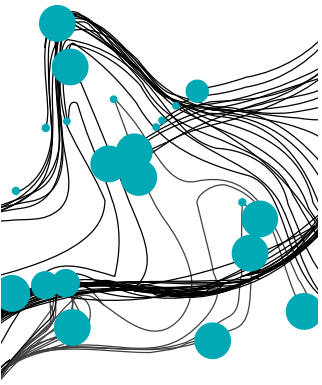
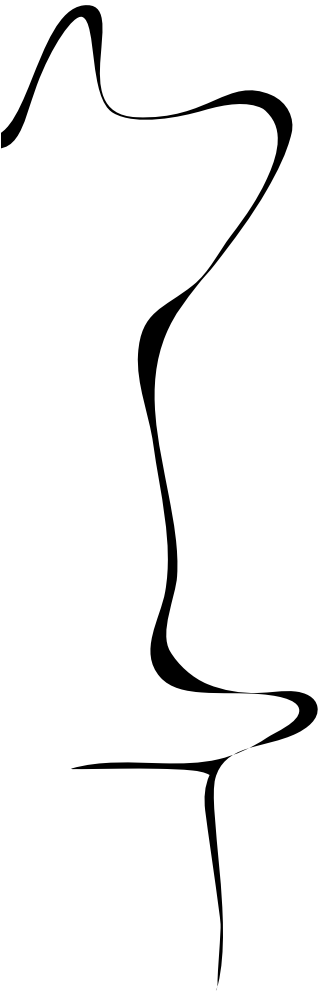


RAM ROBOTICS AND MECHATRONICS



MRI-GUIDED ENDOVASCULAR INTERVENTIONS WITH CATHBOT ROBOTIC PLATFORM

M. (Martin) Reinok

MSC ASSIGNMENT

Committee:
dr.ir. M. Abayazid
dr. G. Dagnino
dr. ir. W.M. Brink
dr. N. Strisciuglio

July, 2024

RaM2024
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE
UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

Endovascular procedures often rely on X-ray fluoroscopy for real-time imaging guidance, which poses risks due to ionizing radiation exposure. MRI offers a safer alternative with superior soft tissue contrast and avoids using ionizing radiation. Still, its adoption for procedure guidance has been limited by technical challenges. This thesis addresses key challenges in MRI-guided endovascular interventions by developing a proof-of-concept solution using the CathBot robotic platform and interactive MR imaging. The primary issues addressed include low spatial/temporal resolution, guidewire visibility, and procedure safety.

The project aims to enhance MRI-guided procedures by optimizing an interactive MRI sequence, detecting an MR-safe passive endovascular guidewire using a convolutional neural network (CNN), and implementing haptic feedback on the CathBot platform. The MRI sequence optimization compares balanced steady-state free precession (bSSFP) and gradient echo (GRE) sequences to select the optimal sequence for signal-to-noise and contrast-to-noise ratios while maintaining interactive capabilities. The CNN-based detection method was developed to identify susceptibility artefacts created by paramagnetic markers on the guidewire in near real-time. Additionally, an automated MRI slice alignment system was created to track the guidewire's movement. The CathBot platform was integrated with the detection system, providing the operator with real-time collision detection and haptic feedback. This integration provides a concept for introducing haptic feedback for MRI-guided interventions, ensuring improved accuracy and reduced risk during procedures.

Contents

1 Introduction

Diseases of the circulatory system continue to be the leading cause of death in the world, accounting for 35 % of the deaths in Europe in 2022, followed up by cancer at 26% [1]. This motivates the exploration of alternative medical interventions to open surgery of the circulatory system, which has higher odds of mortality and complications. A viable alternative has been endovascular intervention, whose primary benefit is faster recovery due to the minimally invasive approach.

Endovascular procedures typically rely on fluoroscopy, which involves using X-rays - a form of ionizing radiation that increases the risk of cancer. Despite this risk, X-ray imaging is favored for its real-time imaging capabilities, lower cost, and ease of physical patient access during interventions [2]. However, X-rays have limited capability for visualizing soft tissue (i.e., brain, muscles, and organs), which might reduce the accuracy of therapy delivery to the targets [2].

Magnetic Resonance Imaging (MRI) offers an alternative to X-ray fluoroscopy, providing high-quality imaging without ionizing radiation. MRI provides excellent soft-tissue contrast and three-dimensional imaging capabilities, although its frame rates are slower than X-ray. Numerous clinical and experimental studies have demonstrated the feasibility of MRI-guided endovascular procedures [2], [3].

Despite the benefits, MRI has several drawbacks that limit its application for interventional procedures. These include low spatio-temporal resolution, which refers to the level of detail in both space (spatial) and time (temporal) that the MR scanner captures, compared to X-ray. Additionally, there are difficulties with endovascular devices visibility as well as the confines of the MRI scanner itself, which might limit patient accessibility during active scan [4].

This thesis focuses on enhancing MRI-guided endovascular procedures by addressing several key challenges raised in the literature, such as spatial and temporal resolution, endovascular guidewire visibility, manual MRI slice alignment, and safety of the procedure [4]. An interactive MRI sequence is used, which provides sufficient update frequency and resolution for guidewire detection. Using the sequence, a near real-time guidewire detection and tracking software is built to move the MRI slice and keep the guidewire in the center of the image. The detected artifact data is visualized in a custom 3D suite, which shows the artifacts in 3D space relative to a scanned 3D model of the phantom and MRI slice position. The 3D space is used to detect potential collisions or safety concerns between vessels and guidewire, and this data is relayed to the CathBot robotic platform to introduce haptic feedback to make the operator aware of collisions with the blood vessels.

1.1 Medical Robotics

Medical robotics has a significant role in directing the future of surgery toward easily achievable precision, repeatability, and targeted therapy. The use of robotics in medical procedures enables earlier diagnosis, improved efficiency, and delivery of therapeutic measures, leading to better patient outlooks and extended survival rates. Consequently, clinicians face greater demands to handle complex, multidimensional information beyond anatomical details. Medical robotics can assist clinicians with procedures that require a high level of dexterity, precision, and reasoning to complete successfully; more common procedures also benefit from repeatability and adaptation to different patient anatomies.

The evolution of medical instrumentation began long ago, with the first endoscope used in 1806 [5]. However, most recent advancements in medical robots occurred in the late 20th and 21st century due to the improved electronics, computer science, and robotics. The earliest record of the use of medical robots dates to 1985 when an industrial robot was used to perform

brain biopsy using CT imaging and preprogrammed movements [6]. The widely known and highly regarded *Da Vinci® Surgical System* (Intuitive Surgical, Sunnyvale, CA, USA) emerged in 2000 and is to this date accepted to be one of the best and most used surgical systems for minimally invasive procedures [6; 7]. *Da Vinci* robot used to rely on visual cues for feedback, whereas artificial tactile haptic force, which allows the surgeon to feel forces generated at the distal end of instruments, has been implemented on the latest versions of the robotic system [8]. Clinical studies show that artificial haptic force has a significant impact on applied forces, completion time, accuracy, and success rates, especially for catheterization procedures [9]. Many medical robots already incorporate artificial feedback to provide these benefits to clinicians [10].

Sensei® (Figure 1.1) is an FDA-approved robotic catheter system originally designed by Hansen Medical Inc., USA (merged with Auris Medical Inc. in 2016. and acquired by Johnson & Johnson in 2019), which is designed for catheterization and endovascular procedures under X-ray. The *Sensei®* catheter's tip is controllable in three dimensions using a remote control, linked to a robotic navigation system that measures forces at the tip. These tactile sensations are transmitted to the user through the controller as haptic feedback [10]. While the robot is very promising for catheterization, its usage is limited to X-ray fluoroscopy.

Figure 1.1: Sensei X2 Robotic Catheter System [11]

Recent technological developments such as MR-safe Pneumatic Stepper Motors [12] also allow for MR-safe robotic catheter systems such as CathBot (Figure 1.2).

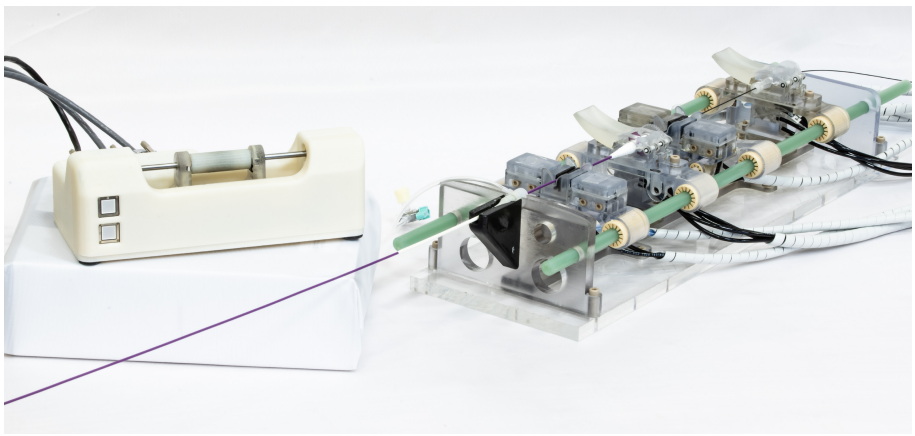


Figure 1.2: CathBot robotic platform with MR-safe robot for catheter and guidewire control (right), and remote controller (left) for endovascular intervention [13]

1.2 CathBot robot for endovascular procedures

The CathBot project [14] proposes an MR-safe teleoperation platform to manipulate endovascular devices remotely. The CathBot has an MR-safe robot intended to be placed in the MR scanner room and used to enable movements such as translation and rotation for catheter and guidewire. These transformations are made possible by multiple MR-safe pneumatic stepper motors [12]:

- Four linear movement motors
- Two rotational motors
- Two Clamps

Air is supplied to the motors by 28 tubes, around 5 meters in length. The tubes are connected to the pneumatic controller, which should reside outside the magnetic field in the MR control room.

The device movements are instructed by a custom Human-Machine Interface (HMI), known as the main controller, intended to be the interface for medical professionals. The main controller teleoperates the endovascular device's movements from outside the MRI room.

This thesis incorporates a feedback loop into CathBot, enabling operators of the main controller to experience haptic sensations during MRI-guided endovascular procedures. This feedback enhances the surgeon's tactile perception, improving the procedure's safety by increasing the friction felt on the CathBot main controller when the guidewire approaches a vessel wall.

1.3 Project Goal

This thesis aims to address several key challenges raised in literature about MR-based interventional procedures by developing a proof-of-concept solution with the CathBot robotic platform and interactive MR imaging. The MRI challenges for which practical solutions are proposed in this work are the following [4]:

1. Low spatial/temporal resolution
2. Guidewire visibility
3. Manual MRI slice alignment
4. Procedure Safety

To address these issues, this research aims to answer the following research question:

How to improve MRI-guided endovascular procedures?

The research question was approached with the following objectives:

- Optimize an interactive MRI sequence for real-time endovascular procedures.
- Detect an MR-safe passive endovascular guidewire during an interactive MRI sequence.
- Implement haptic feedback on the CathBot robotic platform

Interactive MRI sequence optimization is addressed in Section 3.2, by a series of MRI sequence comparisons and evaluations. Two MRI imaging sequences are compared: balanced steady-state free precession (bSSFP) and gradient echo (GRE), and the most suitable sequence with optimal signal-to-noise (SNR), contrast-to-noise (CNR), and sufficient interactive properties is chosen.

MR-safe guidewire detection is covered in Section 3.5 by training and using a CNN to detect the passive MRI-safe guidewire during interactive MRI sequence in near real-time.

Manual MRI slice alignment is discussed and a concept tracking system is implemented in Section 4.2. The MRI slice movement is automated to track the guidewire's movement.

Haptic feedback is implemented on the CathBot platform in Section 4.3, where collision detection between the guidewire and blood vessel is relayed to the CathBot main controller in the form of haptic feedback in near real-time.

2 Background and Resources

This chapter provides an overview of the foundational work and resources that support this project. It covers related work in the field, details of endovascular procedures, the CathBot robotic system, the basics of magnetic resonance imaging (MRI), the use of convolutional neural networks (CNNs) for image processing, and the specifics of interventional guidewires and susceptibility artefacts.

2.1 Related Work

2.1.1 Interactive MR sequences

Real-time MRI (RT-MRI) faces several trade-offs, primarily between spatial resolution, temporal resolution, and signal-to-noise ratio (SNR). Achieving high temporal resolution (faster imaging) often requires compromising on spatial resolution (image quality) and vice versa. Several works have improved the feasibility of RT-MRI by providing robust and consistent imaging techniques.

In this project, RT-MRI is combined with interactive MR imaging. The interactive imaging concept allows for "on-the-fly" changes to the MR scan plane, shim, and other imaging parameters.

K. S. Nayak et al. describe the feasibility of RT-MRI and the differences between the two fundamental sequences that enable real-time scanning: GRE and bSSFP [15]. bSSFP sensitivity to off-resonance is mentioned as a drawback, which manifests as banding artifacts. However, it is mentioned that bSSFP sequences provide higher SNR efficiency than GRE. The choice between bSSFP and GRE may also depend on the imaged anatomical region. For instance, bSSFP provides T_2/T_1 contrast, which is advantageous in cardiac imaging. Undesirable transient effects on the steady-state signal are mentioned, which can influence the MR signal during fast movement of the scan plane or in the case of flow or motion of the imaged regions. The movement during the bSSFP sequence should be limited and gradual to reduce transient effects.

Han Nijsink et al. compared passive marker visibility with a pulsatile flow phantom [16]. Their results concluded that the GRE was more suitable than bSSFP across the board for the following qualities: marker visibility, image quality, and usefulness. Temporal resolution was not considered. However, it is very important in this project, and therefore, similar tests to the paper were performed in Section 3.2, while taking real-time imaging into consideration.

2.1.2 Susceptibility artefact simulation

A lot of work has been documented about simulating susceptibility artefacts. The majority of this work focuses on gaining an understanding of the impact of small objects, such as implants or surgical clips, with different magnetic susceptibilities on MRI signal loss. In these cases, susceptibility artefacts are considered unwanted, and their effect is desired to be reversed or corrected. [17] and [18] provide the calculations to estimate the shape of different artefacts and identify the naturally occurring artefacts in the human body, such as air cavities or foreign bodies.

S. Patil's PhD work describes the simulation and field effects of the susceptibility artefacts and demonstrates an experimental tracking technique using echo-dephased SSFP sequences [19]. The echo-dephased SSFP sequence can separate the artefact from the background, at the cost of having to image two separate sequences, as the anatomy is not visible in the echo-dephased SSFP sequence.

2.1.3 Detecting susceptibility artefacts

Limited research has been done to detect or localize susceptibility artefacts from images using computational methods. Han Nijsink et al. trained a U-Net model with manually labeled data to predict marker location likelihood maps. They used a training dataset consisting of 30 labeled images and 10 images for validation. The results were promising, with the median number of markers detected from images equal to the number of markers in the images [16].

J. Bijlsma et. al. used template matching and blob-finding methods to detect artefacts from 2D MR images and concluded partial success [20]. That is because these methods require a lot of preliminary variables and static imaging conditions to be able to detect properly, and with MR, these conditions can not always be guaranteed. The slice thickness might have to be increased, which causes loss of CNR, and therefore the artefacts would no longer be detected.

2.1.4 MR slice tracking

MR slice tracking is not a novel concept and is already widely performed using active tracking techniques with active guidewire or catheter combinations. For example, using an active guidewire, A.K. George et al. have demonstrated a real-time tracking method called "SnapTo" [21]. This method allows the user to snap the MR slice to a chosen relevant section of the device, using a 3D curve plotted along the device.

Passive guidewire tracking is in active development both in research and in commercial products. As mentioned, S. Patil used an echo-dephased SSFP sequence to track passive guidewire [19], while J. Bijlsma [22] used template matching. Nano4Imaging GmbH (Düsseldorf, Germany) has "TRACKR AI" software under development, in collaboration with ADIS (Lausanne, CH) [23]. This software shows promising results for tracking a commercially available passive guidewire (Nano4Imaging SmartWire).

2.2 Endovascular Procedures

Definition and Scope
Endovascular surgery is a minimally invasive procedure used to diagnose and treat circulatory diseases. In this technique, vascular surgeons utilize catheters (long, thin, flexible tubes) and guidewires to navigate through arteries and veins to deliver treatments. The most common application of endovascular surgery is the treatment of abdominal aneurysms [24]. For this project, we use an abdominal aortic aneurysm phantom, shown in Figure 2.2, to conduct our experiments.

Endovascular surgery has some benefits over traditional, open surgery:

- Shorter recovery time than traditional surgery
- Can be performed with local anesthesia instead of general
- Less pain and discomfort during recovery
- Smaller incisions result in less bleeding

An example of endovascular procedure to treat an abdominal aneurysm is as follows:

1. A thin, flexible guidewire is inserted into the femoral arteries (usually in the groin) with a small surgical incision.
2. Guidewire is navigated up to the aneurysm
3. Once the guidewire is properly positioned, a catheter is inserted over the guidewire and advanced to the aneurysm.
4. Once the catheter is in place, the guidewire is removed, and any treatments, e.g., in the case of the abdominal aneurysm, the stent graft is carried to the aneurysm through the catheter, as illustrated in Figure 2.1.

This project's scope is limited to guidewire navigation, tracking, and positioning. Endovascular procedures as a whole are not undertaken.

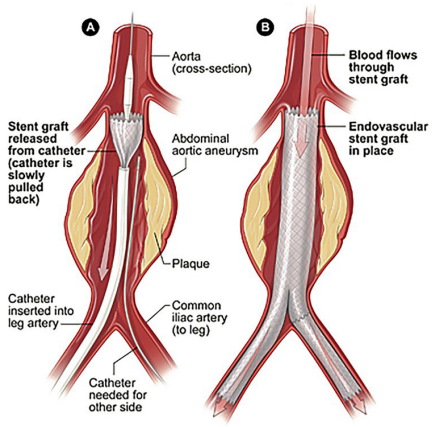


Figure 2.1: a) Abdominal aneurysm treatment using endovascular procedure (Endovascular Repair by National Institutes of Health. Public Domain.)

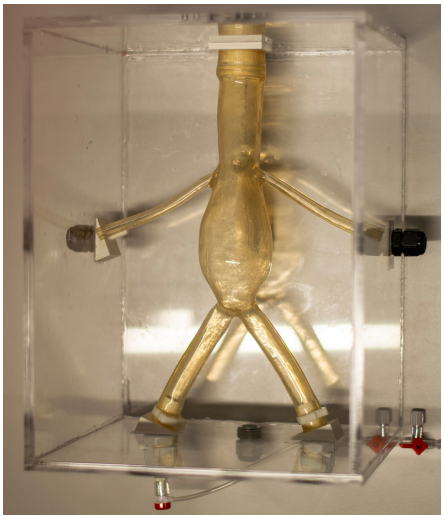
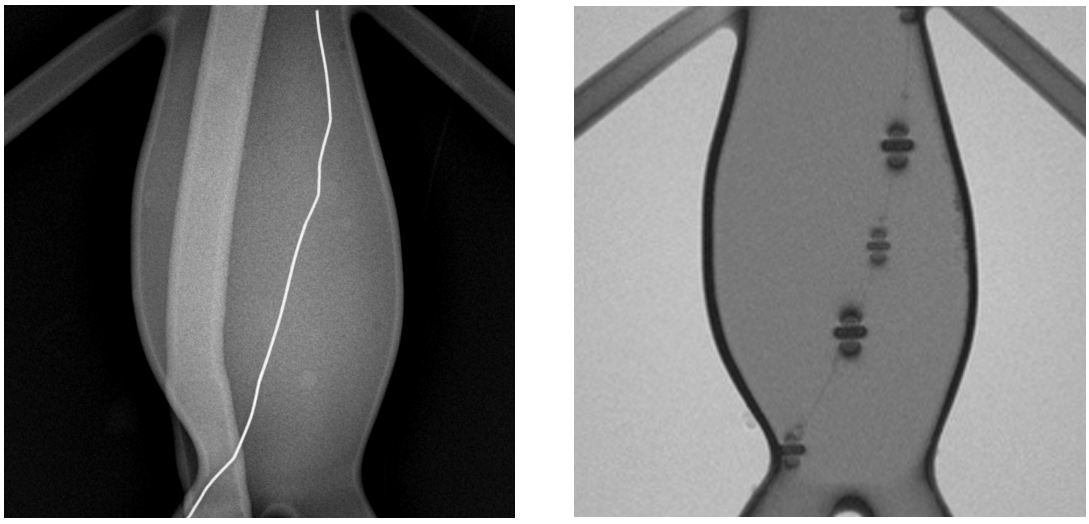


Figure 2.2: b) Aneurysm phantom at University of Twente (UT)

Different methods for Endovascular Intervention

Endovascular procedures often rely on X-ray fluoroscopy, mainly due to the commendable resolution and image update frequency it provides. Most guidewires are designed to be visible under X-ray, so navigation would be easier, as shown in Figure 2.3 a). This is not the case for MRI, as MRI is sensitive to hydrogen atoms, and metallic guidewires do not have any hydrogen in them, making them nearly invisible. Other workarounds are used for this purpose; in this project's scope, MR-safe passive guidewires with embedded markers are used, shown in Figure 2.3 b).



(a) **(b)**

Figure 2.3: Coronal images of UT aneurysm silicone phantom comparing endovascular device visibility under a) X-ray fluoroscopy and b) MRI.

2.3 CathBot

Definition and Scope

CathBot is a versatile robotic platform for fluoroscopy and MRI-guided endovascular interventions [14]. It is an early-stage robotic prototype that has been successfully evaluated in an MR environment and demonstrated task feasibility in a preliminary study [13].

CathBot robotic framework consists of 3 components:

1. **MR-safe robot**: Used for endovascular devices manipulation.
2. **Main controller**: Sends control and teleoperation commands of the endovascular devices to the pneumatic controller.
3. **Pneumatic controller**: Valve controller which uses air to control pneumatic actuators on the MR-safe robot.

From the components, only the MR-safe robot is intended to be in the MR room. The robot uses pneumatic actuators to provide 2 degrees of freedom (DOF) for both, guidewire and catheter. That is, the devices can be advanced forwards or backward and rotated clockwise or counter-clockwise. Figure 2.4 shows the components of the robot. It is designed to be nearly completely 3D-printable, besides some plastic off-the-shelf components. This makes the device very cheap to produce, and in practice, it should be disposed of after a single use.

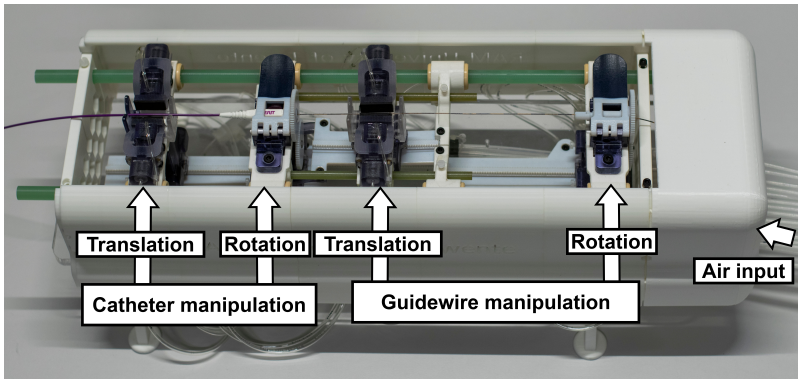


Figure 2.4: CathBot MR-safe robot with guidewire and catheter.

The pneumatic controller, shown in Figure 2.5, is an air-valve controller that generates pneumatic pressure patterns for the MR-safe robot motor control. The air pressure is transferred over 5m of tubes to the robot. The pneumatic controller is intended to be outside the MRI room, in the control room, preferably close to the main controller. The main controller, shown in Figure 2.6, is intended to be used by the endovascular clinician to send control commands to the pneumatic controller and thus remotely operate the endovascular devices from the control room outside of MRI.

