquire volumes with different probe positions, the first volume acquisition must be made with the probe in the reference position. After storing this reference volume data, new movements can be induced with the implemented key commands in the FRI host. This way, the next volumes will be acquired with a new position data relative to the reference one. This explains why it is pointed the subtraction of the current position data to an initial one in the overall system diagram of figure 1.10.

5.2 Evaluation Trials

After building the final acquisition application a series of experimental trials was conducted to both confirm the well functioning of the system and test the predictions formulated in section 4.4. These trials consisted on the ultrasound volumes acquisition of a prostate phantom using different controlled robot positions. Photographs of the experimental setup which include the KUKA LWR, the 4DEC9-5/10 probe, the SonixTABLET and the prostate phantom are shown in figure 5.3.

As recommended in the previous section, before performing any of these trials a reference volume was acquired. To move the probe to the corresponding reference position, the robot was pushed manually in Gravity Compensation control mode. In this reference the probe is inserted in the prostate phantom in a configuration similar to the one of figure 4.3 but with one important difference: the probe is rotated 180° along its longitudinal axis. In fact, this configuration is the one shown in the setup image 5.3. At this point the volume was acquired with the following parameters: \(\text{Deg/Frame}=1.4634°\); \(\text{Frames/Volume}=51\); \(\text{FOV}=74.6339°\). The decision of using the largest FOV value possible with a quite low resolution is supported by the fact that the whole prostate must be always perceived by the system regardless of the probe orientation.

After setting the reference, four volumes were acquired with the same parameters in four new
distinct probe orientations. Each movement was induced using one of the keyboard keys commands implemented in the FRI C++ program. In Table 5.1 is presented data relative to each of these four movements. In the first part is listed the keys used, the input amplitude multiplier \( c \) and the expected rotation amplitude. In the second part is listed the obtained real position data variations \( \delta t \) and \( \delta R \) relative to the reference.

**Table 5.1**: Probe orientation adjustment data for each experimental trial; the columns of Planned refer to the input movement; the columns of Real Measured Data refer to the position data that was measured after.

<table>
<thead>
<tr>
<th>Trials</th>
<th>Planned</th>
<th>Real Measured Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Key</td>
<td>( c )</td>
</tr>
<tr>
<td>1</td>
<td>'4'</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>'6'</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>'8'</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>'2'</td>
<td>15</td>
</tr>
</tbody>
</table>

By analyzing this table it is possible to conclude that the planned movements are not perfectly completed as the rotation variations differ considerably from the expected and the translation has variations of 1 mm. Such imperfection can be explained by the lack of positioning precision associated
to the Cartesian Impedance Control methodology. Still, this difference does not pose a problem to the acquisition system since the position data in which it relies is the real measured one and not the expected.

After acquiring the four volumes, their images were analyzed taking into account the predictions of section 4.4. According to them, by knowing the position data variation of the wobbling center it is possible to make a rough estimation on the geometrical differences that each set of contours will suffer relative to the reference set. To get evidence on this matter, a slice matching between each volume and the reference was executed: after observing a specific set of consecutive reference images and their prostate phantom slices, it was done a search within each volume for the image sequence whose slices best matched them. This analysis is only valid since the variations obtained in the DOF that do not preserve the reference contours \((t_x, t_y, t_z, R_z)\) were not significant. Here, the matching is presented using a set of three images from the reference volume with angles 19.0242°, 17.5608° and 16.9074°.

In figures 5.4, 5.5, 5.6 and 5.7 is depicted the slice matching for the volumes acquired in each of the four performed trials. In the top row is shown the reference image sequence and in the bottom row the corresponding match from the new volume.

Figure 5.4: Slice matching between reference images and set of images acquired in first experimental trial. The transformation between these volumes is defined by rotations \(\delta R_y=0.6207°\) and \(\delta R_z=-4.6409°\).