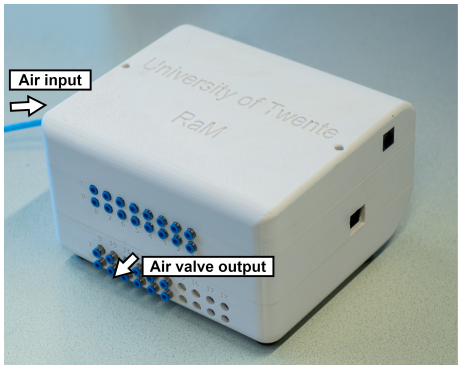


Figure 2.5: CathBot pneumatic controller.

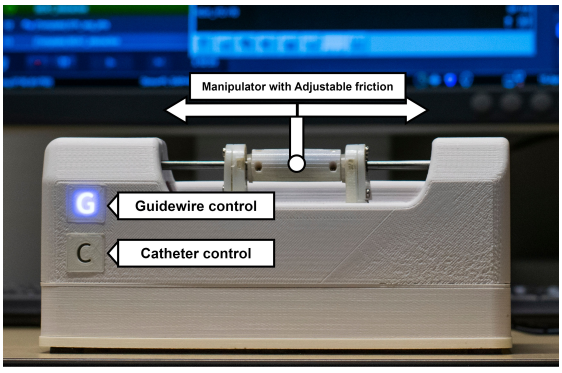


Figure 2.6: CathBot main controller.

This thesis will focus heavily on the main and pneumatic controller's framework, software, performance, and usability. The MR-safe robot is not subject to change in this project.

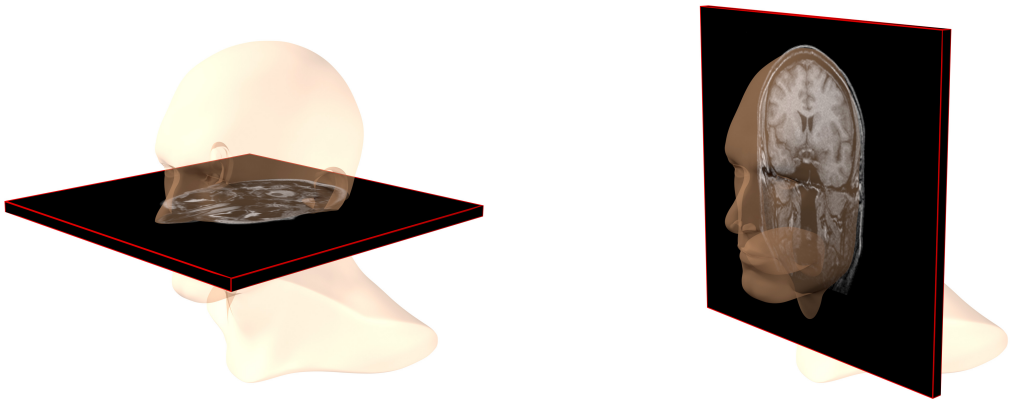
2.4 Magnetic Resonance Imaging

The technology and physics behind MR imaging are very complex, and this section describes a simplified working principle.

Definition and Scope

MR imaging is a non-invasive medical technique used in radiology to form pictures of the anatomy inside the body. MR scanners use strong magnetic fields, magnetic field gradients, and radio waves to rapidly change and record hydrogen atoms' magnetic alignment, which is detected by antennas and converted into images. Since hydrogen atoms are naturally abundant in humans and other biological organisms, particularly in water and fat, MRI is very effective at capturing images of these tissues. This subsequently means that objects and items with no hydrogen atoms do not create any signal under MR and, therefore, can not be imaged.

MR imaging offers full control over the imaging plane, which can be angulated and positioned at any location in the human body. Unlike traditional X-rays, which capture an object's two-dimensional projection, MRI allows gathering data from a specific two-dimensional slice, with a certain slice thickness. Figure 2.7 demonstrates coronal and axial slice placement with a width of 10mm. With MRI, the thickness and orientation of each slice can be chosen to match precisely the area of interest. During an acquisition, the MR system collects all the data within the slice and converts it to an image. The slice position and orientation can be changed in real-time during an interactive scanning procedure using specialized software such as Access-i (Section 4.1). This enables interactive control over the scan plane, and looking at the anatomy from a different angle or following the endovascular device with the MR slice.



(a) **(b)**

Figure 2.7: MR scanner slice placement example in a) Axial and b) Coronal orientation

MRI is considered harmless to the human body, compared to other imaging methods such as X-ray fluoroscopy and positron emission tomography (PET) scans, as it does not involve X-rays or the use of ionizing radiation. However, MRI is not hazard-free. The main safety hazard of the MR scanner is the strong static magnetic field B_0, which is orders of magnitudes larger than the magnetic field of earth and can torque, attract, and accelerate ferromagnetic objects towards the scanner [25]. This field is permanently enabled and can not be switched off, except in an emergency. This causes a significant risk for any metallic and electronic device near the scanner, including vital electronics such as pacemakers and defibrillators, which can malfunction or harm the patient [26].

MR scanning can also take a long time, which may cause some patients with claustrophobia to experience anxiety, although open-MRI systems are also available, which are slightly less powerful but do not enclose the patient inside the MRI [27].

Figure 2.8: MRI scanner at the UT, with the CathBot MR-safe robot and aneurysm phantom.

2.5 Convolutional Neural Networks

Definition and Scope

Deep learning (DL) is used in the domain of digital image processing to solve difficult problems such as image colorization, classification, segmentation, and detection. Performance of DL methods such as Convolutional Neural Networks (CNNs) has been well-established to outperform traditional algorithms, although CNNs have increased computing demands and require training [28].

CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input images in a computationally efficient way. CNNs use methods such as convolution and pooling to maximize a GPU's capability for orders of magnitude performance gain [29].

In this thesis, the nnU-Net architecture [30] is used and implemented for image segmentation purposes. nnU-Net is a semantic segmentation method that automatically adapts to a given dataset. It provides an out-of-box solution that analyzes provided training cases and automatically configures a matching U-Net-based segmentation pipeline. The user requires very little expertise and the output is very robust. It supports 2D and 3D images and understands medical-specific modalities such as voxel spacing and anisotropies [31].

2.6 Interventional guidewires and susceptibility artefacts

Definition and Scope

Interventional guidewires are medical devices used to navigate the vasculature. Guidewires come in different lengths, thicknesses, and shapes according to which vessels are navigated during the procedure. In fluoroscopy, the guidewires are usually made from nitinol (an alloy of nickel and titanium), stainless steel, or another metallic alloy. Nitinol allows more flexibility, memory (ability to maintain the original shape), and resistance to bending [32]. However, these standard guidewires can not be used under MRI as the radiofrequency (RF) induces heating of such guidewires, creating a significant risk for thermal injury [33].

Instead of using metallic alloys, MR-safe guidewires are made from composite non-metallic materials such as glass fiber or aramid fiber. This makes the guidewire nearly invisible under an MR scanner, but safe to use. Active or passive tracking can improve the visibility of inter-

ventional devices under MRI. Active tracking involves integrating a miniaturized RF receive coil into the guidewire, which produces a strong, localized intensity that is visible with dedicated tracking MR sequences. However, this requires additional electronic hardware and is more costly than passive devices [34]. Passive tracking uses small ferromagnetic markers embedded throughout the guidewire, as shown in Figure 2.9. These markers create localized B_0 field inhomogeneities known as susceptibility artefacts. Figure 2.3 b) demonstrates what susceptibility artefacts look like, using an MR-safe guidewire by EPflex (MRLine, EPflex, Dettingen an der Erms, Germany). The guidewire has specialized MRI markers placed throughout the guidewire, as shown in Figure 2.9.

In this thesis, the susceptibility artefact signal is simulated, and the simulated artefacts are used to train a CNN to detect these artefacts in near real-time during an interactive MRI sequence.

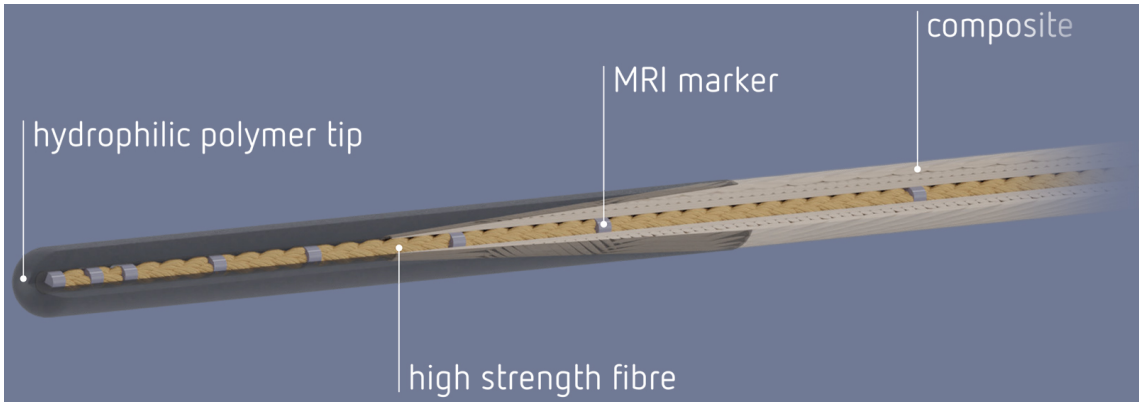


Figure 2.9: EPflex MRline MR-safe guidewire composition [35]

3 Simulation and Detection of Susceptibility artefacts

This chapter will cover the simulation and detection of susceptibility artefacts. A series of experiments was designed to evaluate and optimize artefact visibility during interactive MR sequences for simulation reference and optimal real-time detection. All MR data was acquired on a 1.5T MRI system (MAGNETOM Aera, Siemens Healthineers, Erlangen, Germany) using an 18-channel torso RF coil and a 24-channel spine RF coil.

MR sequence optimization is followed by the calculations and implementation for simulating susceptibility artefacts, and the generation of a CNN-compatible dataset for near real-time CNN detection of susceptibility artefacts.

3.1 Susceptibility artefacts under MRI

The paramagnetic markers embedded in the guidewire create small local magnetic field distortions. These distortions, called susceptibility artefacts, are intentional and make the MR-safe guidewire visible during imaging. The artefacts introduce a three-dimensional magnetic field distortion in the MR main magnetic field B_0, as visualized in Figure 3.1. This means the artefact has a defined, detectable shape when the MR slice is parallel with the B_0 field. When the slice is not parallel, the artefact shape changes. To simplify simulation, training, and detection in this proof-of-concept study, the MR plane will always be parallel with the B_0 field, and the appearance of the artefact on oblique slice orientations will not be considered.

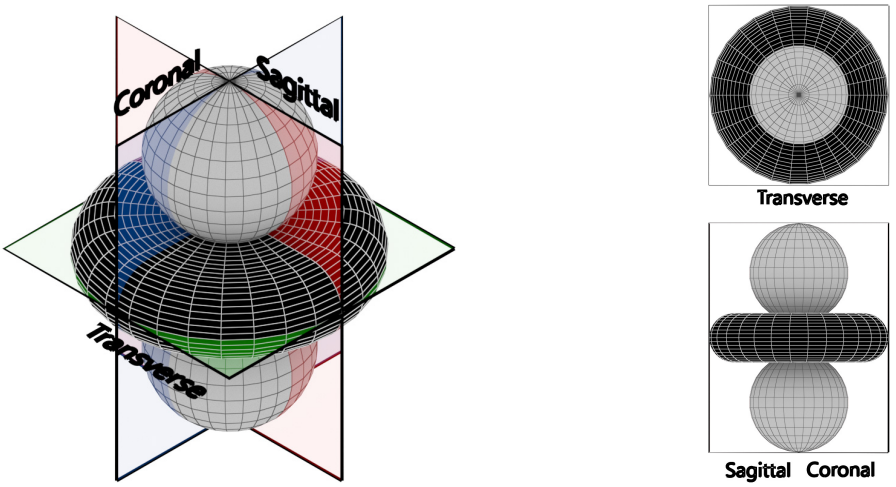


Figure 3.1: Susceptibility artefact three-dimensional shape in different MRI axis. a) isometric view, b) transverse (top) and sagittal/coronal view (bottom).

3.2 MRI sequence optimization

Before the susceptibility marker could be adequately simulated, a real-life dataset was acquired. The marker visuals and SNR can vary greatly with different sequence types, such as GRE or bSSFP. Within the chosen sequence, various parameters affect the appearance of the marker, such as slice thickness, repetition time, in-plane resolution, receiver bandwidth, and more. This section describes and presents our experiments to find the most suitable sequence type and sequence settings for interactive guidewire detection.

3.2.1 MRI sequence type

Following earlier studies [16], [36] and [37], both two-dimensional (2D) gradient-recalled-echo (GRE) and balanced steady-state free precession (bSSFP) sequences were evaluated. [16] suggested the use of GRE sequences instead of bSSFP as GRE does not have off-resonance banding artefacts and is less prone to flow artefacts and transient effects. However, bSSFP offers more effective SNR [15]. Figure 3.2 compares the two sequences with as similar as possible acquisition settings (in-plane resolution of 1.4 mm, slice thickness 10 mm). The figures show that the GRE sequence has significantly reduced SNR compared to the bSSFP acquisition, even though bSSFP has a 30% lower repetition time and is, therefore, more suitable for real-time and interactive procedures. The off-resonance banding artefacts, which inherently are part of the bSSFP sequence, do cause some trouble, but they are located at the edges of the MR image. In our particular setting, they do not interfere with the procedure.

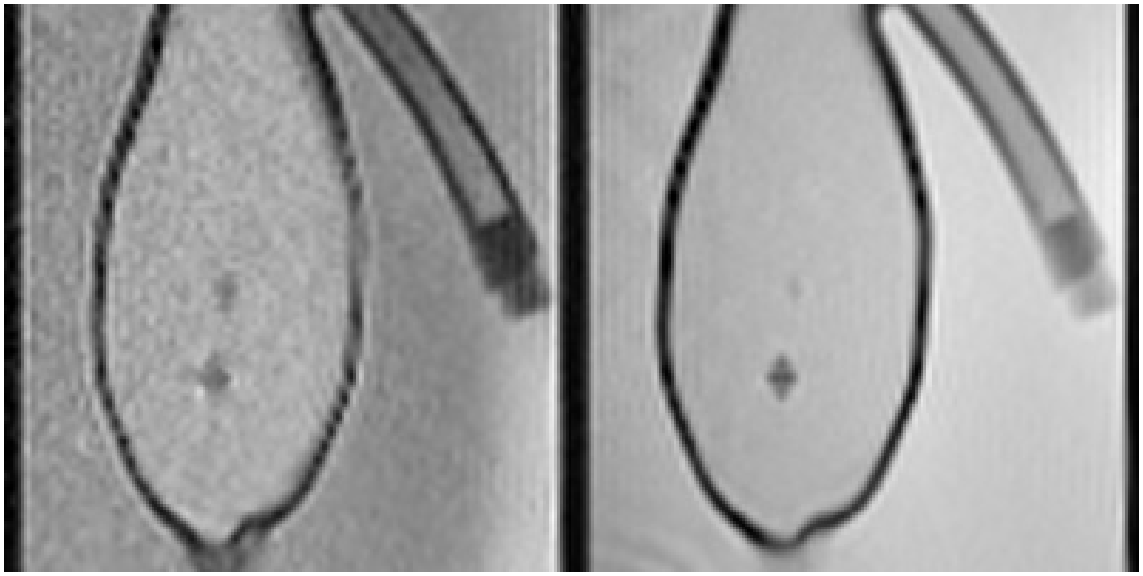
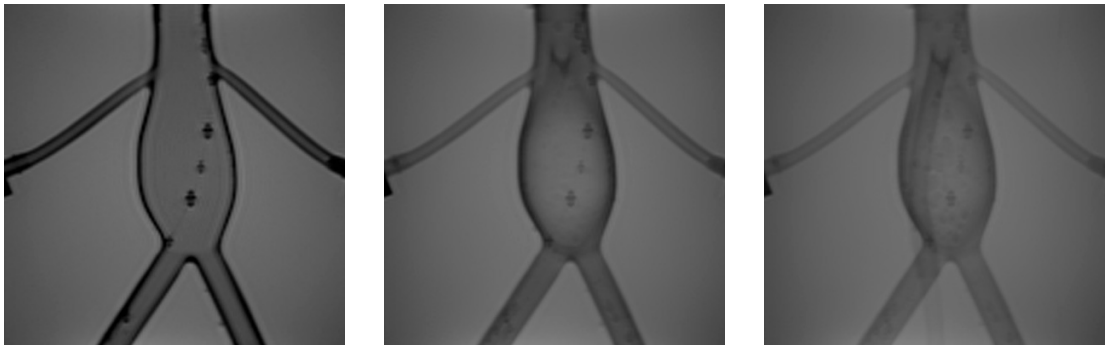


Figure 3.2: MR sequence comparison between GRE with a repetition time of 674 ms (left), and bSSFP with a repetition time of 477 ms (right). Both sequences have in-plane resolution of 1.4 mm and the slice thickness of 10 mm.

3.2.2 Slice thickness

Slice thickness significantly affects the marker's visibility, as the marker's signal will be distributed across a larger area. This effect is demonstrated in Figure 3.3, where the bSSFP sequence was used to capture the phantom at three different slice thicknesses.



(a) **(b)** **(c)**

Figure 3.3: Coronal images of aneurysm phantom with a) 8 mm, b) 30 mm, and c) 60 mm slice thickness.

The CNR comparison between bSSFP and GRE susceptibility artefact conspicuity was further analyzed in a series of varying slice thicknesses. The slice thickness was varied from 2.5 to 30 mm to compare its influence on marker CNR. The marker signal and surrounding lumen were divided by the standard deviation of the background noise, as shown in Figure 3.4. The analysis results are shown in the Figure 3.5.

The marker CNR peaks at around 5mm slice thickness in both sequences, while bSSFP has consistently higher CNR than GRE. The peak effect is due to the marker's three-dimensional susceptibility artefact size; the artefact is approximately 5 mm wide in coronal and sagittal directions. When the slice thickness is larger than 5mm, the CNR of the marker starts to drop as it's mixed with the surrounding lumen. However, the guidewire movement must be considered when choosing the slice thickness. In the case of 5mm, if the guidewire moves out of the slice slightly, the signal will quickly disappear. To counter this, a larger slice thickness was chosen, at 10mm, which loses some CNR but allows movement for the guidewire.

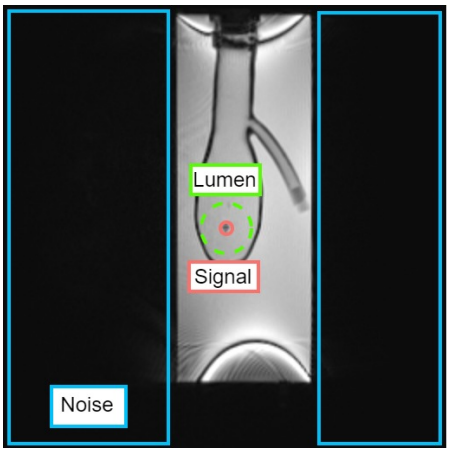


Figure 3.4: Example bSSFP image showing the ROI for noise (blue), lumen (green), and marker (red), used for CNR calculations

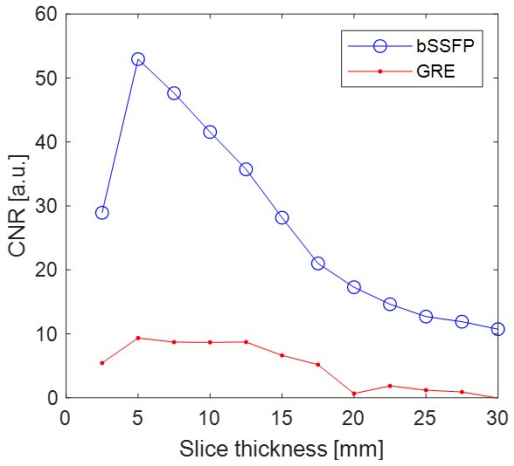


Figure 3.5: Variation of marker CNR with slice thickness for both GRE and bSSFP

3.2.3 Chosen MR sequence settings

After the analysis, the bSSFP sequence was chosen due to its more suitable near real-time capabilities. The acquisition time per frame was minimized as much as possible to approximately 400-500 ms while reaching an adequate in-plane resolution of around 1.3 to 1.4 mm. For imaging purposes, the optimal trade-off for slice thickness is around 10mm, as the CNR is still high, and the extra thickness provides some additional robustness for out-of-place movement.

3.3 Simulation Model and Theory

This section presents a procedural-generation approach to increasing the diversity of training data and avoiding time-consuming manual annotation of susceptibility artefacts. The simulation and implementation are presented using the Python programming language.

3.3.1 Local Magnetic Field Effect

Small paramagnetic rings are mounted on catheters and guidewires in the passive tracking approach of endovascular interventional MRI. These markers produce a local magnetic field inhomogeneity as a result of the difference in magnetic susceptibility of the marker and background tissue. These perturbations can be mathematically described as a magnetic dipole, as given by [38]:

$$B_z = \frac{B_0 \cdot \Delta\chi V}{4\pi} \cdot \frac{(x^2 + y^2 - 2z^2)}{(x^2 + y^2 + z^2)^{\frac{5}{2}}} \tag{3.1}$$

Where B_0 is the main magnetic field, oriented along the z-axis, $\Delta\chi$ is the susceptibility difference between the marker material and the surrounding tissues, V is the volume of the paramagnetic material and x, y and z represent the spatial coordinates.

The simulation of the Bz (Equation 3.1) is shown in Figure 3.6.

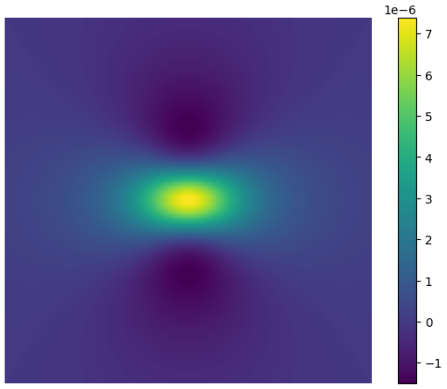


Figure 3.6: Dipole field distortion simulation

Transforming the magnetic field distortion into an artefact requires integration over the slice thickness [38]:

$$S_{coronal} = \int_{-d/2}^{d/2} \rho(x, y, z) \exp(-i\gamma B_z TE)\, dz \tag{3.2}$$

Here, $\rho(x, y, z)$ is the spin density, d is the slice thickness, γ (Hz/T) is gyromagnetic ratio (^{1}H) and TE the echo time. The composite trapezoidal rule was used to implement integration over slice thickness, and both coronal and axial directions were integrated.

The artefact scan was simulated using the following parameters: field of view (FOV) = 20 mm, imaging matrix = 300, echo time (TE) = 4.45 ms, and the susceptibility of the marker was chosen so that the dimensions match with previously gathered MR reference data $\Delta\chi V = 0.018$, due to the guidewire's marker composition being confidential.

The simulation of the integrated signals in the Coronal and Axial directions is shown in Figure 3.7. A linear colormap was applied to the simulation signal output to strengthen the contrast slightly.

3.4 Procedural CNN dataset generation

nnU-Net CNN was chosen for detection due to its ease of use, adaptability, and robustness. Therefore, a semantic segmentation dataset is required to train the CNN.

Semantic segmentation is a method for categorizing each pixel in an image into a class or object. In the case of generated susceptibility artefacts, there would be only two classifications: artefact magnitude and background. To be able to create a mask of the generated marker, the generated signal was normalized to a 0-1 range, and a binary threshold was applied to mask any signal up to magnitude 0.9. This created the outcome seen in Figure 3.8. To be able to input the mask into nnU-Net, the background pixel value was left at 0, and the artefact pixel value was remapped to 1, while the picture itself is 8-bit (0-255). This is because each label has a pixel value associated with it and there could be up to 255 labels for this training dataset.

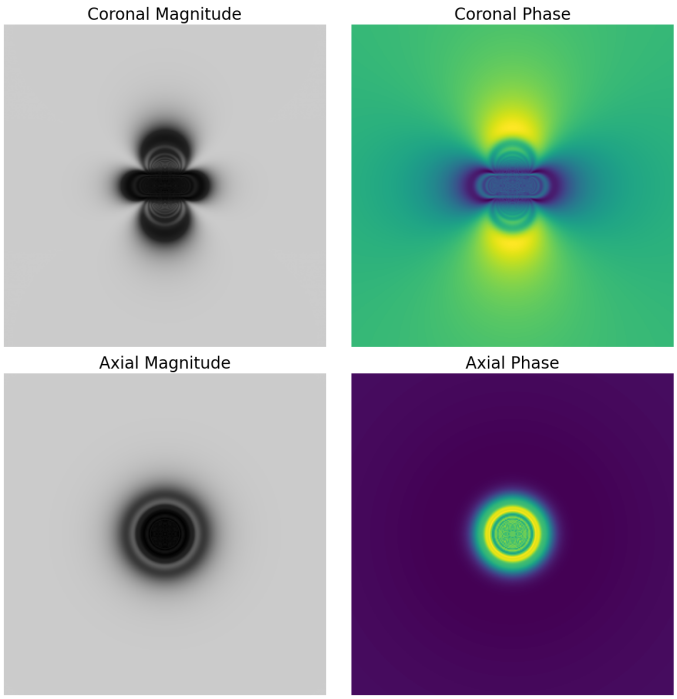


Figure 3.7: Coronal and Axial visualizations of synthetically generated susceptibility artefact.

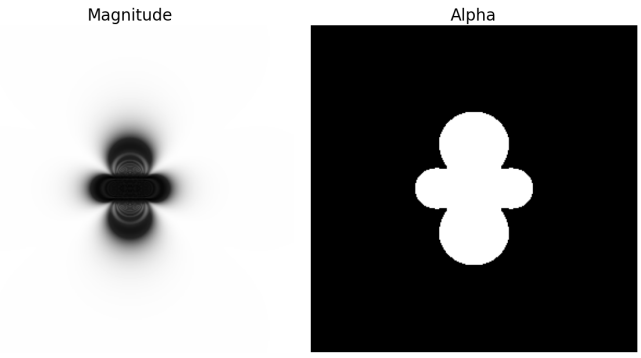


Figure 3.8: Susceptibility artefact segmentation mask derived from signal strength.

3.4.1 Dataset augmentation

On each image in the dataset, the artifacts should change slightly from picture to picture. This was achieved using augmentation methods to procedurally generate a unique dataset in a very short time frame. Only one artefact simulation was performed, all the augmentation methods created clones of the original simulation's output. The following three augmentation methods were used:

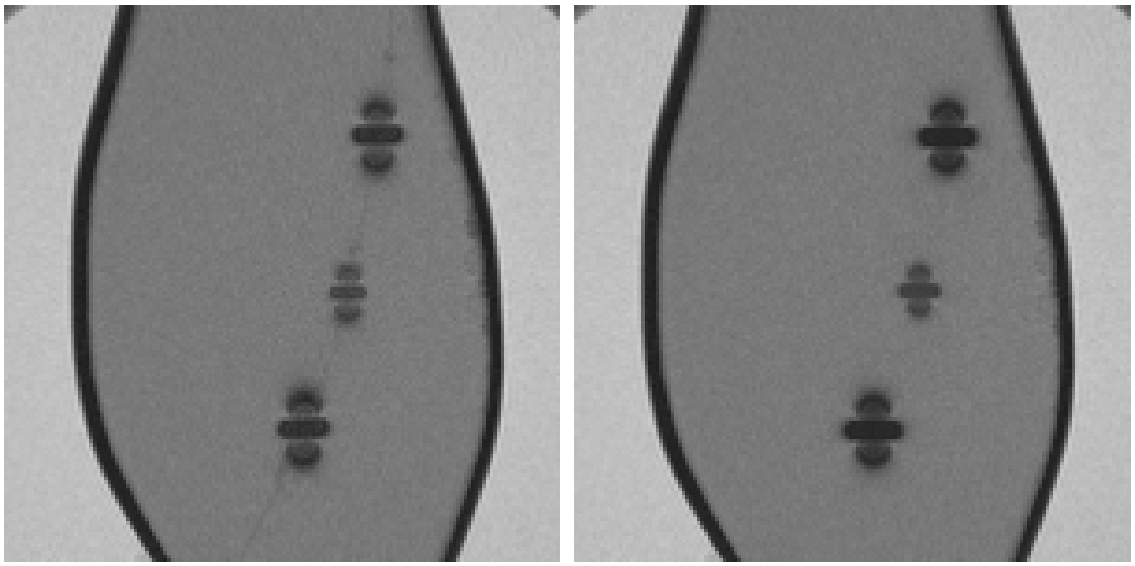
1. Artefact CNR
2. Artefact size variance
3. Artefact position on the image

CNR augmentation was implemented by blending the artefact with a white image based on a specified scale, to ensure various contrast levels across the dataset. The contrast strength was limited to a range, to represent realistic situations.

Size variance was achieved by scaling the simulation of the artefact to a specified range. As the original simulation had a very high resolution, this is quite a data-destructive process, as the image was scaled smaller by nearly 10 times, to reach realistic artefact dimensions.

Position augmentation ensured that new artefact positions were uniformly distributed within the dataset image boundaries (as shown in Figure 3.10) and maintained a minimum distance between artefacts to avoid clustering. The placement boundaries were not chosen to be anatomically correct, but rather to avoid artefact placement outside of the scanned signal.

A very realistic superposition was possible using the augmentation methods, as shown in Figure 3.9. The figure demonstrates superimposed artefacts with different sizes and varying contrast, compared to a real MR scan of the guidewire's artefacts.



(a) (b)

Figure 3.9: (a) True artefacts (b) synthetically generated artefacts.

The augmentation principles were applied to a set of 26 background images of the aneurysm phantom, noise, or real anatomy, which were acquired using the UT MR scanner or downloaded from an open-source dataset ([39]) in the case of real anatomy. A region of interest (ROI) is chosen for each background image, where simulated artefacts can be placed. For procedural placement, the following limits and settings were implemented:

- Output image size: 512x512 px
- Artefact size range: 6 - 12 px
- Artefact contrast range: 0.5 - 0.85 (1.0 would be a clean signal)
- Minimum distance between placed artefacts: 30 px
- Number of artefacts on image: 3 - 7

Sample images from the dataset are shown in Figure 3.10, while a sample procedurally generated image is shown in Figure 3.11.

3.5 Training and detection using CNN

nnU-Net is a very convenient tool for segmentation purposes. It is designed to analyze the provided training cases and automatically configure an optimized U-Net-based segmentation pipeline. It is geared towards medical image segmentation, so datasets from MRI, X-ray, or other mediums, as well as 3D data, are all accepted. In this case, the default 2D U-Net will be used as it is the most suitable option for near real-time detection.

Using the dataset generation and augmentation methods above, a dataset consisting of 700 images was generated and provided to nnU-Net. Each generated image has a matching ground truth mask linked to it. When the dataset is provided to nnU-Net, it generates a fingerprint, determining how the U-Net will be trained. In this case, the fingerprint created by nnU-Net has

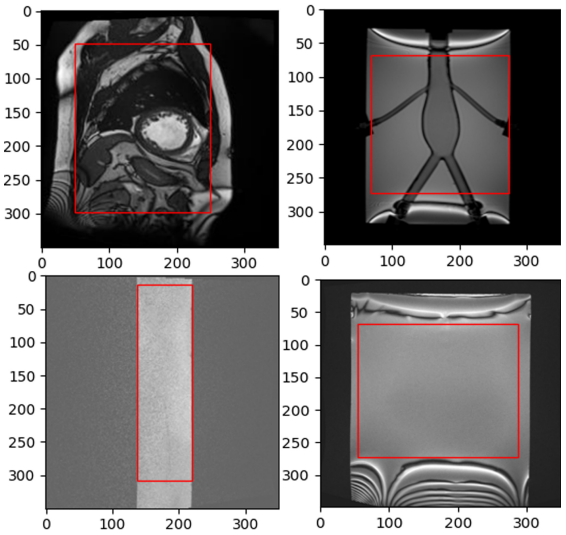


Figure 3.10: Examples of background images with a red rectangle region where procedurally generated marker artefacts are synthetically placed.

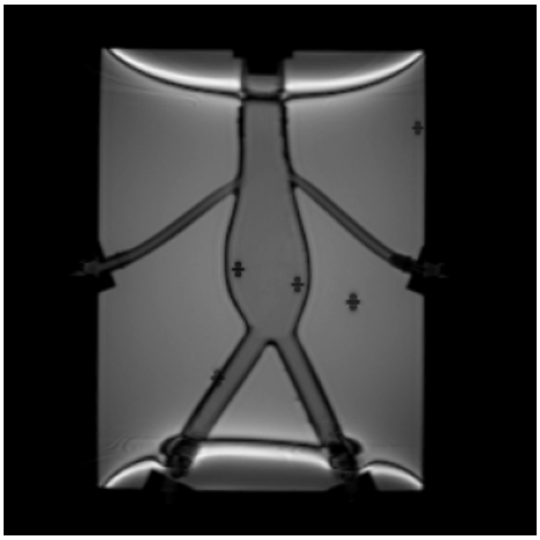


Figure 3.11: Procedurally generated dataset image.

2D architecture and uses a `PlainConvUNet`, a variant of the U-Net architecture with eight stages.

The training process involved standard 5-fold cross-validation, with each fold trained for 100 epochs. The training was conducted on an NVIDIA RTX 3090 GPU, each checkpoint of 100 epochs taking approximately 1 hour. Consequently, the entire training process spanned 5 hours. During the training, the pseudo-dice coefficient was monitored to ensure the segmentation accuracy was improving. The Figure 3.12, provided by nnU-Net, shows the training progress as a chart. The loss function shown in the graph (blue - training data, red - validation data) measures how well the model's predictions match the true labels in the data. Lower training loss values indicate that the model is learning to make better predictions on the data.

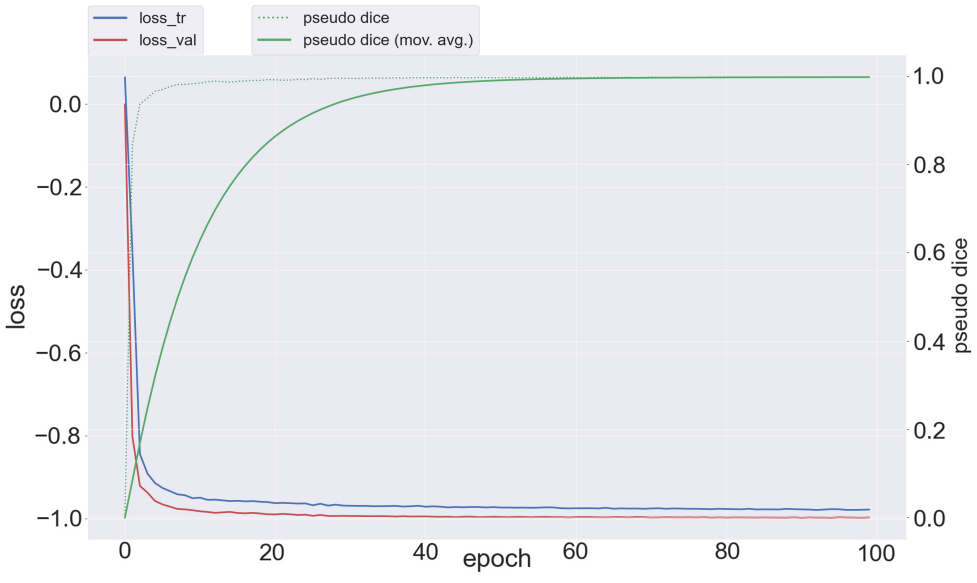


Figure 3.12: nnU-Net training progress over 100 epoch.

3.6 CNN training results and evaluation

The CNN performance was evaluated by comparing the prediction with manual annotation of susceptibility artefacts. A selection of nearly 70 images of various MR images of the guidewire, MR noise, and the aneurysm phantom was made. Python script was developed to annotate the selected images by allowing the user to pick points on each image where the user can see an artefact. Each picked point was cross-referenced with the CNN prediction to find matching, missed, and false detections. Inevitably, some human error will be part of the evaluation because some artefacts are hard to spot or do not look like artefacts, and therefore were not picked.

The entire collection of the results from the evaluation script is publicly available here: [1]. The number of manually annotated, matching, missed, and false detections is shown for each image, alongside the time taken for the CNN prediction.

Figure 3.13 shows a sample from the evaluation script output, with color-coded user-picked points and CNN prediction. The following color codes are displayed throughout this section:

- Red ■: This susceptibility artefact was predicted by CNN. If it does not have a yellow or green circle, it's either a false positive, or manual annotation user error (missed artefact).
- Green ■: This susceptibility artefact was detected by both, CNN and manual annotation.
- Yellow ■: This susceptibility artefact was detected only by manual annotation.

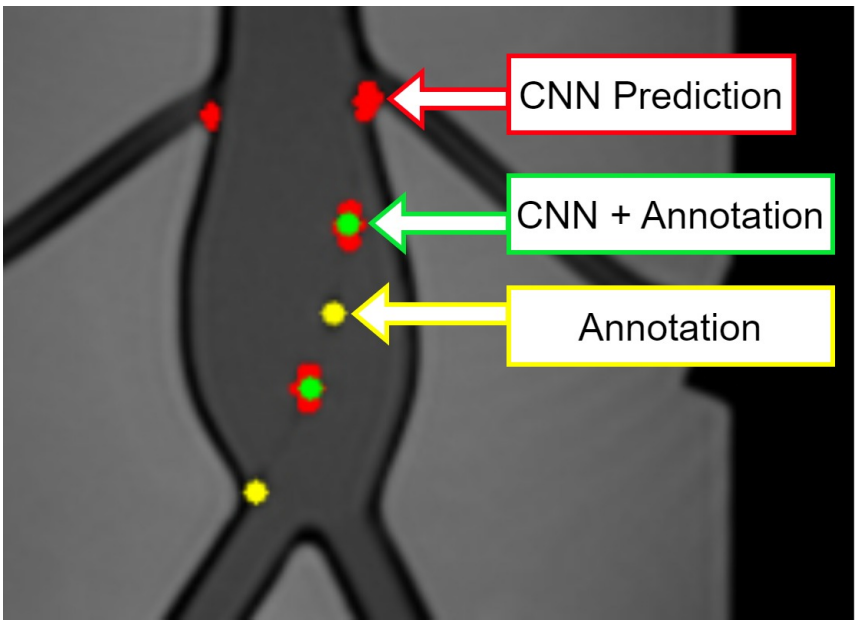


Figure 3.13: Sample from CNN evaluation script with color-coded manually annotated points (green, yellow) and the CNN prediction (red).

Several metrics were employed to quantify the CNN's performance in detecting susceptibility artefacts. These metrics provide insights into the model's ability to identify artefacts compared to human annotation.

[1] M. Reinok. "CNN Manual Evaluation Jupyter Notebook." Available online: https://github.com/martinreinok/master-thesis/blob/master/susceptibility-simulation/cnn-manual-evaluation.ipynb (accessed May 18, 2024).

3.6.1 DICE score

Dice is an evaluation metric used in computer vision and medical imaging to measure the similarity or overlap between two sets [40]. It quantifies how well the predicted region (e.g., object boundaries) aligns with the ground truth region (the actual location of objects in the image).

The Dice score is given by:

$$\text{DICE} = \frac{2|A \cap B|}{|A| + |B|}$$

where $A \cap B$ is the number of overlapping artefacts between manual annotation and CNN prediction. A represents the total number of predicted artefacts, and B represents the total number of manually annotated artefacts (considered ground truth).

Dice score provides a value between 0 and 1, where 0 indicates no similarity between the two sets, whereas a score of 1 indicates perfect overlap.

This evaluation dataset obtained a Dice score of **0.69**.

The achieved Dice score of 0.69 indicates a good, but not perfect, overlap between the predicted markers and the ground truth markers, suggesting that the model's predictions are fairly accurate but still have room for improvement.

3.6.2 Missed detections

Missed detections are defined as the number of artefacts the CNN did not detect, but were detected by the manual annotation, on average per image. This metric does not take human error into account, so in a case where the CNN would detect more legitimate artefacts than manual assessment, the score will not decrease.

To obtain missed detections, the difference between the number of manually picked points and CNN matching detections is averaged over the entire dataset. Our analysis shows the average number of missed detections per image is **1.33**.

Missed detections were further analyzed to obtain the rate of all missed detections, which is the total number of detections divided by the total number of annotations. Out of **256** manually annotated markers throughout the dataset, **164** were correctly detected, and **92** markers were missed, resulting in a **missed detection rate of 36%**.

Figure 3.17 shows an example of a situation where the CNN could not detect the artefacts, or missed many. As the Figure 3.17 illustrates, guidewire movement was the most common reason for missed detections. Under a bSSFP sequence, any movement will cause transient effects on the signal of the moved object, while the MR is re-establishing steady-state magnetization. These transient effects can be visually described as a ghosting effect. The effect's ability to reduce and smear signal is shown in Figure 3.15.

3.6.3 False positives

False positives are defined as the number of artefacts the CNN detected but were not detected by the manual annotation, on average per image. Similarly to the previous metric, this metric does not take human error into account, so if the CNN detects a legitimate artefact that is not detected by manual annotation, the score will not decrease but instead will increase.

To obtain false positives, all the CNN detections without matching manual annotations are averaged over the entire dataset. Throughout the dataset of a total of 69 images, **45** false positives were identified. In this analysis, the average number of false positive detections per image is **0.65**.

The most common case for false positives are actual false positive detections of elements that look like artefacts but are not. These might be air bubbles in the phantom, parts of the anatomy, or any paramagnetic inhomogeneity such as metal implants that will create a susceptibility

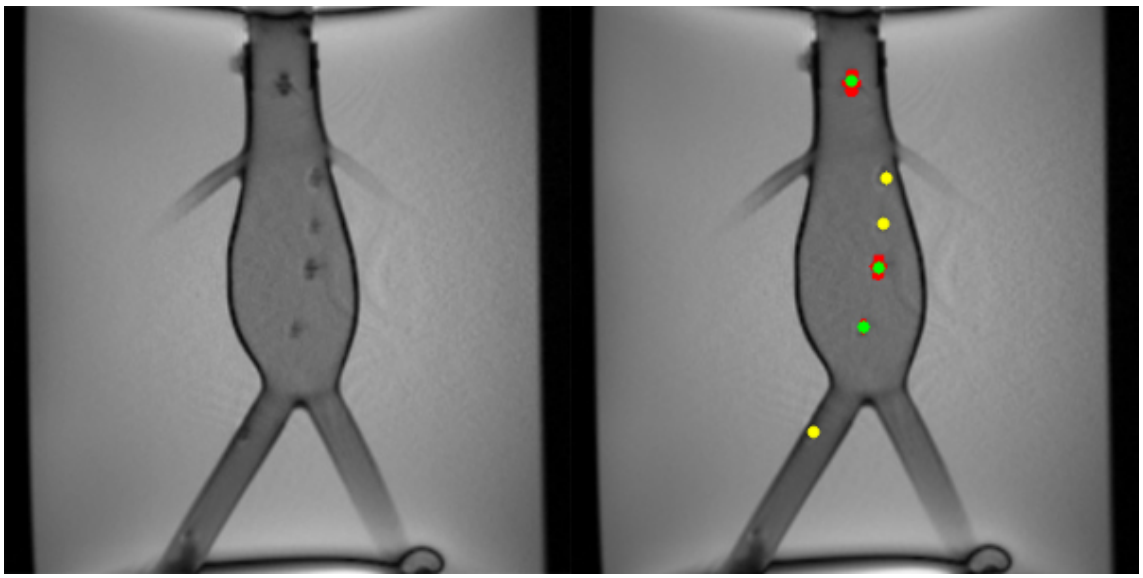


Figure 3.14: A frame with many missed detections caused by guidewire movement and the resulting loss of contrast in artefact signal. The left image shows the original MR image, while the right image has the evaluation script overlay.

artefact similar to those produced by the guidewire [41]. Figure 3.16 shows an example of such a case, where air bubbles trapped in the aneurysm phantom cause false positive detection. Improvement to the CNN was not considered to attempt to reduce such false detections as they can be quite indistinguishable from artefacts. Instead, the guidewire tracking algorithm described in Chapter 4 was adapted to be able to handle some false positives, while still tracking the guidewire.

In rare cases, the CNN was able to detect more artefacts than manual annotation. In such a case, the artefacts which are not verified by manual annotation are still considered false positives. Figure 3.17 shows an example of such a situation.

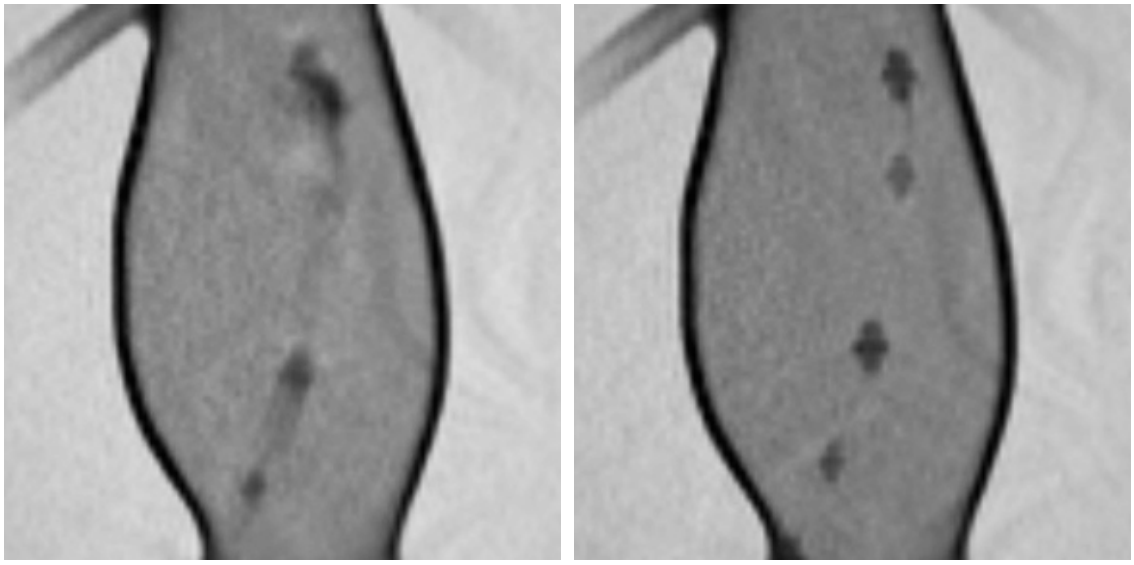
In earlier versions of the CNN training, noise was also an issue. The CNN detected many false positives if the image was very noisy or pure MR noise. This was significantly improved by including images with more noise in the training dataset. Figure 3.18 compares the detection of noise between an early version of the CNN prediction and the final CNN version, which does not predict anything from noisy images.

3.6.4 Prediction latency

The prediction speed was evaluated as the CNN prediction is intended to be used in a near real-time use case. Prediction latency quantifies how long each image takes to predict. As the nnU-Net CNN architecture is determined from the dataset, there is little control over the latency. The dataset image resolution could be lowered. However, this would also reduce image quality and, therefore, prediction accuracy.

To obtain prediction latency, the prediction time over the evaluation dataset was averaged. A histogram of the latency throughout the evaluation dataset is shown in Figure 3.19. In this analysis, the average prediction latency per image is **0.23** (seconds).

The latency can have considerable variance; however, no perceptible correlation with detection complexity was observed; rather, the variability appears to be stochastic, or connected to hardware/GPU optimization limitations. The histogram follows the Rayleigh distribution due to the many independent underlying factors that may contribute to the latency, such as data



(a) (b)

Figure 3.15: (a) Transient effects when moving guidewire very fast (b) Image after re-establishing steady-state magnetization after movement

loading, GPU processing time, and any hardware optimizations. Nevertheless, the majority of predictions fall under 0.25 seconds.

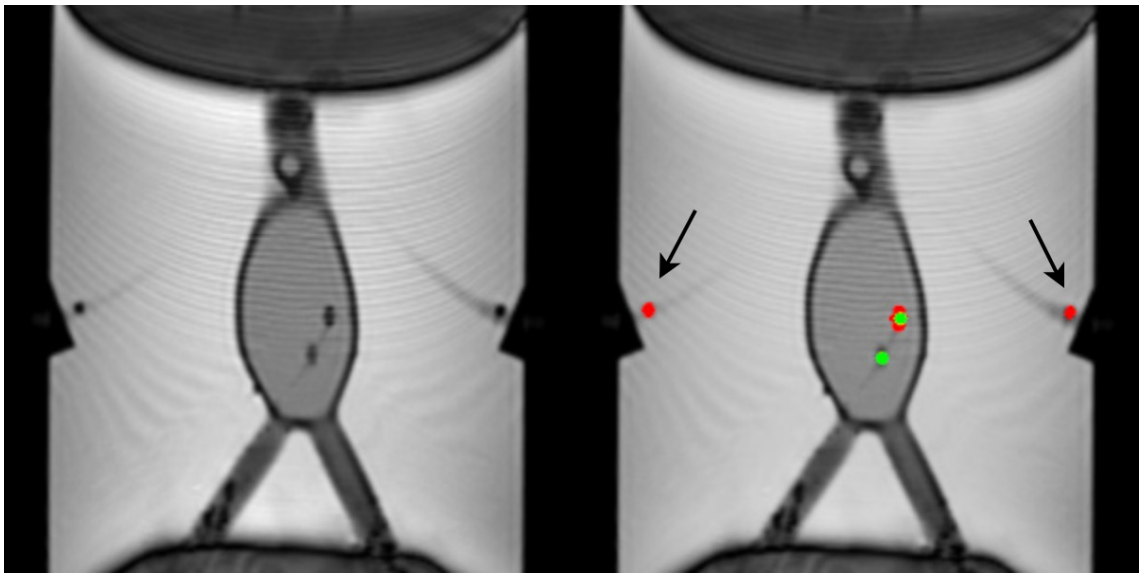


Figure 3.16: Example of false positive detection of air bubbles in the renal arteries of the aneurysm phantom (black arrows).

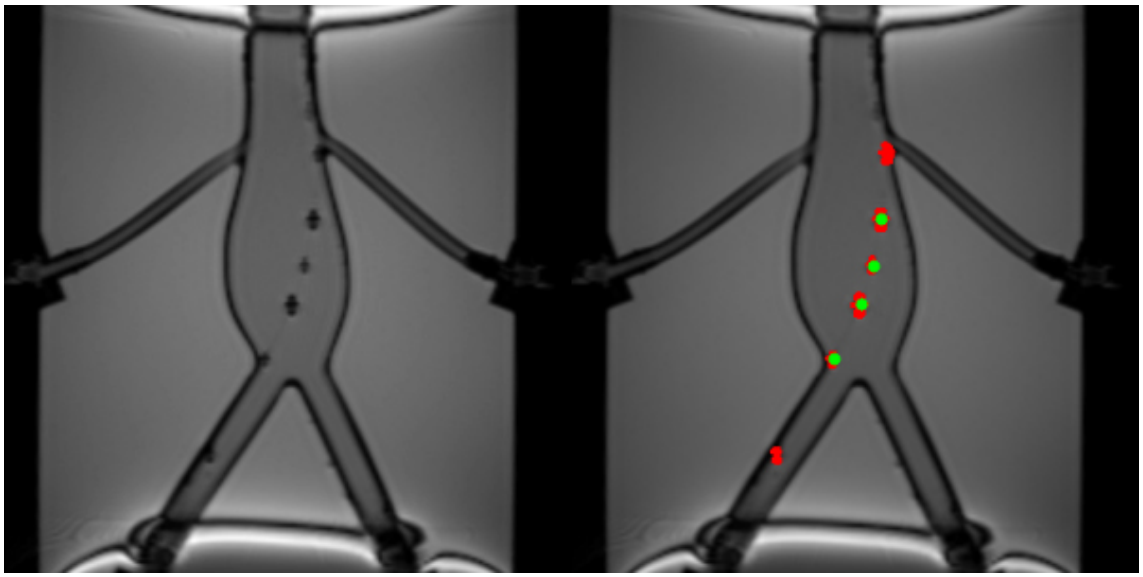


Figure 3.17: In this example, two false positives are detected, even though they are actually correct artefacts, however, during manual annotation, the artefacts looked ambiguous enough not to be selected.

(a) (b) (c)

Figure 3.18: Improvements to CNN prediction resistance to noise and noisy images. a) Original MR image, b) early version of the trained CNN with many false positive predictions, and c) final version of the CNN, which is nearly fully noise-resistant.

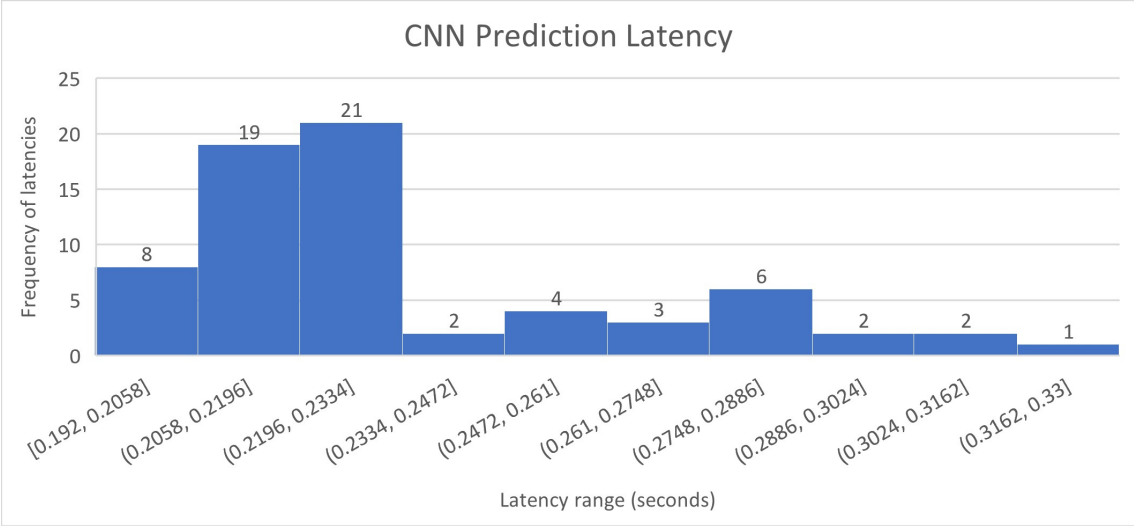


Figure 3.19: CNN prediction latency variation over the evaluation dataset. The latency is perceived as stochastic, not correlated with the detection difficulty, and follows the Rayleigh distribution.

4 Guidewire tracking and collision detection software

This chapter describes how the trained CNN was used with the UT MR scanner in a near real-time environment. This section's content outline is as follows:

1. A Python library for the Siemens-provided Access-i interface was created to interface the developed software with the MR scanner. Access-i allows software remote control of the MR scanner, which is crucial for the project.
2. A tracking algorithm was developed to track the detected guidewire and move the MR slice. The slice movement occurs in near real-time, while images from the interactive MR sequence are obtained through Access-i.
3. A guidewire collision detection system was developed, which utilizes a pre-scanned 3D model of the aneurysm phantom to detect collisions between the phantom and the guidewire. For this purpose, a 3D software suite has been developed, which allows the user to see the MR slice, image, and detected guidewire in relation to the aneurysm phantom 3D model in near real-time.
4. A software suite was developed to combine and intuitively use all the software described in this section.

4.1 Access-i

Access-i is Siemens MAGNETOM MR Scanner Interface for a 3rd party developer. The purpose of Access-i is to allow 3rd party clients (such as the author of the thesis) to establish bi-directional communication with the MR scanner. With Access-i, 3rd party clients are enabled to remotely control the MR scanner in the sense of parameterizing, executing, and retrieving image and projection data from arbitrary pulse sequences to establish a complete workflow and business logic beyond the MR host [42].

Access-i includes a comprehensive development guide for interfacing purposes and a simulator for testing the developed interface. The simulator allows the simulation of the Access-i interface in the MR scanner without having access to a physical scanner. From the MR host perspective, Access-i functions as a "Dot-AddIn" that can be added to a template or MR sequence to allow remote control.

This section describes the architecture and development of a Python library for interfacing with Access-i. Access-i was installed on the MR scanner at UT during this thesis, and this work was the first to use it, which is why a custom library was deemed useful and necessary. The library is a starting point for any future work or developments that might require remote control of the MRI scanner.

4.1.1 Communication

Access-i is an HTTPS-based REST interface that supports HTTPS GET and HTTPS POST requests. This means that commands can be issued to Access-i using requests, and Access-i will return an HTTPS response that contains the answer to the request in JSON format.

For example, the following GET request can be made to Access-i, to inquire about the MR slice thickness data:

```
https://125.28.36.11:7787/SRC/v2/product/parameter/standard/
getSliceThickness?sessionId=26eb060c-553d-4c15-bf5b-23cf29012763
```

Access-i will respond with the following return values represented as key/value pairs:

```
{"result":{"success":true,"reason":"ok",
"time":"20170608T143325.423"},
"value":10.0}
```


The response first returns the success of the request, followed by the value for slice thickness.

MR image data from the Access-i server is transmitted via a WebSocket connection. WebSocket offers a continuous, low-latency, two-way communication channel, which is advantageous compared to REST. Unlike the stateless, request-response model of REST, WebSockets maintain an open connection, allowing for low latency and efficient data exchange without the overhead of repeated HTTP requests.

The Access-i developer guide ([42]) provides guidelines and best practices for developing client-side applications. These were followed to create the library described below.

4.1.2 Python library

This library aims to make integrating software against the Access-i interface as easy as possible. The Access-i developer documentation provides hundreds of requests that can be sent to the MR scanner to inquire about the scanner state or change settings. Most of these requests have since been implemented within this library, which saves a lot of implementation time for future developments or integration with Access-i.

First, the library creates user-configurable settings to establish a connection to Access-i. These settings are in a class called `config`, and they allow configuring the IP address, port, version, and many more settings, which may require adjustment depending on the setup.

Next, the library follows the same hierarchy as the Access-i developer documentation, where the requests are categorized under services. Each service is unique and capable of controlling a specific part of the MR imaging process. For instance **Patient Service** can be used to retrieve the patient-related info, and **Template Execution Service** allows retrieving and executing all suitable templates. There are a total of 17 services.

10 services have been implemented in the Python library, with around 60 methods, focusing on the methods important for this project. This means that services or methods that do not benefit this work, for instance, **Table Service**, which is used to position the MR table, are not implemented as such functionality has not been necessary yet.

For every service in the Access-i developer documentation, a class is created that provides abstract methods for sending REST requests according to the documentation. For instance, **Authorization Service**, used to register the client and make sure that a valid license is used, lists the following operations in the documentation:

- register
- getIsRegistered
- deregister

Similar to the documentation, the Python library was made to have a class called **Authorization**, with three methods: **register**, **get_is_registered**, and **deregister**. To use these methods, for instance, when connecting to the Access-i Simulator, the following structure is used:

```
# Library is imported
import accessi as Access

# Configuration for accessing the simulator
Access.config.ip_address = "127.0.0.1"
Access.config.version = "v2"
```

```
# Registering to simulator
Access.Authorization.register()
```

The JSON response from Access-i is converted into a Python object to make the library more convenient to use. This makes accessing data from the response easier, for example, continuing the registration example from before:

```
# Registering to simulator
Access.Authorization.register()

# Check if the client is registered successfully
Access.Authorization.get_is_registered().result.success
```

For each implemented method in the library, the structure of the JSON response has also been copied to the Python method docstring so developers can easily understand the expected format of the data returned by Access-i without needing to refer to the documentation.

4.2 Tracking susceptibility artefacts

In MRI-guided interventions, real-time tracking of the passive guidewire is essential for accurate device navigation and automatic MR slice movement to the guidewire. However, there are many inherent challenges:

- Guidewire can move out of the plane.
- Noise and low SNR/CNR can affect detection.
- Flow and transient effects may impair the marker visibility.

To address these challenges, an ad hoc tracking algorithm was developed. Existing algorithms, such as the Kalman Filter, were considered. Kalman Filter is commonly used for linear state estimation in dynamic systems, combining measurements and predictions to provide accurate and efficient estimates of system states. However, in the current system, there can be a lot of unpredictable behavior, which makes implementing existing algorithms more complicated. The standard Kalman filter does not perform optimally with incomplete data and outliers [43], which can be the case with the CNN prediction data.

For instance, the CNN prediction can fluctuate during guidewire movement. In frame one, all artefacts can be detected. In frame two, the transient effects reduce artefact CNR slightly, causing detection not to work. In frame three, the artefact could be detected properly again. Figure 4.1 shows an example of such a situation, where a sequence of three consecutive frames are shown during guidewire movement. The slight CNR changes cause CNN to be unable to predict anything in the second frame, while in the third frame, all the artefacts are perfectly detected.

A custom Euclidean distance-based algorithm was created to cope with the instability, with a timer for tracking fluctuating artefacts. A tracker is initiated on a pixel-space coordinate, for each detected artefact. Each detected susceptibility artefact is labeled, and tracked for as long as they are visible. For each new frame from the MR, the tracker is updated, which causes it to search for the artefact it has been assigned to from a list of all tracked artefacts. This search has a limited distance and does not span across the image.

If the tracker can not find its artefact, it will be removed after five frames. This also makes the algorithm more robust against noise or false positives, which may cause a detection. The tracking for these false positives will be initialized as soon as they are (incorrectly) detected. However, the tracker will soon time out and be removed, as the CNN should not be able to detect noise/false positives consistently in the same spot.

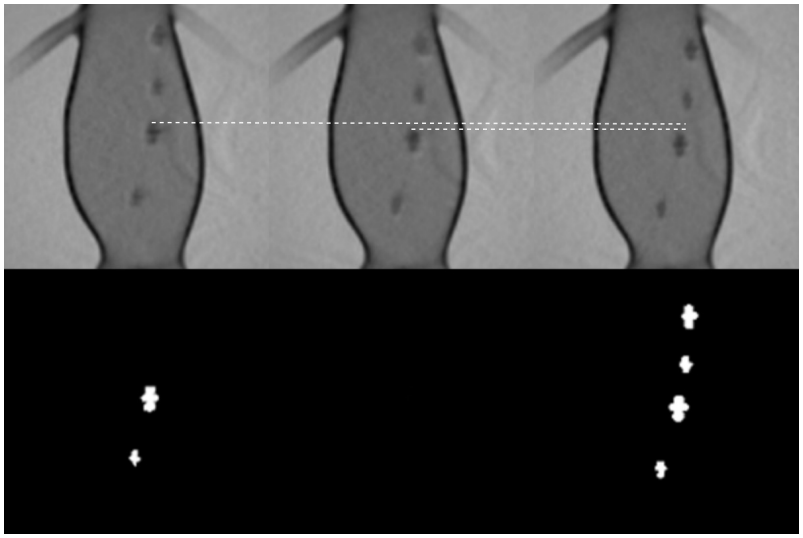


Figure 4.1: CNN detection fluctuation due to guidewire movement. White dashed lines are added to be able to spot the guidewire movement.

There could still be false positives, as any paramagnetic inhomogeneity such as air or metal implants will create a consistently detectable susceptibility artefact (as shown in Figure 3.16). However, these false positives are static in nature and stay in one position, as opposed to the guidewire which is in movement. To differentiate the artefacts produced by the static background from those produced by the guidewire, only the artefact with the largest movement vector is considered appropriate for moving the MR plane. An example of tracking is shown in Figure 4.2, where the algorithm could track two artefacts over multiple frames. The purple track is much longer, so when the user requests MR slice movement, calculations based on the purple start and end positions will be used, while blue will be ignored and reset. After the slice has been moved, the purple track will be re-established automatically, as its start and end positions are known. Any other trackers are reset, requiring a minimum of two frames to be re-initialized and considered for slice movement.

Continuous and manual tracking methods have been implemented to move the MR slice. In the case of continuous tracking, all possible artifacts are tracked. When an artefact crosses a minimum threshold from its starting position, it triggers an automatic slice alignment to the guidewire.

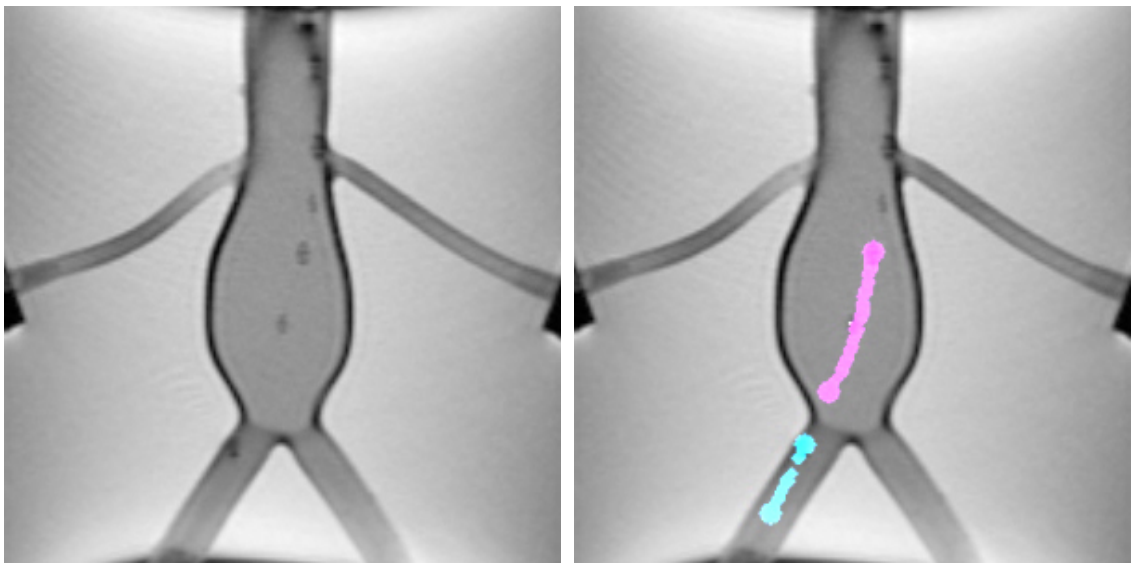
With manual tracking, the program user can press a button on the user interface, to command MR slice movement. Provided that sufficient tracking data is available, the slice will be moved.

After the slice movement request, the tracking algorithm enters a deadlock state. In this state, the algorithm continuously checks each incoming image's metadata to see if the MR slice position has moved to the target position. The complete flow of the tracking algorithm has been shown in Figure A.1.

Tracking accuracy

The tracking accuracy depends on the conversion from pixel space to MR voxel space in millimeters. Before the movement, the current slice position is obtained from the MR image metadata, and the tracking request is added to it. The voxel-size parameter, which can also be obtained from the image metadata, is used to convert the pixel tracking distances to millimeters. Voxel size is the 3D MRI analog to a pixel, except in 3D space, and it has a unit of millimeters, usually ranging from 0.5 to 2.0. The MR image metadata has both variables, voxel size and pixel dimensions, which means any location on the image can be converted to milli-

meters and vice versa. Once the slice movement request has been sent, the algorithm will wait until it receives an image with the new slice position, and continue the algorithm.



(a) (b)

Figure 4.2: (a) Original MR image (b) Tracked artefacts, larger circles represent the start and end position, while smaller dots show the movement trajectory over multiple frames.

4.3 3D Suite and collision detection

The guidewire collision detection was implemented using a pre-scanned 3D model of the phantom. The 3D model scanning workflow is similar to CT Fusion, combining multiple imaging modalities. In this case, a 3D isometric scan of the phantom is taken just before the procedure. This 3D scan is converted into a 3D mesh using 3D Slicer. 3D Slicer is a free, open-source software for visualization, processing, segmentation, registration, and analysis of medical, biomedical, and other 3D images and meshes [44]. The model obtained from 3D Slicer is input to the collision detection software. The collision detection software combines interactive imaging with the 3D model, to cross-validate if the detected artefacts intersect or collide with the 3D mesh.

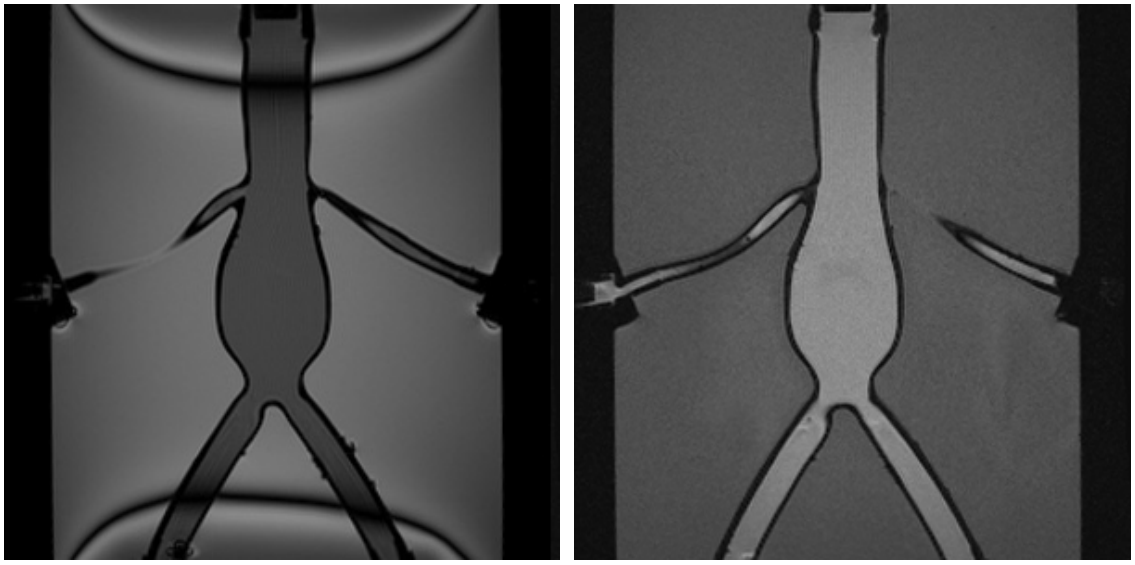
4.3.1 Scanning the 3D model

First, the aneurysm phantom is prepared at the MR table. This involves filling the phantom lumen compartment with 72 μM $MnCl_2$ blood-mimicking solution (T_1 = 770 ms and T_2 = 200 ms), and the outer compartment is filled with tapwater. The contrast difference between the solution and water is significant enough to be able to separate one from the other using 3D Slicer.

The sequence used to scan the model is a standard coronal isotropic 3D sequence with a 1 mm slice thickness, repeated over the entire phantom thickness. Both GRE and bSSFP were compared, and GRE is much preferred for this purpose, as it does not have the off-resonance banding artifacts that would otherwise distort the 3D model. Figure 4.3 compares a single slice from both sequences. The final sequence settings for creating the 3D scan are shown in Table 4.1.

3D Slicer

The Dicom image data from the MR scanner is imported in 3D Slicer, and a Median Image Filter with a neighborhood size of (1,1,1) is applied to the entire dataset. This makes the dataset more uniform for the next step, which is a threshold. A threshold with a suitable range is applied to the dataset, which extracts the lumen inside the phantom from the rest of the image.



(a) (b)

Figure 4.3: (a) 1mm slice taken from bSSFP isotropic scan, strong off-resonance banding artifacts can be noticed at the top and bottom of the image. (b) A 1mm slice taken from the GRE isotropic scan.

Table 4.1: 3D GRE Coronal Isotropic Scan Sequence

Parameter	**Value**
Sequence type	3D GRE
FOV [mm]	384 × 264
in-plane resolution [mm]	1 × 1
slice thickness [mm]	1
TR [ms]	1330
TE [ms]	5.07
Flip angle [°]	10

After the threshold step, the output segmentation might already be suitable, but cleaning some background noise with the scissors tool is often necessary. Then the model is ready to be used for collision detection. A screen capture of the 3D Slicer UI, 3D GRE dataset, threshold, and the resulting 3D model has been shown in Figure 4.4.

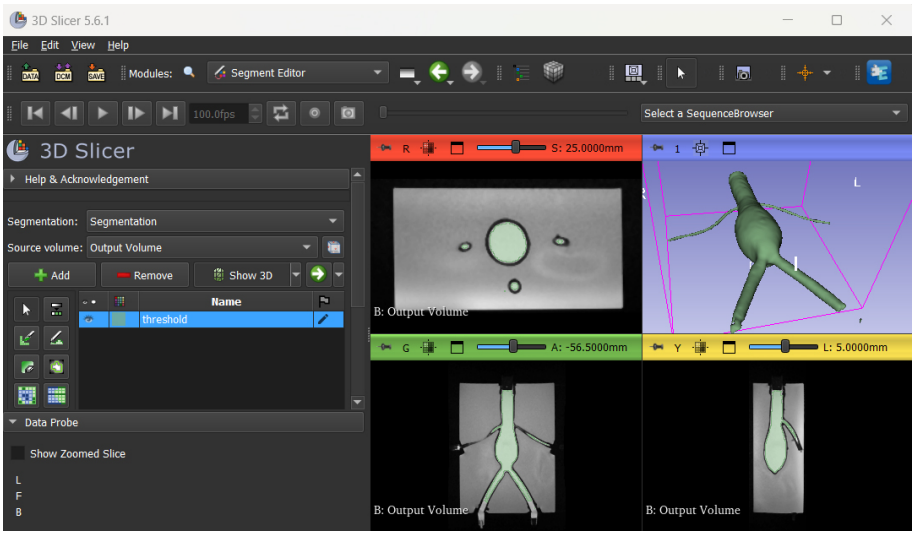


Figure 4.4: 3D Slicer user interface and the generated mesh from the 3D isotropic MRI scan.

4.3.2 3D Suite

For detecting collisions using a 3D model, an application was developed, which was called "3D Suite". 3D Suite aims to present CNN detection data in 3D space in relation to the 3D model, real-time MR image, and slice position.

To implement the 3D Suite, The Visualization Toolkit (VTK) was used. VTK is open-source software for manipulating and displaying scientific data. It comes with state-of-the-art tools for 3D rendering, a suite of widgets for 3D interaction and plotting [45].

VTK was used to load the mesh extracted from 3D Slicer. The mesh is a "shell" type with thin polygon walls. The walls were made see-through so that artefacts inside the lumen could be seen. The coordinate space used within VTK is the MR scanner's coordinate space, where both the MR slice and the phantom mesh are positioned relative to the MR isocenter.

The near real-time MR image, slice shape, and position were overlayed on the model to make it more integrated with the MR. The interface of 3D Suite is shown in Figure 4.5. All the MR-related data is updated in near real-time; as soon as an image or slice position is received from the scanner, it is relayed to the 3D Suite. The phantom mesh is statically in place, and if the phantom is physically moved, it would have to be re-scanned.

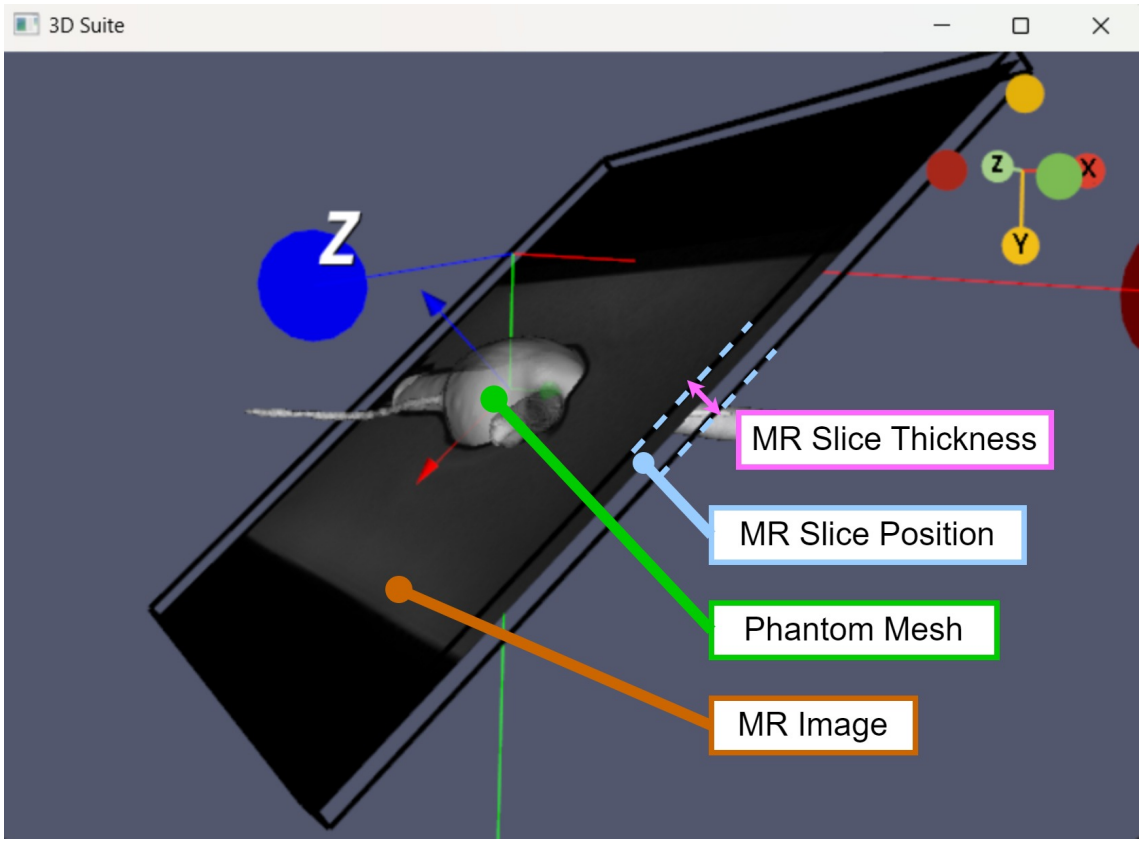


Figure 4.5: 3D Suite interface with aneurysm phantom mesh, MR slice, and MR image.

4.3.3 Collision detection

The collision between CNN detected guidewire, and the mesh is obtained using VTK's built-in method called `vtkCollisionDetectionFilter`, which calculates the intersection between two meshes. For this to work, the detected artefacts also need to become meshes. This was achieved by simplifying each artefact shape into a sphere with a fixed radius. The sphere radius allows setting the collision accuracy threshold; if the radius is large, collision will be detected early, while with a small radius the collision is detected late. The default sphere radius is 4mm, which is approximately the size of the artefact shape under MR.

If a collision is detected between the phantom mesh and guidewire, the artefact which is causing the collision is drawn as a red sphere, while normal artefacts are drawn as yellow spheres. The collision detection and presentation are demonstrated in Figure 4.6.

4.4 "Hub" software overview and working principle

An intuitive software platform was created to combine and integrate the 3D Suite, CNN detection, tracking algorithm, and Access-i. The software allows for creating a connection with the MR scanner, receiving, showing, and saving MR image data, and changing MR scanner parameters in real-time. This software was called "Hub".

4.4.1 Access-i Client

On the left side of the Hub, there is a sub-program called "Access-i Client". This program implements a GUI with the developed Access-i library (Section 4.1), which allows for requesting control of the MR scanner, choosing the MR template, starting the sequence, and changing parameters, all with a few button presses. This allowed for rapid prototyping and testing of

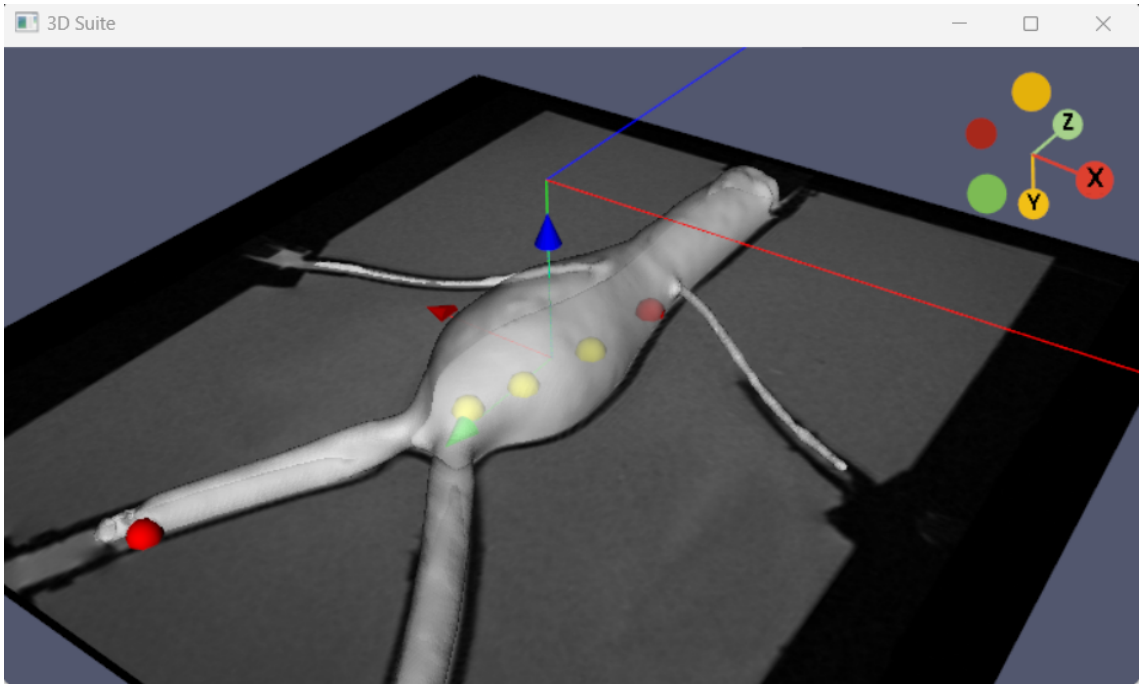


Figure 4.6: Collision detection in 3D Suite. The CNN-detected artifacts are drawn as spheres, where red spheres are currently colliding with phantom mesh, while yellow are not.

other software, such as the CNN or tracking algorithm, with the real MR and the Access-i simulator. The user interface of Access-i Client sub-program is shown in Figure 4.8.

4.4.2 Modules

On the right side of the Hub, there is a selection of sub-programs, known as modules, that establish the project's features and workflow. All communication between the modules was achieved using a publish-subscribe messaging pattern, where each module receives its input by subscribing to the output of another module, and also publishes its output for other modules to use. The complete Hub data flow has been shown in Figure 4.9. This section describes what each module does.

Access-i Websocket

The Access-i Websocket module receives any images the MR scanner sends and makes them available for the other modules, such as CNN detection or 3D Suite. Once the client registers with the MR scanner, this module will be immediately activated. As is common with all the other modules, the Websocket module allows "Show Output" and "Save Images" features, which correspondingly open a new window to show the MR image in real-time and save any received images into a folder on the local computer.

Additionally, while the module receives images, the real-time latency is calculated and shown in the module's options. This latency is calculated by comparing the time difference between the Dicom image date stamp and the current time in the computer. This latency is also calculated for the CNN module, as can be seen on Figure 4.10. The latency is accurate only if the MR scanner computer and the Hub computer are using the same network time server. For the purposes of this thesis, the UT internal NTP server was used, as the MR scanner computer does not have access to the external network.

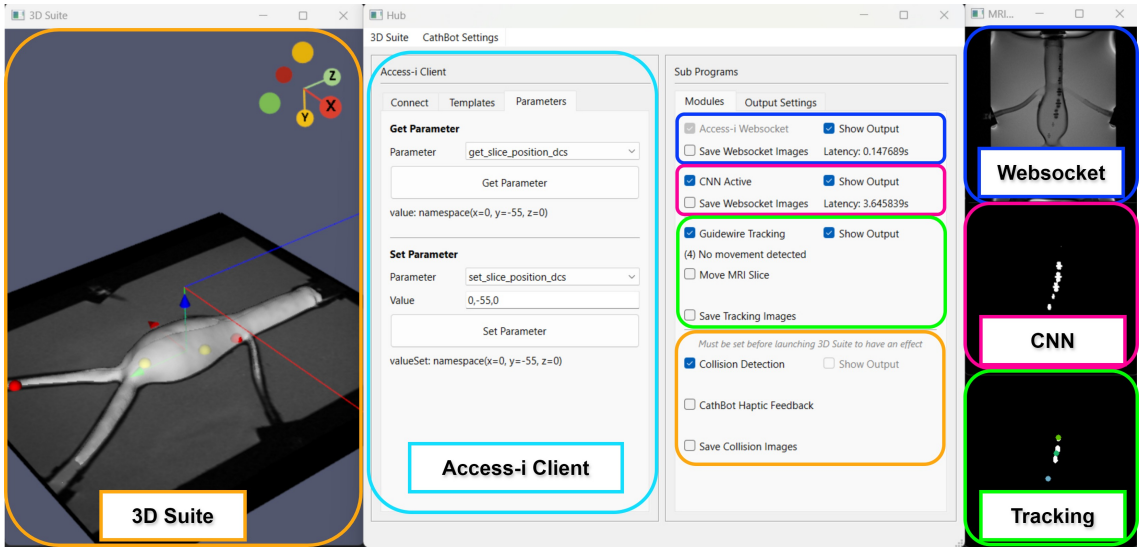


Figure 4.7: The developed integration application called **Hub**. The Hub allows access to nearly all the software developed over the course of this project. **Access-i Client** allows for connection to the MR system and starting or stopping the scan. The **Websocket** module manages communication with the MR scanner and receives real-time image data. **CNN**, **Tracking**, and **3D Suite** use the **Websocket** image data to detect the guidewire, track the guidewire or move the MR slice, and detect collisions and provide haptic feedback accordingly.

CNN detection

CNN detection loads the pre-trained CNN model and prepares it for running inference in real-time. The images are obtained from **Access-i Websocket** module using a LIFO (last in, first out) queue, so the most recently obtained image will be processed first, while the rest are discarded. This is due to the frequency difference between the Websocket images and CNN prediction. Websocket can receive and output up to 30 frames per second, while the CNN can predict approximately five images per second. Once the CNN prediction module is up and running, the remaining modules become available: guidewire tracking, collision detection, and 3D Suite (in full capacity).

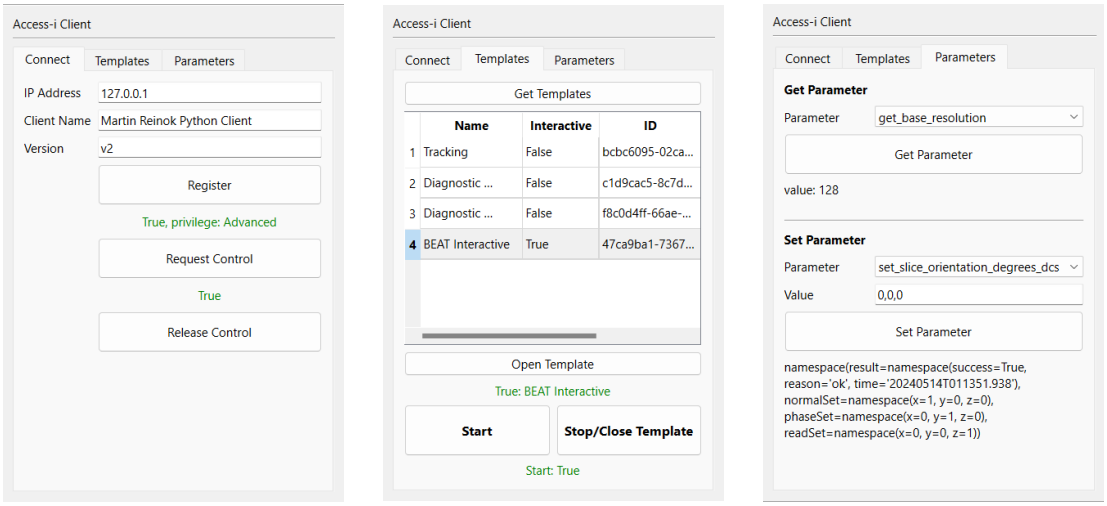
Guidewire tracking

The Guidewire tracking module is responsible for moving the MR slice and tracking the guidewire. It receives input from the CNN module and processes the images to pinpoint artefact coordinates, as described in Section 4.2. The tracking module also provides a checkbox option to toggle MR slice tracking on or off. When slice tracking is turned on, the Hub continuously sends MR slice movement requests to the MR scanner. New requests are only sent once the previous request has been completed and the slice has moved.

Collision Detection

The collision detection module enables the 3D Suite to detect collisions. If collision detection is enabled, the Hub sends detected artefact coordinates to the 3D Suite, which processes this data to calculate collisions.

Collision detection has an additional checkbox option to enable haptic feedback for the CathBot main controller. Once enabled, the 3D Suite will periodically send CAN bus messages to increase or decrease the haptic feedback on the main controller. For this function to work, a USB to CAN bus device must be connected to the Hub computer.



(a) **(b)** **(c)**

Figure 4.8: Access-i library graphical implementation in the Hub. a) The Connect tab allows for establishing a connection to the MR scanner. b) The Template tab is used to select interactive templates and start the MR scan. c) The Parameters tab can be used to change or request any MR scanner parameters.

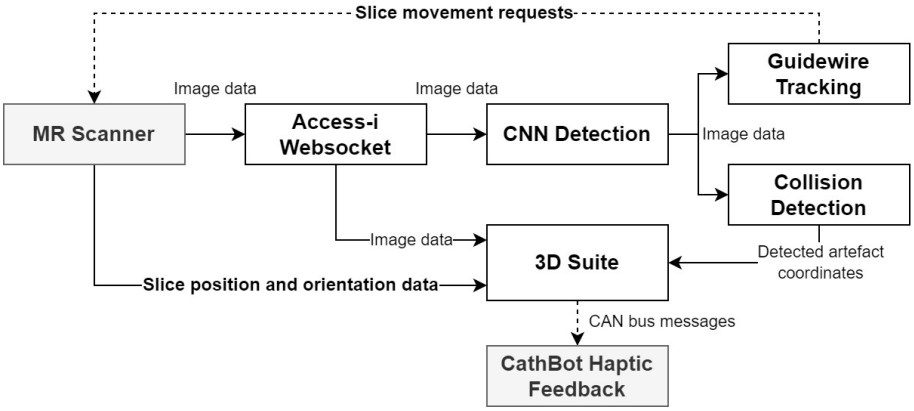


Figure 4.9: The Hub publish-subscribe data flow visualization.

4.5 End-to-end System Latency

The end-to-end system latency was analyzed to understand the real-time feasibility of using the developed software. The tests involve the complete system, where the time for the following modules is measured: MR sequence, Access-i server, CNN prediction, Guidewire Tracking, and 3D Suite. During the tests, the CathBot main controller was connected to the system and the 3D Suite was sending out CAN bus messages to change the resistance of the main controller's manipulator. However, the CAN bus messages were not included in the latency since they are delivered in less than one millisecond.

Three tests with different MR sequences were conducted to determine how the sequence choice affects the system's latency.

To set a reasonable target, the desired time window for post-processing was chosen here to be equal to the MR image acquisition and reconstruction time. This will be hereafter referred to as the 'deadline' to deliver all post-processing. This target establishes that the software should be able to fully process an image and request haptic feedback before a new image is retrieved from the MR. In the sections below, Figure 4.11, Figure 4.12 and Figure 4.13 show the average latencies, where the "MRI SEQUENCE" and "ACCESS-I" modules establish the deadline, shown with a green dotted line, for the modules.

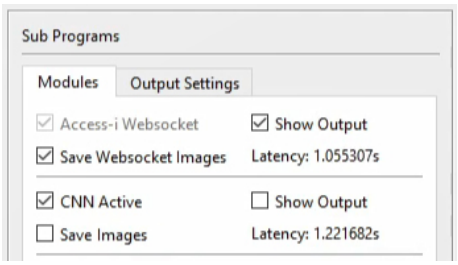
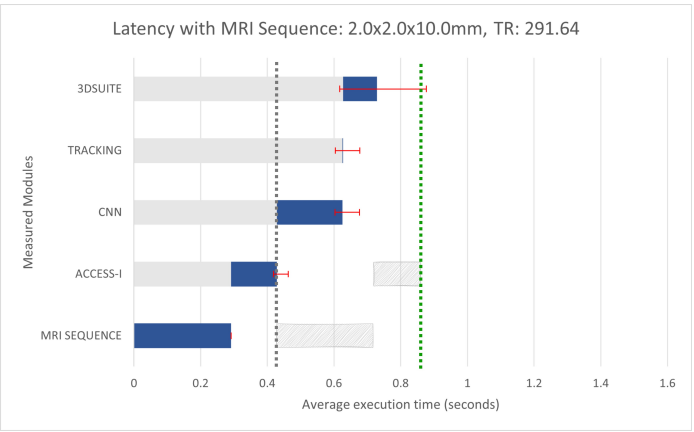


Figure 4.10: Websocket and CNN modules options and the real-time latency of the images.

4.5.1 Latency 1: Low spatial resolution

The first test was measured using a 2.0 mm voxel size, which resulted in a repetition time of 291.6 ms. Due to the low spatial resolution, this sequence is not reliable for interactive collision detection. The CNN was not able to detect an artefact consistently from this image. On average, the combined modules' execution took 0.37 seconds, whereas the MR was able to produce an image every 0.42 seconds. When the combined modules' execution took slightly longer than average, the deadline would be crossed, and the system would skip an image for prediction. With the current software and hardware combination, this sequence would be the fastest imaging rate achievable, if the CNN artefact detection were reliable. Increasing the MR sequence framerate even further would not provide any faster haptic feedback or guidewire detection and will negatively impact CNN prediction if the spatial resolution goes even lower.



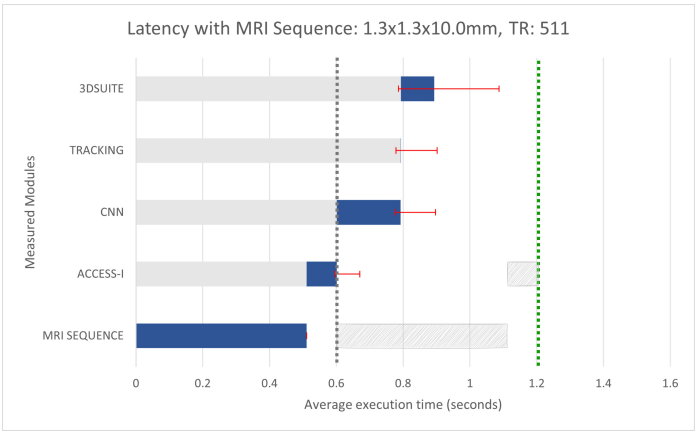
(a)

(b)

Figure 4.11: a) The latency graph for a MR sequence with low spatial resolution, but high temporal resolution. Each bar shows the average execution time with minimum and maximum deviation. The green dotted line is the deadline for modules. b) A sample image with this sequence, TR: 291, Voxel size: 2mm, image resolution: 160px

4.5.2 Latency 2: Optimal spatial resolution

The second test used a 1.3 mm voxel size and a repetition time of 511 ms. This is the preferred sequence for interactive collision detection since it provides a fast enough imaging rate while keeping the SNR quality at a suitable range for stable CNN prediction. With this sequence, the modules' execution took 0.37 seconds on average, whereas the MR was able to produce an image every 0.6 seconds. The modules can comfortably reach the deadline even in worst-case scenarios.



(a)

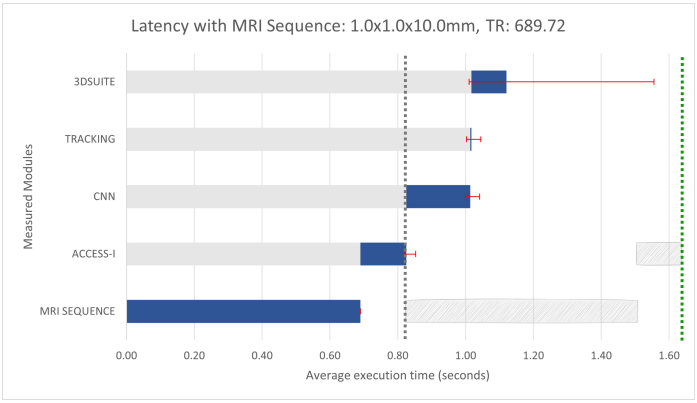
(b)

Figure 4.12: a) The latency graph for an MR sequence with optimal spatiotemporal resolution. Each bar shows the average execution time with minimum and maximum deviation. The green dotted line is the deadline for modules. b) A sample image with this sequence, TR: 511, Voxel size: 1.3mm, image resolution: 256px

4.5.3 Latency 3: High spatial resolution

The third test used a 1.0 mm voxel size, with a repetition time of 689.7 ms. This sequence shows excellent SNR for collision detection. However, the longer acquisition time does not make it the preferred option for collision detection with the software. Once again, the modules' execution took 0.37 seconds on average, whereas the MR was able to produce an image every 0.83 seconds.

During this particular test, the 3D Suite camera position was moved by the user, which caused the 3D rendering to hang until the user stopped moving the 3D camera position. This resulted in a very high deviation for the latency measurement, as seen in Figure 4.13.



(a)

(b)

Figure 4.13: a) The latency graph for a MR sequence with high spatial resolution, but low temporal resolution. Each bar shows the average execution time with minimum and maximum deviation. The green dotted line is the deadline for modules. b) A sample image with this sequence, TR: 689, Voxel size: 1mm, Resolution: 320px

The analysis of end-to-end system latency shows the feasibility of using the developed software for real-time applications with MR imaging. The tests provide insights into the system's performance and limitations under different MR sequences.

The low spatial resolution sequence (2.0 mm voxel size) showed that while the system could meet the real-time deadline, the CNN's reliability in detecting artifacts was compromised due to the lower image quality.

The optimal spatial resolution sequence (1.3 mm voxel size) achieved the best balance, providing a suitable imaging rate and stable CNN predictions, comfortably meeting the real-time processing requirements.

The high spatial resolution sequence (1.0 mm voxel size) offered excellent image quality but at the cost of longer acquisition times. The low temporal resolution made it less ideal for real-time collision detection.

5 CathBot Robotic Platform Improvements and Integration

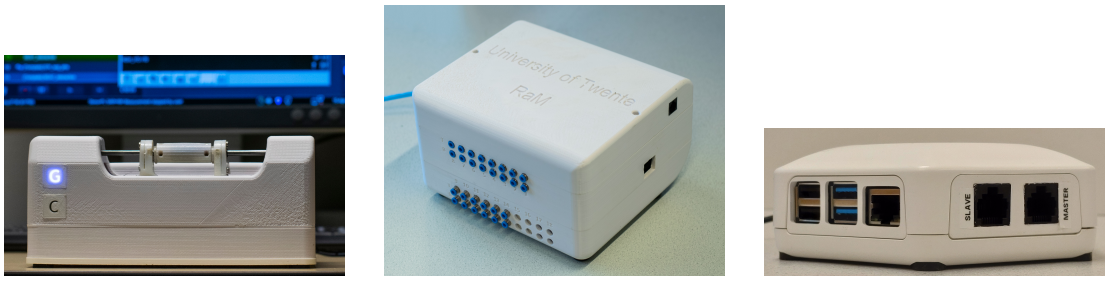
This chapter describes integrating the CathBot robotic platform with the software and methodologies outlined in Chapter 3 and Chapter 4.

First, a comprehensive overhaul of the CathBot software is undertaken, aiming to improve system stability, simplify management, and ensure future scalability. This involves analyzing inefficiencies and incorrect behavior in the existing system and implementing effective solutions. The software analysis covers all three embedded software components of CathBot: the main controller (ESP32), the pneumatic controller (2x ESP32), and the high-level controller (Raspberry Pi) (Figure 5.1).

Second, the tracking software described in Chapter 4 is integrated with CathBot to provide haptic feedback during endovascular procedures. This involves designing a haptic feedback model that is smooth and comfortable to use while providing valuable information about the endovascular devices controlled by CathBot.

5.1 Original CathBot Platform Analysis

CathBot Platform consisted of three components with embedded software: The main controller, which has an ESP32 microcontroller; the pneumatic controller, which has two ESP32 microcontrollers; and a high-level controller Raspberry Pi with ROS2 nodes.



(a) Main Controller **(b)** Pneumatic Controller **(c)** High-Level Controller

Figure 5.1: CathBot platform software components

Communication between the components uses the CAN bus protocol, with one bus designated for each, the main controller, and the pneumatic controller. Both buses are connected to, and created by the Raspberry Pi. This architecture is visualized in Figure 5.2. When the "Operator" pilots the main controller, a CAN bus (CAN1) message is sent to the high-level controller where a ROS2 node dedicated to the main controller picks it up and forwards it to another ROS2 node dedicated to the pneumatic controller. The second ROS2 node parses the data and sends a CAN bus (CAN2) message to the pneumatic controller, which generates appropriate pneumatic pressure patterns to move the endovascular devices.

Some issues were discovered while operating the original CathBot. These issues, described in the following sections, had to be solved to get the most functional value from the CathBot integration with the haptic feedback software layer.

5.1.1 Stability issues

The high-level controller (Figure 5.3), or Raspberry Pi 4B used the Ubuntu MATE lightweight Operating System. However, it was outdated and crashed or froze often, limiting the system's capability. The Raspberry Pi's functionality in the system was quite critical, as it parsed and re-

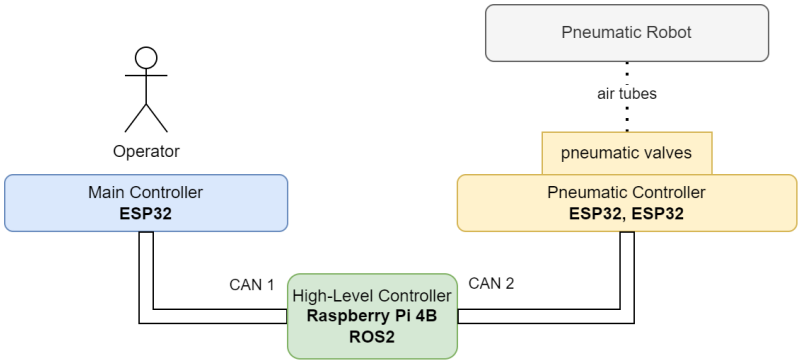


Figure 5.2: Original CathBot platform architecture.

layed messages from one component to another. Additionally, the Operating System had some troubles with the initial boot; it might get stuck and never boot until a monitor is connected and a button is pressed on the keyboard.
Furthermore, the Raspberry Pi had a dual-channel CAN bus expansion HAT, where each channel was supposed to be designated to a device (either pneumatic or main controller). However, the Linux kernel randomly mapped the CAN bus channels to physical ports whenever the Raspberry Pi was restarted. After multiple attempts, a solution was not found for this issue and it caused some uncertainty in the system as sometimes the CAN bus cables had to be switched around for connections between devices to work.

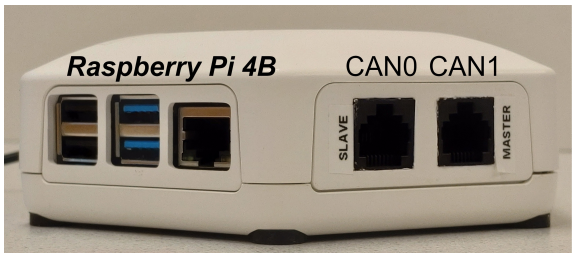


Figure 5.3: CathBot High-Level Controller, Raspberry Pi 4B with dual-channel CAN bus HAT.

The main controller (Figure 5.4) also had some stability issues. The manipulator was difficult or very heavy to interact with, and sometimes, it became stuck, causing the pneumatic valves of the pneumatic controller to switch on and off continuously and out of control. Additionally, the manipulator's rotation was a bit temperamental. It would sometimes forcefully rotate only in one direction, driven by the rotary motor, even if the user tried to rotate in a different direction.

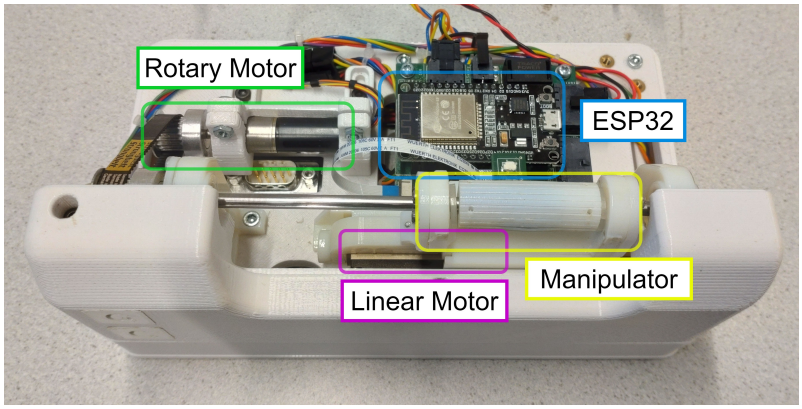


Figure 5.4: CathBot main controller internal components.

5.1.2 Comfort and usability issues

CathBot was quite complicated to set up. First, the Raspberry Pi had to be turned on with a monitor and keyboard. Then, two ROS2 nodes had to be started, each communicating over the designated CAN bus channel. After this, the main and pneumatic controllers could be connected to the Raspberry Pi. A special message had to be sent over the CAN bus from the Raspberry Pi to initialize and home the main controller. If sending this message did not work, the CAN bus channels were assigned incorrectly during boot and the cables had to be switched around. Now, CathBot was ready for operation.

During operation, as mentioned earlier, the main controller's manipulator had quite a lot of resistance to it. It was not easy to move and did not feel realistic.
Additionally, the pneumatic controller clamped the selected endovascular device whenever the operator gripped the manipulator with their fingers. While this seems correct and natural, device rotation was impossible, as rotating a guidewire or catheter while clamped would strain and break it.

5.1.3 Addressing Issues

After analyzing the original CathBot platform, several critical challenges held back its functionality and usability. These challenges included stability issues, operational complexities, suboptimal design choices, and poor user experience. The following section (Section 5.2) aims to improve all the identified issues, with the final goal of having a stable CathBot platform with user-friendly and responsive controls, while not compromising the functional operation of the robot.

5.2 CathBot Platform Optimization Approach and Implementation

All the issues mentioned in Section 5.1 are software-related. The hardware is well-designed and functional. The software side of the CathBot main, high-level, and pneumatic controllers was overengineered. It was not built to the standard and did not consider modularity, readability, or expandability and due to that, the errors and faults may have been coded in. The communication over two separate CAN bus channels was not necessary as the CAN bus load never raised above 10% in either channel. The high-level controller was considered unnecessary to the system, as it did not perform any vital role besides forwarding messages from one controller to another (or one CAN channel to other).

This section will describe how these challenges were resolved.

5.2.1 High-Level Controller and Architecture

The high-level controller Raspberry Pi was removed from the system. Its only functions were creating the CAN bus channels and mirroring messages from controller to controller. This did not justify its overhead and extra ROS2 code required for parsing and mirroring CAN bus messages. Instead, the CathBot was redesigned to work off of a single CAN bus channel, which starts at the main controller and ends at the pneumatic controller. Redesigned architecture is visualized in Figure 5.5 and Figure 5.6.

The new architecture allows for system expansions by branching out the CAN bus, as shown in the Figure 5.5, "USB to CAN" adapter is used in this case. It could also be a raw CAN bus device. When connected, these devices see all communication between the main, pneumatic, and other components connected to the bus, and communicate with any device.

This architecture was selected for its straightforward setup, where the system would be fully functional by directly connecting the main controller to the pneumatic controller, using an Ethernet cable. This architecture also keeps in mind that if any functionality should be added to the system, e.g., haptic feedback for the main controller as implemented in this thesis, the

functionality can be added by creating a branch in the CAN bus line using an Ethernet splitter. When connected, the new device can communicate directly with any other device on the bus.

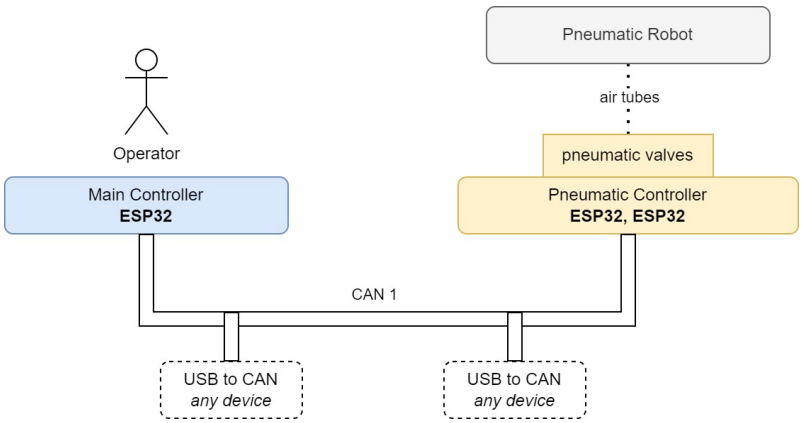


Figure 5.5: CathBot redesigned architecture overview.

5.2.2 CAN bus communication

As the Raspberry Pi was removed from the architecture, the CAN bus communication had to be changed because the main controller had to communicate directly with the pneumatic controller, instead of through an intermediary, the Raspberry Pi.

The minimal CAN bus communication in the CathBot system consists of two motor controllers (linear and rotary), the main controller, and the pneumatic controller (Figure 5.6). The main controller communicates two-way with both motor controllers to move them and acquire position or current data. The pneumatic controller only receives messages from the main controller and does not communicate anything back. As shown in Figure 5.6, any external devices will be connected between the main and pneumatic controllers, as a separate branch created by a Ethernet splitter.

It must be noted that the High-Speed CAN bus standard (ISO 11898) recommends keeping the branch lengths up to 0.3m to ensure the maximum bus speed of 1Mbps [46]. In our case, the CAN bus line connecting the main controller to the pneumatic controller is relatively short at 5m, compared to the ISO 11898 suggested limit of 40m (for 1Mbps), which means slightly longer branch lengths (0.3m - 1m) do not interfere with the transmission reliability or speed as much.

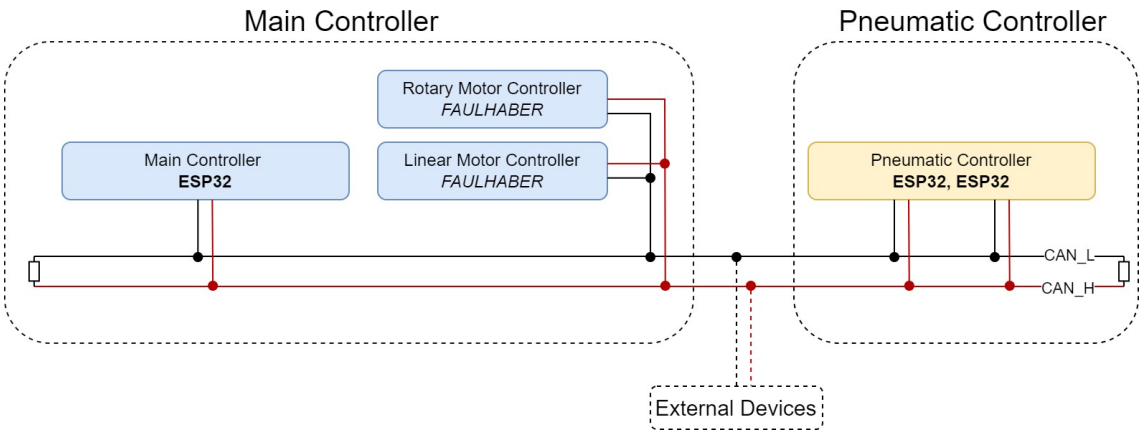


Figure 5.6: Redesigned CathBot CAN bus communication

5.2.3 Main controller

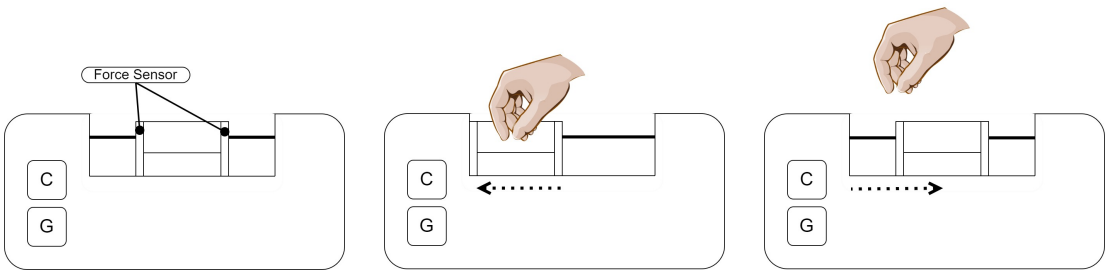
The software for the main controller ESP32 was redesigned from scratch. The original code was difficult to read, understand, and troubleshoot. To address these issues, significant changes were made, such as transitioning from C to C++ and adopting an object-oriented programming (OOP) approach. In OOP, abstraction is key. It's one of the four OOP pillars (Abstraction, Inheritance, Polymorphism, and Encapsulation), and abstraction aims to hide the implementation details and show only necessary information. It reduces the complexity of implementing the control loop of the controller and makes it easier to develop, and get an overview of the robot's inner workings.

To begin OOP, the following objects were identified:

- **ADC**: Analog-to-digital converter (ADC) uses a light sensor to check if a user is holding the manipulator of the main controller.
- **CAN**: CAN bus object uses the ESP32 built-in CAN bus to send and receive data.
- **GPIO**: The general-purpose input/output (GPIO) object is used to check the state of two buttons on the main controller, and also trigger the homing procedure of the controller.
- **MOTOR**: The motor object defines and abstracts all the motor controller communication, which allows for easy control and command of the two motors on the main controller.
- **CONTROL**: The control object orchestrates the behavior of the main controller. It uses all the objects above and makes them work together.

Control loop improvement
Aside from the OOP abstraction layer, implementing the main controller manipulator movement behavior was a major change. The manipulator's default home position is in the middle of its shaft. When the user moves the manipulator to any other position and releases it, it will automatically move back to the home position. This working principle is visualized in Figure 5.7.

In the original implementation, the movement of the manipulator was made possible by a virtual mass-damper model, which used force sensors embedded on each side of the manipulator to check which side the user wanted to move it (Figure 5.7 a)). Based on the force sensor data, a backward implicit Euler discretization was calculated to estimate the desired motor setpoint position. This made the device a bit unexpected; sometimes the estimation was off, and the manipulator could move unexpectedly and forcefully. An infinite autonomous movement cycle also happened occasionally, where the manipulator moved with high acceleration, triggered a force sensor reading, and then moved in the other direction, once again triggering a force sensor reading.



(a) Manipulator home position **(b)** User moved manipulator **(c)** Manipulator returns to home

Figure 5.7: Main controller working principle

Instead of using force sensors and estimation, a much simpler Proportional–integral–derivative (PID) based control was implemented. The FAULHABER motor controllers already have a

built-in PID control capability, which was used and tuned until a smooth and stable return-to-setpoint was achieved. Besides tuning the PID, implementing this in software was as simple as setting the setpoint to the home position once, and the motor controller did the rest of the work.

An added benefit is that all the parameters of the PID can be changed during runtime using CAN bus messages. This opens up the possibility for smooth haptic feedback in the form of PID motor controller current limits. High current limits make it difficult for the manipulator to move away from its setpoint position, while low current limits make the manipulator very light and easy to move. Besides the motor current limits, changing the PID tuning during runtime is not done and is not recommended as it can cause unexpected behavior.

Homing Procedure

The homing procedure for the main controller's manipulator was also simplified. During the homing, the limits of the manipulator's movement are determined so that the manipulator can be centered. In the original implementation, a CAN bus message had to be sent to the main controller to trigger the homing procedure. This, however, requires a designated device capable of sending the message (e.g., Raspberry Pi or a USB to CAN device).

Instead, the homing procedure is now triggered by the two physical buttons on the main controller (Catheter and Guidewire). It starts when both buttons are held down for 1.5 seconds.

CAN bus communication

As the CAN bus communication had to be rewritten, the main controller's CAN bus output was restructured and concatenated into a single message. The state of the main controller is in a standard 8-byte message, where each byte corresponds to a specific parameter, carrying the state of various components. An example of the CAN bus message is shown in Table 5.1 and the contents of the message are described in Table 5.2.

Table 5.1: 8-byte CAN bus message example from the main controller

Byte number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
CAN ID: 500	00	01	00	f7	08	64	08	00

Table 5.2: Main controller CAN bus output message description

Byte	**Description**	**Min**	**Max**
1	Grip Status	true (1)	false (0)
2	Guidewire Button	true (1)	false (0)
3	Catheter Button	true (1)	false (0)
4	Rotary Position	0	255
5	Rotary Scaling	0	100
6	Linear Position	0	200
7	Linear Scale	0	100
8	*Reserved*	-	-

The source component of CAN bus data and the limits are shown visually in Figure 5.8.

In addition to the state message, input CAN bus messages were implemented to change the behavior of the main controller during runtime. For instance, changing the current limits affects the resistance of the manipulator. The following messages can be sent to the main controller:

- Collision Detected

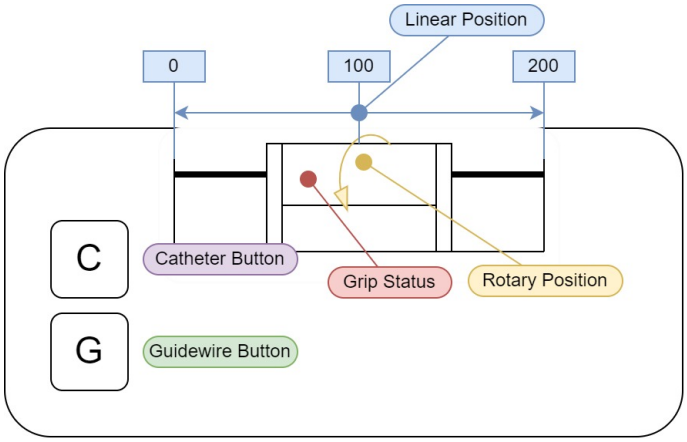


Figure 5.8: Main controller CAN bus message sources

  - Collision flag to toggle haptic feedback.
- Default Current Limit
  - Integer to change the current limit when the collision flag is inactive. The value can be set from 0 milliamperes (mA) to 300 mA, at which point the device becomes very heavy to move.
- Collision Current Limit
  - Integer to change the current limit when the collision flag is active. This value is intended to be slightly higher than the current default limit, so collisions will be intuitively noticeable.
- Linear and Rotary motor translation
  - Integers to change how much the MR-safe robot should move or rotate the devices corresponding to the main controller's output. By default, this value is set to 8, which means that after every 8 encoder values from the main controller, the pneumatic controller will do one step with the pneumatic stepper motors.

5.2.4 Pneumatic controller

The pneumatic controller's software suffered from the same issues as the main controller's original code. To keep the codebase consistent, it was also rewritten using a similar OOP approach. Additionally, several significant changes were made to simplify the control loop and make the device more predictable and stable.

The pneumatic controller has two embedded ESP32 microcontrollers, and the working principle is architecturally similar to the main controller, therefore very similar objects were created:

- **CAN**: CAN bus object uses the ESP32 built-in CAN bus to send and receive data.
- **GPIO**: The GPIO object initializes and opens up controls for the pneumatic valves, which send air to the MR-safe robot.
- **MOTOR**: The motor object defines and abstracts all the functionality (e.g., movement, rotation) for the air-valve controlled motors.
- **CONTROL**: Similarly to the main controller, the control object orchestrates the behavior of the pneumatic controller. It combines all the objects defined above to create the control loop of the controller.

The significant difference between main and pneumatic controllers is that the pneumatic has two ESP32 microcontrollers instead of one. This solution was chosen due to the limited availability of PWM channels for generating timed and shifted GPIO sequences for six motors. Each motor has 4 air valves designated for it, and a PWM timer controls each air valve. This requires

24 air valves, while the ESP32 has only 16 PWM channels. Custom PCBs have been designed for each ESP32, to connect the GPIO to the air valves. Due to this, the system must use two ESP32s, unless the PCBs are redesigned.

Moving pneumatic motors
However, besides the limitation of using two ESP32s, the software was made simpler by not using designated PWM for each valve but rather manually switching GPIO levels to high or low. While manually switching the GPIO could be considered a slower operation than using a PWM, the frequencies used for moving the pneumatic motors can not be any faster than 1/20 Hz due to the air tube lengths used to transfer the air. Therefore, technically the system could be simplified to use only a single ESP32 as they have 34 GPIO available.

Using manual GPIO switching, the controller became more flexible and reactive to the main controller's actions. Now, it was possible to map encoder values directly from the main controller's CAN bus message into GPIO states. For both the linear and rotary motor, a modulus of 4 from the encoder position results in the sequence required to make the pneumatic motor(s) move, as shown in the following code block:

```
switch (encoder_value % 4) {
    case 0: gpio_set_level(this->coil_a, 1); break;
    case 1: gpio_set_level(this->coil_b, 1); break;
    case 2: gpio_set_level(this->coil_a, 0); break;
    case 3: gpio_set_level(this->coil_b, 0); break;
    default: break;
}
```

To change the pneumatic motors translation or movement amount for encoder values, the "Rotary Scaling" and "Linear Scaling" parameters can be used (Table 5.2). The encoder value is divided by the scaling parameter, resulting in fewer encoder values to switch the GPIO.

Guidewire and catheter clamping
The original design had a flaw: the guidewire or catheter clamp would immediately close when the user grabbed the main controller manipulator. The clamps are designed to translate the endovascular devices forward or backward by imitating gripping by human fingers. However, when the clamp is closed, rotation of either endovascular device is impossible as they might strain and break. The clamps are shown in Figure 5.9.

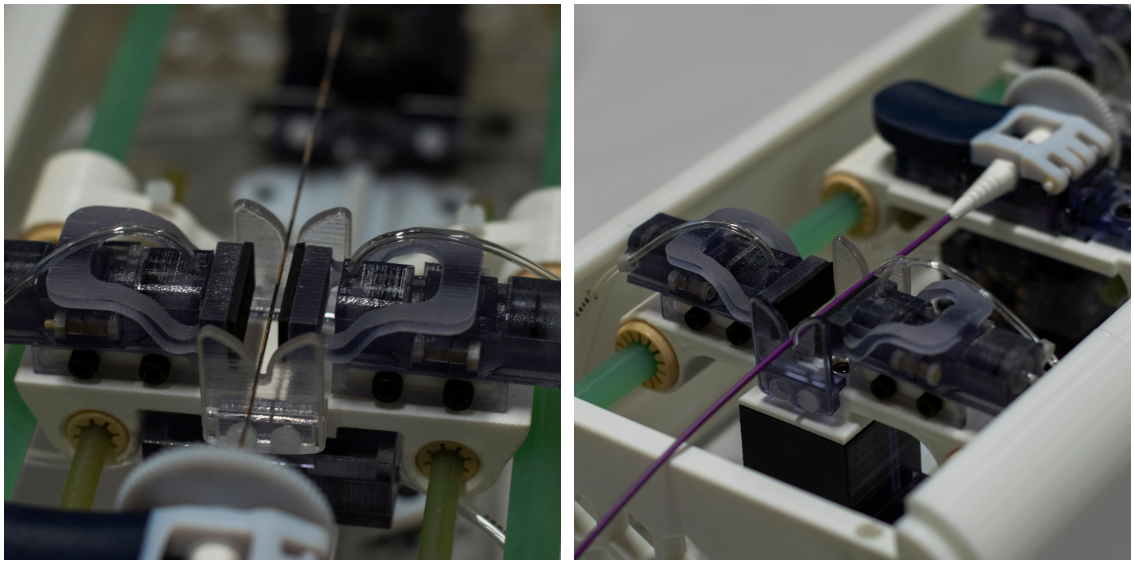
A clamp-closing limit was implemented to counter this, so the clamp is open when the linear motor is within ±3% of the home/center position. The clamp will close if the linear motor is moved more than 3%. This allows for static rotation of both devices, although rotation while moving is still unfeasible.

5.3 Integration with Tracking Software

This section describes how the designed software in Chapter 3 and Chapter 4 was integrated with the CathBot. The collision detection and haptic feedback implementation will be described.

5.3.1 Integration with Collision Detection

The collision detection functionality from the 3D Suite (discussed in Section 4.3) was integrated with the main controller using two predefined haptic resistance settings. One setting determines the controller's resistance without collisions, while the other adjusts the resistance during a collision. A standalone application, shown in Figure 5.10, was developed to modify these haptic feedback settings in real-time or prior to a trial. Additionally, this application can simulate collisions, enabling users to anticipate and prepare for actual collision scenarios.



(a) Guidewire clamp **(b)** Catheter clamp

Figure 5.9: CathBot MR-safe robot clamps

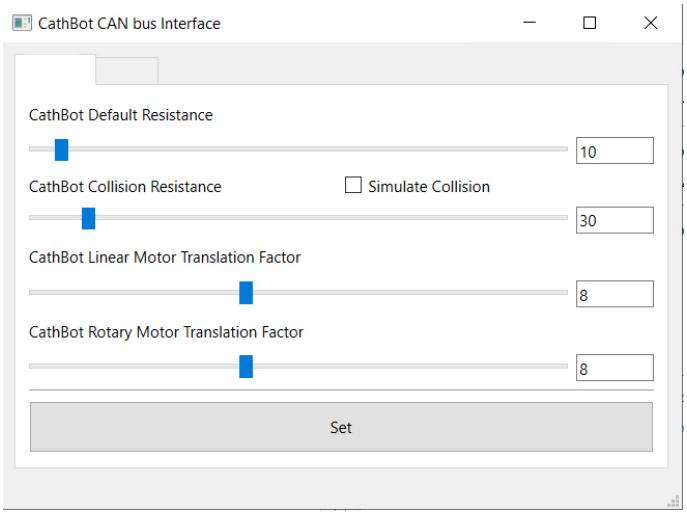


Figure 5.10: CathBot CAN bus interface, which allows changing and simulating the main controller's haptic feedback.

As mentioned in Section 4.3, 3D Suite periodically sends collision detection status to the main controller. This status is a boolean variable, determining the main controller's haptic feedback state. If a collision is detected, the main controller will immediately set its linear motor controller's current limit to the limit defined using the CAN bus interface. If a collision is no longer detected, the current limit will be switched back to the default resistance, also defined using the CAN bus interface.

A higher current limit will give the motor controller's PID more power to return to the home position. This added power will also make the main controller tangibly heavier to move away from the home position, imitating a collision of the guidewire, which would be slightly harder to move in the case of friction with a blood vessel.

A very heavy-to-move main controller would not be comfortable to use. Therefore, this haptic feedback aims to be as light as possible while subconsciously notifying the user that there may be some friction between the guidewire and the patient and that they should be more careful.

Too light resistance could also cause inaccuracies, as this is a robotic device, not an actual guidewire, and the pneumatic controller interprets any main controller movements. Therefore, the main controller's default resistance is adjustable, as it may feel too smooth and light for some users. A heavier default resistance could minimize the impact of involuntary human movements that would otherwise cause the endovascular device to move.

5.4 CathBot user study

The CathBot robotic platform with the collision detection software was evaluated in a user study with a diverse group of participants to assess its performance and usability. This section outlines the experimental setting, workflow, and feedback.

5.4.1 Experiment setup

The study was conducted at the UT and involved a qualified vascular surgeon (>30 years of experience), a technical physician (5 years of experience), and three students with no prior experience with endovascular operations.

The experiment was conducted on the abdominal aneurysm phantom using a RIM catheter (56535RIM, Merit Medical Systems Inc., South Jordan, Utah, United States) and the EPflex MR-safe guidewire. The task was to maneuver the guidewire from the left iliac artery across the abdominal aortic bifurcation and advance into the right iliac artery. The objective is further illustrated in Figure 5.11. The catheter and guidewire were initially placed so that both endovascular devices had to be manipulated using the robotic platform. The procedure was also chosen due to the high likelihood of collisions occurring, which would create haptic feedback.

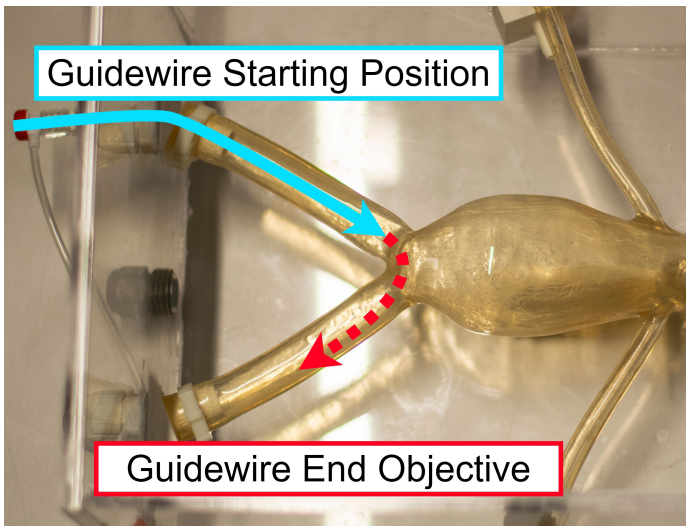


Figure 5.11: Phantom task to navigate guidewire from the left to right iliac artery.

The CathBot MR-safe robot with the endovascular devices and the phantom were positioned as shown in Figure 5.12. An example acquisition from the study is shown in Figure 5.13.

Before the study, the 3D GRE sequence was used to create an accurate 3D mesh of the phantom for collision detection.

Each participant was introduced to the robot, the software, and the task. Before the task, a dry run of less than 5 minutes with the main controller was conducted. During this time, the basic functionality of the main controller and haptic feedback capabilities were introduced so users would know what to expect. The haptic feedback was adjusted to a suitable range for each user.

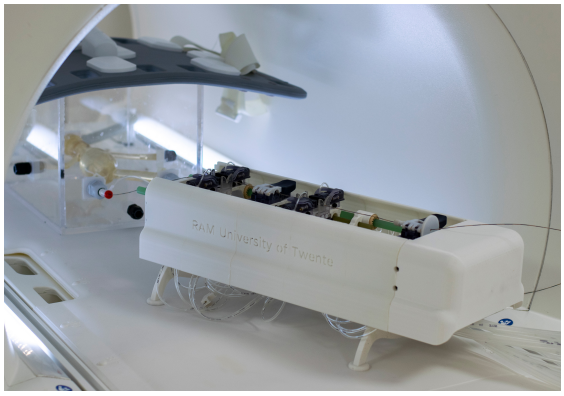


Figure 5.12: CathBot and phantom placement during the experiment.

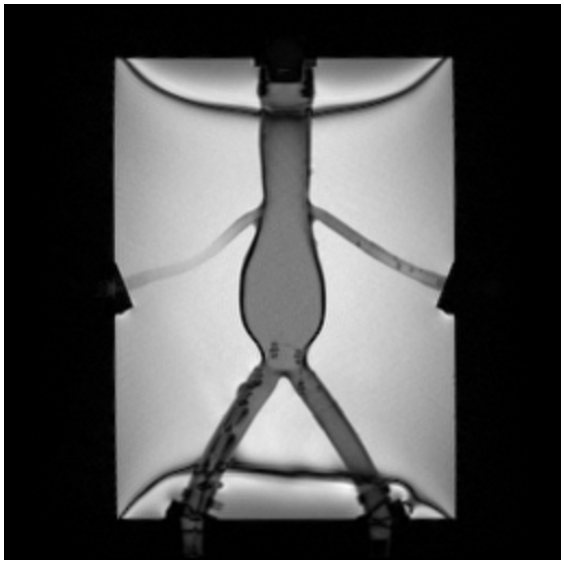


Figure 5.13: Example of an acquisition during the experiment.

Afterward, each subject conducted the task, for which 10 minutes were allocated. After this, some extra time was allowed to use the robotic system and try out other features, such as slice tracking. The robotic system's kinematics data and the Hub software data, including MR images, CNN detection, and guidewire tracking, were recorded throughout the task. Additionally, a screen recording was taken of the 3D Suite output.

Subsequent to participating in the study, subjects were asked to complete a survey consisting of 12 statements to evaluate their MR-guided robotic intervention task experience. The questionnaire consisted of the following statements, each with a 0 to 10 ordinal scale, ranging from rejection to agreement.

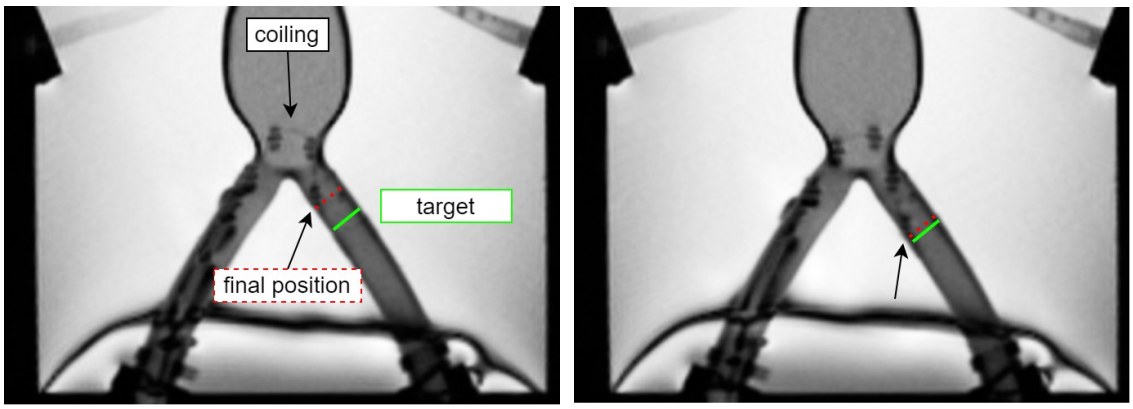
Survey
1. Your experience with endovascular procedures
2. The control system was precise.
3. The presence of haptic feedback was helpful in completing the task.
4. The guidewire detection software felt stable and predictable.
5. Guidewire tracking with MR slice movement was helpful.
6. I felt stressed, irritated, or annoyed while using this system during the task.
7. It was easy to control/use the system.
8. It was easy to make mistakes with the system.
9. Staying focused on what I was doing was easy.
10. The system was safe to use.
11. I would like to use the system again for this kind of task.
12. I would recommend this system to a colleague.

Table 5.3: Statements from the survey that each participant completed following the study.

5.4.2 Results

Assessment of the experimental procedure shows that all of the participants managed to advance the guidewire to the right iliac aorta. However, two out of five subjects advanced the guidewire far enough into the aorta, whereas others managed just to enter it. The guidewire

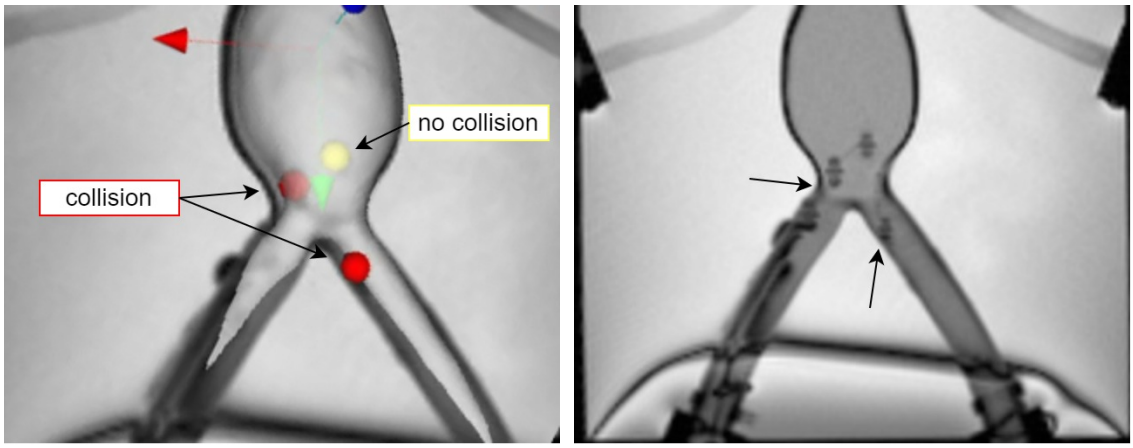
was not advanced very far due to guidewire coiling or CathBot's limited range. An example of both scenarios is shown in Figure 5.14.



(a) **(b)**

Figure 5.14: Example of an experiment where target distance a) was not reached and b) was reached. In the experiment a), the guidewire started coiling and was not advancing forward.

Each participant experienced haptic feedback at some point during the study. An example of detected collisions that created haptic feedback is shown in Figure 5.15.



(a) **(b)**

Figure 5.15: Example of collisions detected during the study. a) 3D Suite detecting two collisions between guidewire and phantom and b) The MR scanned image which is causing the collisions, detected collisions shown using arrows.

The results of the post-experimental survey are summarized in the Figure 5.16. The survey results show an overall positive-neutral outlook for the system. The main controller and MR-safe robot's precision were considered sufficient (Q2, 5.8 ± 2.4). The participants were moderately satisfied with the haptic feedback (Q3, 6.6 ± 2.8), while the guidewire detection could be improved (Q4, 5.2 ± 3.6). Responses to guidewire tracking were mixed (Q5, 5.4 ± 3). Subjects did not feel uneasy or distracted while using the robotic system (Q6, 1.8 ± 2; Q7, 6.4 ± 3.3; Q9, 7 ± 0.7). However, it was noted that mistakes with the system could be common (Q8, 5.4 ± 3.2). Despite this, users appreciated the safety aspects of the system (Q10, 7 ± 3.5).

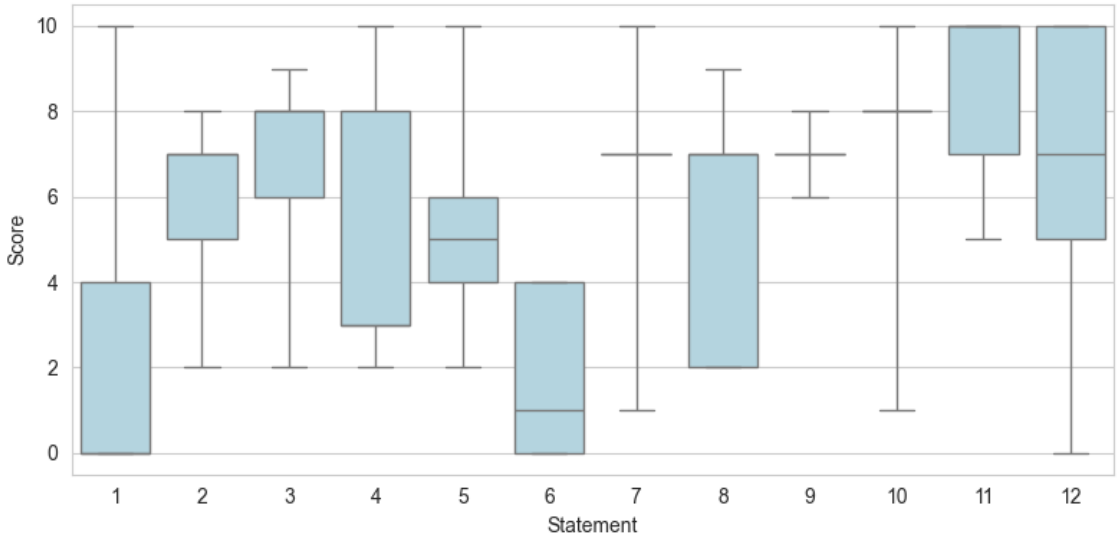


Figure 5.16: Post-experiment survey results. Statement number matches question number as shown in Table 5.3.

6 Discussion

This chapter discusses the findings and results presented in the project in relation to the objectives set out at the beginning of this study: **How to improve MRI-guided endovascular procedures**, by optimizing an interactive MRI sequence, detecting MR-safe passive guidewire, and integrating haptic feedback for the CathBot platform.

6.1 Optimized MRI sequence

The optimized bSSFP sequence described in Section 3.2 provides a moderately fast framerate, with an imaging rate at around 2 Hz, while retaining a good SNR and spatial resolution. While traditional X-ray imaging techniques supersede the achieved interactive MR framerate and spatial resolution, MR remains a viable alternative due to the excellent soft-tissue contrast and lack of ionizing radiation.

The optimized MR sequence aims to provide a solution for the "low spatial-temporal resolution" as highlighted in Chapter 1. This goal is successful, as the sequence has an optimized trade-off between spatial and temporal resolution for the MR scanner available at UT (1.5T MRI). Stronger MR scanners can acquire more SNR within the same scanning time and may optimize the sequence further by increasing the temporal resolution while keeping the spatial resolution as-is.

The choice between GRE and bSSFP was found to be heavily geared towards bSSFP due to its higher SNR efficiency and temporal resolution than GRE. However, bSSFP has two important limitations: banding artifacts caused by sensitivity to off-resonance and the transient approach to steady state, which causes a motion-flow effect for objects that move under the scanner. Due to these shortcomings, bSSFP was not used for the 3D model scanning stage, as the banding artifacts would be transferred into the 3D model and provide inaccurate representation. However, bSSFP remains the most suitable option for interactive real-time scanning as the imaging frequency and SNR outweigh the limitations.

6.2 Susceptibility simulation and detection using CNN

The susceptibility artefact simulation proved to be both accurate and flexible. The simulated susceptibility artefact signal achieved a very close similarity to the ground truth. The procedurally generated CNN training dataset was very effective, saving the effort of manually annotating and acquiring MR images experimentally. Procedural generation also allows the flexibility to include the artifact's appearance from oblique slice orientations into the dataset, allowing for detection at any angle instead of only sagittal and coronal.

The CNN trained on the procedurally generated dataset achieved a very high mean and DICE accuracy on the training data. This accuracy also translated into excellent real-time detection, with an evaluation DICE score of 0.69; false positives were low, and most markers were detected. The CNN evaluation, described in Section 3.6, showed that the CNN missed, on average, 1.3 markers from an image, which were confirmed to be markers by manual review. The most significant reason for missed detections was the transient effects caused by guidewire movement. These effects reduced the SNR of each marker and caused detection gaps in the CNN. Nijsink et al. had improved results with a manually annotated dataset and GRE sequence, where the median false positive detected markers was 0, and the CNN detected 90% of markers. The improved results could be due to the GRE sequence, which provides a clean signal even during guidewire movement or with flow patterns through the phantom. However, GRE sequences were not considered for detection in this project, due to the poor SNR and temporal resolution relation as described in Section 3.2.

The CNN proved sufficient for real-time detection, as the average prediction latency was around 0.23 seconds, while the MR sequence with optimal spatial resolution produced an image every 0.6 seconds. This leaves room for future improvement to allow 4 Hz or up to 5 Hz imaging when a stronger MR scanner is used.

6.2.1 Passive Guidewire tracking

The ad-hoc tracking algorithm demonstrated smooth performance under controlled test scenarios. During interactive imaging, the algorithm had mixed success. Sometimes, the MR slice would be correctly tracked to a marker, whereas mistakes and erroneous behavior were commonly experienced. This could be accounted for by the reactive nature of passive artefact tracking. With active MR tracking, the MR slice is moved predictively by running a very quick localizer sequence to determine the position of the guidewire tip in 3D space, followed by MR slice alignment and the normal sequence [21]. With passive tracking, localizing is not an option, and the MR slice can be moved only based on the historic guidewire movement data. Due to this, the tracking algorithm inherently introduces an extra frame delay into the system; if the guidewire is moved, the movement will be seen in the next acquisition, and the MR slice movement will be seen one more acquisition later.

Therefore, active tracking remains advantageous over passive tracking due to the ability to move the MR slice predictively. However, S. Patil's PhD described an experimental tracking technique using echo-dephased SSFP sequences, which allow localizing artefacts at the cost of an additional acquisition [19].

Comparing fluoroscopy-based tracking with MR tracking reveals a significant disadvantage for MRI. Fluoroscopy provides a continuous projection of the entire guidewire, offering real-time visualization of its shape and path, as demonstrated by P. Wang et al. [47]. In contrast, MR tracking relies on discrete markers to estimate the guidewire's trajectory and shape, which can result in less accurate visualization and increased tracking errors.

This project attempted to estimate the guidewire shape using the detected markers. However, this approach was quickly abandoned due to instability caused by occasional false positive CNN detections, which resulted in incorrect guidewire visualization. As false positives can be commonly detected, such visualizations were not attempted further due to the project's time constraints. It was concluded that displaying the raw detection data is preferable to showing an inaccurate guidewire path until a more stable visualization is developed. The initial visualization attempts are shown in Figure A.2. An accurately visualized guidewire based on CNN prediction would require further elaboration. A possible simplification for visualization is to follow Nano4Imaging's TRACKR AI, which, based on published visualizations on YouTube [48], uses a specific guidewire which has passive markers only at the tip, and therefore only the tip of the guidewire will be tracked and visualized.

6.2.2 3D-model based Haptic feedback

Implementing haptic feedback based on real-time collision data between detected susceptibility artefacts and a pre-scanned 3D model of the phantom was successful. The developed 3D Suite accurately represented the 3D position of the MR slice and the phantom model in real-time. All the detected artefacts were shown in the 3D Suite, with alternating colors depending on whether a collision was detected. Collision detection based on polygon data can be considered a viable way to introduce haptic feedback into a medical robot; however, this project used a static phantom, which does not move like an in-vivo patient would (e.g., breathing or heartbeat). Generating haptic feedback when using this system on in-vivo patients would cause inaccuracies, as even the slightest movement will cause the pre-scanned 3D model to become misaligned.

Scanning the 3D model of the phantom was an uncomplicated process, as preexisting tools such as 3D Slicer can process the MR data quickly and accurately. However, the phantom is a simplified model of the aneurysm without any background signal, and achieving similar 3D model accuracy with an in-vivo patient would be more complicated.

Commercially available robotic endovascular systems, none of which are MR-compatible, many lack haptic feedback so that the vessel/guidewire contact forces are not transmitted to the surgeon [49]. Systems that implement haptic feedback have been able to do so by embedding force sensors at the distal portion of their catheter, such as Sensei X2 [11]. However, embedding such force sensors in MR-safe robotic systems can introduce complications, so a vision-based solution is the most appropriate. Early developments with the CathBot platform demonstrated the benefit of continuous fluoroscopy vision-based haptic feedback, which resulted in a 70% decrease in mean force and 61% decrease in maximum force when haptic feedback was provided [49]. This project adapted the CathBot system to use interactive MR imaging instead, while providing vision-based haptic feedback.

6.3 CathBot platform feasibility

The CathBot platform proposes intuitive instrumentation for remote-controlled endovascular procedures. The redesigned software successfully addressed several pre-existing issues with the robot. The main controller is smooth and easy to use, and the implemented haptic feedback is easily noticeable and highly configurable in strength. The pneumatic controller's software was addressed to align with the main controller's software architecture. However, the functionality remained the same as the original, besides allowing simultaneous guidewire and catheter manipulation.

Experimental procedure results show that the CathBot platform is considered a great first concept. However, reliability, precision, and visualization could be improved. The most significant issue raised during experiments was that part of the guidewire was outside the MR slice, making it very hard to understand what movements should be made with the robot. It was also noted that mistakes were easy to make with the system, related to the guidewire outside of the slice and the latency between moving the robot and seeing the movement on the screen. The presence of haptic feedback was considered positive, while the guidewire tracking was generally noted as unpredictable. The detection and tracking could be significantly improved once the CNN is trained with a dataset that detects artefacts with slight transient effects or motion blur. This would improve the stability and predictability of the CNN during an interactive bSSFP sequence.

The MR-safe robot prototype has two significant design limitations: 1) finite instrument manipulation strokes, and 2) rotation and translation combined movement is not supported. Finite instrumentation can cause the MR-safe robot to reach its range of motion and prevent the guidewire or catheter from moving forward. Individual rotation and translation of the devices were noted as a shortcoming, as it is common practice to rotate and push the guidewire together.

7 Conclusion

The development and evaluation of the CathBot robotic platform for MRI-guided endovascular interventions highlight advancements in hardware and software integration. This project successfully addressed several critical challenges in MRI-based interventional procedures, particularly optimizing MRI sequences, improving guidewire detection, and enhancing procedural safety through a combination of CathBot and software solutions.

The optimized MRI sequence, which balances spatial and temporal resolution, demonstrated feasibility for real-time endovascular procedures. While MRI cannot yet match the temporal resolution of traditional X-ray imaging, it offers superior soft-tissue contrast and avoids ionizing radiation, making it a viable alternative for interventional clinical applications. A balanced steady-state free precession (bSSFP) sequence provided the best compromise between imaging speed and quality, although further advancements in MR scanner technology or experiments with alternative MR sequences could improve these capabilities.

The convolutional neural network (CNN) trained for detecting susceptibility artefacts showed promising results in real-time applications. The procedurally generated training dataset enabled the CNN to detect artefacts with reasonable accuracy and speed. However, the performance could be further improved by incorporating transient effects into the training data to improve the robustness of marker detection under dynamic conditions.

Integrating haptic feedback into the CathBot platform improved user experience and procedural safety. The feedback system, which alerts the user of potential collisions through tangible resistance changes, demonstrated its potential to increase procedural safety while using the CathBot robotic platform. However, the platform's design limitations, such as finite instrument manipulation strokes and the inability to support combined rotation and translation, were identified as areas needing further refinement.

The user study conducted with the CathBot platform provided valuable insights into its practical application and usability. While the feedback was generally positive, indicating the system's potential for clinical use, issues like guidewire tracking reliability and marker detection stability were noted. Addressing these issues in future iterations could enhance the system's precision, safety, and user confidence.

In conclusion, the CathBot platform with haptic feedback capabilities represents a step forward in the field of MRI-guided endovascular interventions. The project achieved its primary objectives, demonstrating the feasibility and benefits of integrating MR imaging with robotic assistance. Future work should focus on refining the hardware design, improving CNN training datasets, and enhancing real-time imaging capabilities to fully realize the potential of this approach to minimally invasive surgery.

7.1 Future Work

This section highlights what could be improved with the developed and existing systems involved in this project.

7.1.1 Susceptibility simulation and detection

The transient effect augmentation should be incorporated into the training data to improve the shortcomings of missed detections during guidewire movement. Transient effects are visually very similar to linear motion blur, and adding the motion blur to individual artefacts within the training data could make the CNN prediction much more stable.

Additionally, to support oblique MR slice orientation, the susceptibility artefact simulation should be expanded to simulate artefact appearance from oblique slice orientations, and these simulations should be included as part of the dataset augmentation.

7.1.2 Guidewire tracking and visualization

The current system's most significant limitation is tracking susceptibility artefacts only on two axes. If the guidewire could be tracked on all three axes, much better visualization of its 3D position would be possible, for instance, showing its position in 3D space in reference to the scanned model. Three-dimensional tracking can be complicated to implement and might reduce the temporal resolution, as extra scans must be taken. For instance, following the experimental method in [19], an echo-dephased SSFP sequence could be used to locate the artefacts. An alternative method would be to create a custom sequence that scans multiple slice orientations simultaneously. For instance, the first scan has a coronal orientation, while the second scan has a sagittal orientation, and this sequence will repeat. However, temporal resolution will be halved, as multiple scans must be taken. Using such a sequence with the CNN detection would provide three-dimensional data of the guidewire's position and allow for improved tracking and MR slice movement.

In the current implementation, the developed 3D Suite uses the VTK toolkit to present 3D information to the CathBot user. Instead of VTK, a more common and flexible platform could be used, such as Unity or Unreal Engine. This platform switch would also open the possibility of integrating the CathBot with VR headsets or controllers.

7.1.3 CathBot platform

To improve the CathBot platform, the device's linear translation and rotation should be made possible concurrently. With the current design, linear translation requires a pneumatic clamp to close and grab the device on the MR-safe robot. Closing the clamp means that rotation is impossible while the device is clamped.

An additional improvement to the MR-safe robot is to improve the limited forward movement range. CathBot can move either device around 10 cm forward with the current design. This is quite a severe limitation; a design without a limited movement range would be much more convenient.

The existing design of the CathBot main controller requires repetitive movements that aim to replicate the manual pulling and pushing of the guidewire or catheter, to utilize the surgeon's trained dexterity with manual intervention. A proposed improvement is to streamline these movements, stabilizing the robotic system and making it more predictable and easier to use. To achieve this, instead of repetitively advancing the main controller's manipulator forward or backward, the manipulator would be moved forward slightly and held at that position. While the manipulator is not in the center position, the MR-safe robot would continuously move the endovascular devices forward or backward. This modification would likely lower the learning curve and simplify operations. However, it is important to consider that such simplification might also diminish the use of advanced manual skills that surgeons acquire through traditional procedures, as the robot would no longer mimic these movements.

A Appendix

A.1 Susceptibility Simulation implementation

With Python and Numpy, Equation 3.1 can be implemented using a mesh grid approach. First, the simulation limits are defined:

```
x = np.linspace(-FOV.x, FOV.y, SIMULATION_RESOLUTION)
y = np.linspace(-FOV.x, FOV.y, SIMULATION_RESOLUTION)
z = np.linspace(-FOV.x, FOV.y, SIMULATION_RESOLUTION)
X, Y, Z = np.meshgrid(x, y, z)
```

Second, the mesh grid can be plugged into the Equation 3.1:

```
c = B0 * delta_x_V / (4 * np.pi)
Bz = c * (X**2 + Y**2 - 2 * Z**2) /
     ((X**2 + Y**2 + Z**2)**(5 / 2))
```

Equation 3.2 integration with the composite trapezoidal rule is implemented as follows:

```
S_axial = 1 / ST * np.trapz(P_XYZ *
          np.exp(-1j * GR * Bz * TE), Z, axis=2)
S_coronal = 1 / ST * np.trapz(P_XYZ *
          np.exp(-1j * GR * Bz * TE), Y, axis=0)

ST: Slice Thickness
GR: Gyromagnetic Ratio
```

A.2 Tracking algorithm UML overview

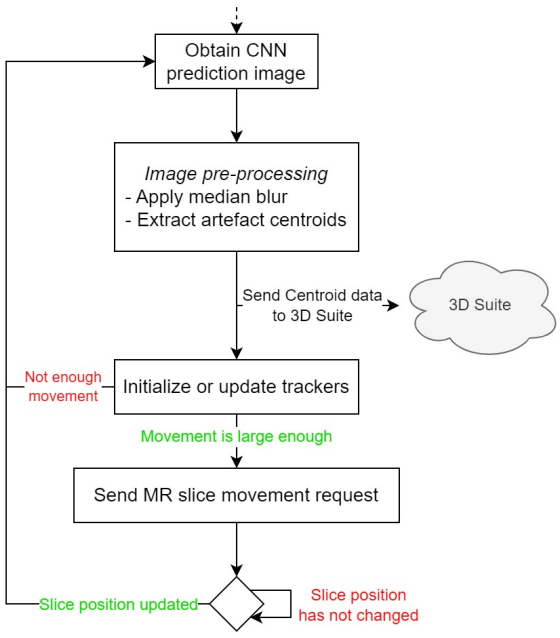
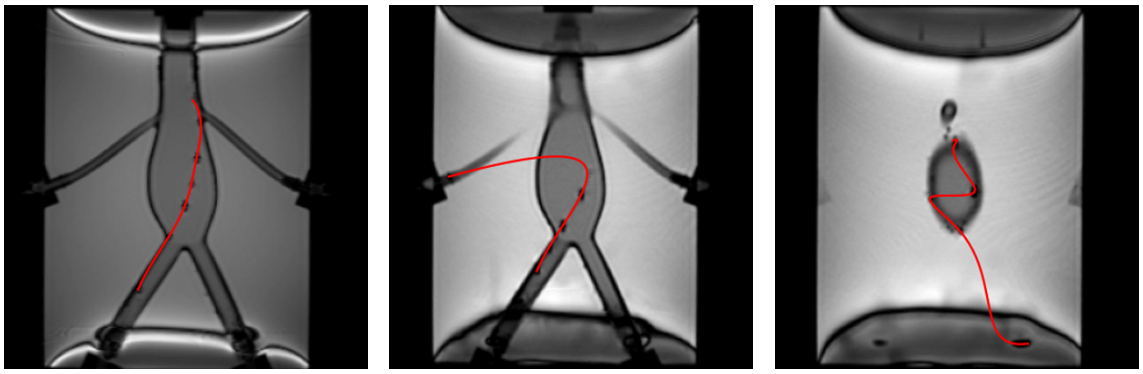


Figure A.1: Tracking algorithm graphical overview.

A.3 Guidewire path visualization

Guidewire visualization: if more than one marker is detected, the centroids of the CNN detection are used to create a spline. This was attempted by finding the B-spline representation of an N-D curve, using the `scipy.interpolate.splprep` function.

It works well under controlled and predictable scenarios. However, the output becomes incorrect and unpredictable when the data contains false positives or the guidewire is coiling.



(a) **(b)** **(c)**

Figure A.2: Guidewire shape and path estimation attempts using spline; a) successful estimation, b) and c) show incorrect estimations.

Bibliography

[1] Union O. Health at a Glance: Europe 2022. OECD Publishing; 2022.

[2] Saeed M, Hetts SW, English J, Wilson M. MR fluoroscopy in vascular and cardiac interventions (review). Int J Cardiovasc Imaging. 2012 Jan;28(1):117-37.

[3] Saeed M, Henk CB, Weber O, Martin A, Wilson M, Shunk K, et al. Delivery and assessment of endovascular stents to repair aortic coarctation using MR and X-ray imaging. J Magn Reson Imaging. 2006 Aug;24(2):371-8.

[4] Nijsink H, Overduin CG, Willems LH, é MC, Fütterer JJ. Current State of MRI-Guided Endovascular Arterial Interventions: A Systematic Review of Preclinical and Clinical Studies. J Magn Reson Imaging. 2022 Nov;56(5):1322-42.

[5] Ramai D, Zakhia K, Etienne D, Reddy M, Bozzini P. Philipp Bozzini (1773-1809): The earliest description of endoscopy. J Med Biogr. 2018 May;26(2):137-41.

[6] Fortuna L, Ginoya T, Maddahi Y, Zareinia K. A Historical Review of Medical Robotic Platforms. Journal of Robotics. 2021;2021:6640031. Available from: https://doi.org/10.1155/2021/6640031.

[7] Simorov A, Otte RS, Kopietz CM, Oleynikov D. Review of surgical robotics user interface: what is the best way to control robotic surgery? Surg Endosc. 2012 Aug;26(8):2117-25.

[8] Surgical I. Da Vinci Surgical System; 2024. Accessed: 2024-06-02. https://www.intuitive.com/en-us/products-and-services/da-vinci/5.

[9] Bergholz M, Ferle M, Weber BM. The benefits of haptic feedback in robot assisted surgery and their moderators: a meta-analysis. Scientific Reports. 2023 11;13(1):19215. Available from: https://doi.org/10.1038/s41598-023-46641-8.

[10] Peters BS, Armijo PR, Krause C, Choudhury SA, Oleynikov D. Review of emerging surgical robotic technology. Surg Endosc. 2018 Apr;32(4):1636-55.

[11] Yakoubi R, Hillyer S, Haber GP. In: Robotic Urological Surgery: State of the Art and Future Perspectives; 2012. .

[12] Groenhuis V. Robotic systems for breast biopsy using MRI and ultrasound imaging: Optimal guidance to target lesion in deformable tissue [PhD Thesis - Research UT, graduation UT]. Netherlands: University of Twente; 2020.

[13] Abdelaziz MEMK, Kundrat D, Pupillo M, Dagnino G, Kwok TMY, Chi W, et al. Toward a Versatile Robotic Platform for Fluoroscopy and MRI-Guided Endovascular Interventions: A Pre-Clinical Study. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2019. p. 5411-8.

[14] Kundrat D, Dagnino G, Kwok TMY, Abdelaziz MEMK, Chi W, Nguyen A, et al. An MR-Safe Endovascular Robotic Platform: Design, Control, and Ex-Vivo Evaluation. IEEE Transactions on Biomedical Engineering. 2021;68(10):3110-21.

[15] Nayak KS, Lim Y, Campbell-Washburn AE, Steeden J. Real-Time Magnetic Resonance Imaging. J Magn Reson Imaging. 2022 Jan;55(1):81-99.

[16] Nijsink H, Overduin CG, Brand P, De Jong SF, Borm PJA, é MC, et al. Optimised passive marker device visibility and automatic marker detection for 3-T MRI-guided endovascular interventions: a pulsatile flow phantom study. Eur Radiol Exp. 2022 Feb;6(1):11.

[17] Bos C, Viergever MA, Bakker CJ. On the artifact of a subvoxel susceptibility deviation in spoiled gradient-echo imaging. Magn Reson Med. 2003 Aug;50(2):400-4.

[18] deke KM, schmann P, Tischler R. Susceptibility artefacts in NMR imaging. Magn Reson Imaging. 1985;3(4):329-43.

[19] Patil S. Passive Tracking and System Interfaces for Interventional MRI [PhD Thesis]; 2009.

[20] Bijlsma J, Kundrat D, Dagnino G. MR-based navigation for robot-assisted endovascular procedures. International Journal of Intelligent Robotics and Applications. 2024 April. Available from: https://doi.org/10.1007/s41315-024-00340-3.

[21] George AK, Derbyshire JA, Saybasili H, Saikus CE, Kocaturk O, Guttman MA, et al. Visualization of active devices and automatic slice repositioning ("SnapTo") for MRI-guided interventions. Magnetic Resonance in Medicine. 2010;63(4):1070-9. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.22307.

[22] Bijlsma J. MR Navigation Framework for Robot-Assisted Endovascular Intervention [MSc assignment]. University of Twente; 2021.

[23] Nano4Imaging MR guidewires;. Accessed 05.03.2024. Available from: https://www.nano4imaging.com/.

[24] Sethi RK, Henry AJ, Hevelone ND, Lipsitz SR, Belkin M, Nguyen LL. Impact of hospital market competition on endovascular aneurysm repair adoption and outcomes. J Vasc Surg. 2013 Sep;58(3):596-606.

[25] Sammet S. Magnetic resonance safety. Abdom Radiol (NY). 2016 Mar;41(3):444-51.

[26] Kodali S, Baher A, Shah D. Safety of MRIs in patients with pacemakers and defibrillators. Methodist Debakey Cardiovasc J. 2013;9(3):137-41.

[27] Hailey D. Open magnetic resonance imaging (MRI) scanners. Issues Emerg Health Technol. 2006 Nov;(92):1-4.

[28] Mahony NO, Campbell S, Carvalho A, Harapanahalli S, Velasco-Hernández GA, Krpalkova L, et al. Deep Learning vs. Traditional Computer Vision. CoRR. 2019;abs/1910.13796. Available from: http://arxiv.org/abs/1910.13796.

[29] Chen Q, Wu R. CNN Is All You Need. CoRR. 2017;abs/1712.09662. Available from: http://arxiv.org/abs/1712.09662.

[30] Isensee F, Jaeger PF, Kohl SA, Petersen J, Maier-Hein KH. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. Nature methods. 2021;18(2):203-11.

[31] Isensee F, Jaeger PF, Kohl SA, Petersen J, Maier-Hein KH. nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation; 2021. Accessed: April 15, 2024. https://github.com/MIC-DKFZ/nnUNet.

[32] Brandão D. Choosing the Right Guidewire: The Key for a Successful Revascularization. In: Saiz-Sapena N, Aparici-Robles F, Tsoulfas G, editors. Art and Challenges Involved in the Treatment of Ischaemic Damage. Rijeka: IntechOpen; 2022. Available from: https://doi.org/10.5772/intechopen.104484.

[33] Basar B, Sonmez M, Yildirim DK, Paul R, Herzka DA, Kocaturk O, et al. T MRI system. Magn Reson Imaging. 2021 Apr;77:14-20.

[34] Settecase F, Martin AJ, Lillaney P, Losey A, Hetts SW. Magnetic Resonance-Guided Passive Catheter Tracking for Endovascular Therapy. Magn Reson Imaging Clin N Am. 2015 Nov;23(4):591-605.

[35] epflex. epflex;. Accessed: Thursday 20th June, 2024. https://www.epflex.com/en/products-2/mrline/.

[36] Riederer SJ, Tasciyan T, Farzaneh F, Lee JN, Wright RC, Herfkens RJ. MR fluoroscopy: technical feasibility. Magn Reson Med. 1988 Sep;8(1):1-15.

[37] Bock M, ller S, Zuehlsdorff S, Speier P, Fink C, Hallscheidt P, et al. Active catheter tracking using parallel MRI and real-time image reconstruction. Magn Reson Med. 2006 Jun;55(6):1454-9.

[38] Seppenwoolde JH, Viergever MA, Bakker CJ. Passive tracking exploiting local signal conservation: the white marker phenomenon. Magn Reson Med. 2003 Oct;50(4):784-90.

[39] Bernard O, Lalande A, Zotti C, Cervenansky F, Yang X, Heng PA, et al. Deep Learning Techniques for Automatic MRI Cardiac Multi-Structures Segmentation and Diagnosis: Is the Problem Solved? IEEE Trans Med Imaging. 2018 Nov;37(11):2514-25.

[40] Dice LR. Measures of the amount of ecologic association between species. Ecology. 1945;26(3):297-302.

[41] Braun HJ, Gold GE. Advanced MRI of articular cartilage. Imaging Med. 2011 Oct;3(5):541-55.

[42] GmbH SH. Access-i Developer Guide; 2021. Classification "Restricted".

[43] Cipra T, Romera R. Kalman filter with outliers and missing observations. TEST: An Official Journal of the Spanish Society of Statistics and Operations Research. 1997 02;6:379-95.

[44] Slicer . 3D Slicer;. Accessed: Thursday 20th June, 2024. https://www.slicer.org/.

[45] VTK. The Visualization Toolkit;. Accessed: Thursday 20th June, 2024. https://vtk.org/.

[46] Corrigan S. Controller Area Network Physical Layer Requirements. Application Report. 2008 January;(SLLA270).

[47] Wang P, Chen T, Zhu Y, Zhang W, Zhou SK, Comaniciu D. Robust guidewire tracking in fluoroscopy. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009. p. 691-8.

[48] Nano4Imaging. TRACKR AI software: Tracking passive guidewires; 2023. Accessed: 2024-06-09. Available from: https://www.youtube.com/watch?v=G-Zx751wyGU.

[49] Dagnino G, Liu J, Abdelaziz MEMK, Chi W, Riga C, Yang GZ. Haptic Feedback and Dynamic Active Constraints for Robot-Assisted Endovascular Catheterization. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2018. p. 1770-5.