For the first trial in which the movement was induced by the ’4’ key, the images that match the reference are captured at a lower angular position 5.4. This effect of imaging plane switch is predicted by the situation modeled in figure 4.5 for rotations \(\delta R_z\): the prostate phantom was rotated to an area closer to the field of view edge and the contours are approximately preserved. Still, the angular offset of the switch was different from expected: since the measured rotation \(\delta R_z\) was a value around the
triple of the resolution 1.4634°, the slice matching would occur with a "jump" of three images. In this case, the match was verified with a "jump" of five images, implying that the amplitude of the rotation perceived by the phantom may have been closer to five times the resolution.

**Figure 5.5:** Slice matching between set of reference images and set of images acquired in second experimental trial. The transformation between these image sets is defined by rotations $\delta R_y^x = -0.7341°$ and $\delta R_z^x = 4.8240°$.

The second trial is similar to the first as the key '6' produces also a rotation around the $z_p$ axis but with a different sign. In its match 5.5, the imaging plane switch is also observed but in the opposite direction: the prostate phantom is perceived closer to the center and has also the contours approximately preserved. Regarding the angular offset, although the rotation $\delta R_z^x$ measured was also around the triple of the angular resolution and an image "jump" of three images was expected, the match was verified with a "jump" of two.

In the third trial a positive rotation around the $y_p$ axis with the key '8' was intended. The resulting match 5.6 shows two volume geometrical alterations: an imaging plane "jump" of one image and a small positive rotation of the contour center of mass around the wobbling center. This second effect is expected by the model of figure 4.4 for rotations $\delta R_y^x$: the prostate contours are approximately preserved but their position in each image suffers a slight rotational displacement. The first alteration is unexpected since the rotation $\delta R_z^x$ had a value half of the wobbling resolution.
Figure 5.6: Slice matching between reference images and set of images acquired in third experimental trial. The transformation between these image sets is defined by rotations $\delta R_y = 3.751^\circ$ and $\delta R_z = 0.7521^\circ$.

Figure 5.7: Slice matching between reference images and set of images acquired in fourth experimental trial. The transformation between these image sets is defined by rotations $\delta R_y = -13.3897^\circ$ and $\delta R_z = -4.6985^\circ$.

In the fourth and final trial the key '2' was used to impose a rotation in the same direction as the third trial but with different sign. In its image matching present in figure 5.7 it is also observed an imaging plane switch and a contour center of mass rotation. Regarding the switch, the results agreed
with the prediction as a jump of three images was verified after obtaining a rotation amplitude $\delta R_z$ around the resolution triple. The contour rotation in the images also followed the predictions: the prostate phantom slices are much closer to the center of the image due to a pronounced rotation $\delta R_y$ of $-13.3897^\circ$.

5.2.1 Trial results discussion

Overall, the geometrical transformations that a probe rotation induces in the acquired images can be reasonably well predicted. In case of rotations around the $z_p$ axis an imaging plane switch occurs whereas in case of rotations around the $y_p$ axis the position of the slices within each image changes.

In these trials some incoherence was verified in the amplitude of the imaging plane switch effect since most of the verified image “jumps” were different from the expected. This fact might be explained by several factors associated to the experimental setup. Firstly, the prostate phantom might have been moved slightly between acquisitions. Secondly, the robot tool reference point was not perfectly coincident with the probe wobbling center as the calibration performed in section 3.3 had an error of $1.8\, \text{mm}$. This error is of major relevance since the predictions assume movements around the wobbling center.

Also, it can be concluded that using a value near to $\text{Deg/Frames}$ in the rotation $R_z$ planning is not essential as the executed movements are not perfect and the system measures always the real obtained position. Furthermore, even if these multiples are not obtained, the use of a high resolution guarantees that the prostate slices do not change considerably between two consecutive slices.

These trials also served to confirm the proper functioning of the implemented acquisition system. The geometrical transformation analysis was only possible since the implemented system acquired reasonably well synchronized data from the KUKA LWR and Propello hence 4DEC9-5/10 transducer.
6

Conclusions and Future Work

Contents

6.1 Conclusions .................................................. 70
6.2 Future Work ............................................... 71
6.1 Conclusions

The main objective of this work was fulfilled, the implementation of a 3D TRUS acquisition system that performs synchronous acquisition of ultrasound images and kinematic data of the corresponding ultrasound transducer.

A great focus was given to the Propello software that controls the 3D integrated transducer 4DEC9-5/10. Such is explained by the fact that currently these softwares are not commonly used and information about them is scarce. The Propello capabilities were successfully explored, and it is thought that the routines developed are optimal in terms of efficiency. From the two acquisition routines developed, the one used in the final setup was the automatic triggered due to its high image acquisition frequency (14.676 fps). Still, if the incompatibility problem of manual mode with portaRunImage() is solved in a future version of Porta SDK, the optimal manual routine would best fit the final system. With this routine it would be possible to use all the mechanical range of the transducer wobbler freely instead of restricting its movements to the parametrizations of table 2.2.

The overall acquisition frequency of the system is around the 14.476 Hz value as the TCP/IP communication with KUKA LWR does not present a considerable time expense increment. During the work, it was given a major importance to minimization of the time expense of the system not only because it concerns a real time application, but also because of the time expense that the real time semi-automatic registration algorithm will introduce. The task improving even more this frequency proved to be difficult since the speed of the TCP/IP socket communication can not be raised. Moreover, the electrical impulse generation with the NI USB 6008 boards also has a unchangeable time expense.

The performed trials served to test the functioning of the system and the probe orientation strategy that takes into account the requirements of a semi-automatic ASM segmentation. The predictions made in terms of preservation of the prostate contours visualized in a reference volume are prone to error as the probe movements are never perfect. Still, sampling the prostate with a high number of images might reduce this problem since in the trials with 51 image volumes it was possible to find very similar sets of slices and estimate the approximate transformation that the prostate sampling suffered. If these transformation estimations are obtained in real time, segmentation can be achieved by applying this transformation to a chosen SSM based in the reference sampling. This means that to link the registration tool of [44] to the developed system the implementation of a new SSM is required. It is important to note that the prostate translation phenomenon during the procedure is not taken into account in this strategy.

The trial results were probably influenced by the error of 1.8 mm in the calibration of the robot end-effector to the wobbling center point. Even though this process could be repeated in order to get a smaller error, measuring a point that is not in the surface of the probe but inside the transducer housing is a challenging task.

Although this system was developed with the purpose of being integrated in a robotized prostate biopsy system, the synchronous data acquisition from a 3D integrated transducer and a robotic arm
is a feature that could be of great utility for other medical applications, namely the real time reconstruction of medical images. Such is possible since the Propello software is prepared to control other transducers and all the routines implemented can have their wobbling parameters modified accordingly. For the robot, only the mechanical coupling process would have to be repeated.

6.2 Future Work

The proposed future work follows the implementation of the robotized biopsy system. Firstly, in order to perform MRI/3D TRUS registration with the data acquired by the developed system, further work should emphasize in the adaptation of the segmentation tools. Although this work considered strictly the case of registration based in a slicewise segmentation, options of direct volume segmentation could also be considered. Still, this approach may introduce a much larger computational expense [53].

Once the registration software is operational a robot targeting strategy could be implemented. Such algorithm would perform probe orientation adjustments while having as feedback the prostate MRI/3D TRUS data fusion registration results.

In order to have any of these processes working, a process of needle guide and ultrasound probe calibration such as the one proposed in [42] must be also performed. The objective of such procedure is to find the coordinate transformation between the space of the ultrasound images and the position of the robot hence probe and needle guide. This way, it is possible to calculate accurately distances between the probe and specific points visualized in the ultrasound images. Only through such feature the robot can calculate with precision the target to which the needle guide should point.
Bibliography


[41] A. Pelletier, F. Aoun, F. El-Khoury, E. Hawaux, K. Limani, K. Narahari, N. Sirtaine, and R. van Velthoven, “3D versus 2D Systematic Transrectal Ultrasound-Guided Prostate Biopsy:


