POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering

Master of Science Thesis

Design and development of a real-time localisation algorithm for the magnetic manipulation of the Magnetic Flexible Endoscope.

Supervisors

Professor Alessandro Rizzo, Politecnico di Torino - IT

Dr. Bruno Scaglioni, University of Leeds - UK

Professor Pietro Valdastri, University of Leeds - UK

Candidate

Luca Audino

Abstract

Colorectal cancer (CRC) is the third leading cause of cancer related death worldwide, and its incidence is constantly rising in developing nations. It usually starts as a benign tumour in the form of a gastrointestinal polyp, which becomes cancerous over time. Colonoscopy is the most common recommended screening procedure. During this exam, a semi-rigid endoscope with diagnostic and therapeutic capabilities is inserted in the colon via the rectum and pushed forward through the large intestine by the physician. However, owing to the method of actuation, traditional colonoscopy often requires long training sessions for the physician and causes patient discomfort. In the past twenty years, a wide range of tethered and wireless devices have been developed to mitigate these limitations, so to decrease the risk associated with the colonoscopy. In this area of research, magnetically actuated and controlled mesoscale capsules have shown to have the potential both to reduce the pain of the patient and simplify the colonoscopy procedure, so to revolutionize GI endoscopy. Among these devices, the Magnetic Flexible Endoscope (MFE), designed and developed by the Science and Technologies Of Robotics in Medicine (STORM) Lab UK and USA, uses a purpose-built magnetic system mounted on the end-effector of a robotic manipulator to control a tethered capsule endoscope. To apply the necessary forces and torques to the endoscopic tip and to enable a precise polyps' detection, the accurate real-time pose estimation of the device is crucial. Two generations of localization algorithms have been developed. The current version solves the singularity problems faced by the first generation but, being based on the particle filter, is computationally cumbersome. This last feature represents a constraint for real-time performance.

The work presented in this Master Thesis is focused on the design of a new localisation algorithm for the MFE based on Kalman Filters. A preliminary research is carried out to identify the most suitable version according to the nonlinear mathematical models used for the prediction of the capsule pose. Special attention is given to the Kalman Filter (KF), Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). This preliminary study highlights the impossibility to use traditional EKF and UKF with the currently used magnetic field models and suggests a feasible algorithm based on a Kalman Filter. The estimation method developed hereby represents a proof of concept for the application of this widely used estimator to achieve robust and real-time performances with a lower computational effort. Although static and dynamic tests have been performed to verify the correctness of the algorithm, this localisation method needs to be optimized on the robotic platform. In fact, due to the containment measures against the spread of the new coronavirus COVID-19, the physical Magnetic Flexible Endoscope has not been employed in the project. To overcome this limitation, a detailed ROS simulator of the MFE has been developed in Python and used for the design of the new localisation algorithm. The main target of the simulator, which involves the use of an IMU module and an STM32 Nucleo board, is to generate at 100Hz a set of inertial and magnetic field data as similar as possible to the ones retrieved by the sensors embedded in the physical capsule.

Sommario

Il Carcinoma del Colon-Retto (CRC) è ad oggi la terza causa di morte per cancro a livello globale e la sua incidenza è in costante aumento nei paesi in maggior via di sviluppo. Il CRC è solito manifestarsi nelle sue prime fasi sotto forma di polipo intestinale, che muta in tumore con il passare del tempo. La colonscopia è la procedura di screening e prevenzione ad oggi più utilizzata e raccomandata dai medici. Durante tale esame, un endoscopio semirigido, con capacità sia diagnostiche che terapeutiche, viene posto in prossimità del retto e fatto avanzare lungo il colon grazie ad un apposito sistema di teleoperazione manuale. Tuttavia, a causa della tecnica di avanzamento forzato da parte dell'operatore, gli attuali endoscopi causano frequentemente grossi disagi al paziente e introducono lunghe sessioni di formazione per il chirurgo. Al fine di diminuire il rischio associato alla colonscopia, negli ultimi vent'anni è stata sviluppata un'ampia gamma di dispositivi endoscopici innovativi, sia dotati di connettori che wireless. Particolare attenzione è stata rivolta alle capsule endoscopiche che, venendo controllate da remoto tramite accoppiamento magnetico, hanno dimostrato di avere il potenziale per rivoluzionare l'endoscopia gastrointestinale diminuendo il dolore provocato al paziente e semplificando questa procedura di screening e diagnosi. Tra questi dispositivi, l'Endoscopio Flessibile Magnetico (MFE), progettato e sviluppato dal Science and Technologies Of Robotics in Medicine (STORM) Lab UK e USA, utilizza un apposito sistema magnetico, fissato sull'end-effector di un manipolatore robotico, per controllare e direzionare una capsula endoscopica dotata di connettore. Per garantire una corretta applicazione delle forze e coppie necessarie all'attuazione e per consentire una mappatura ad alta precisione degli eventuali polipi intestinali, un'accurata stima in tempo reale della posa del dispositivo riveste un ruolo di fondamentale importanza. Sebbene l'algoritmo di localizzazione attualmente implementato risolva importanti problemi riscontrati nelle sue versioni precedenti (singolarità del campo elettromagnetico e inizializzazione dell'orientazione della capsula), un elevato sforzo computazionale è richiesto a causa dell'approccio basato su un Particle Filter. Il lavoro presentato in questa tesi di laurea è incentrato sulla progettazione di un nuovo algoritmo di localizzazione per l'MFE basato sui filtri di Kalman. Date le molteplici implementazioni di questi stimatori e i modelli matematici utilizzati per predire posizione e orientazione della capsula endoscopica, la fase di progettazione è stata anticipata da una ricerca volta ad identificare la tipologia di filtro più adatta. Particolare attenzione è stata rivolta al Kalman Filter (KF), Extended Kalman Filter (EKF) e Unscented Kalman Filter (UKF). Tale studio ha messo in evidenzia l'impossibilità di utilizzare l'EKF e lo UKF con i modelli di campo magnetico attualmente utilizzati e ha quindi suggerito di basare il nuovo algoritmo di localizzazione su un Kalman Filter.

Il lavoro qui riportato è volto a dimostrare la validità del Kalman Filter per lo sviluppo di una nuova localizzazione robusta contro i disturbi elettromagnetici e adeguata ad applicazioni real-time. E' bene però notare che le misure di contenimento contro la diffusione di COVID-19 hanno impedito di testare il nuovo algoritmo di localizzazione sull'Endoscopio Magnetico Flessibile. Per far fronte a tale limitazione, un simulatore ROS della MFE è stato sviluppato in Python e utilizzato in fase di progettazione e durante la validazione dell'algoritmo. Il simulatore, basato su una IMU e una scheda di sviluppo STM32 Nucleo, garantisce la generazione di valori di campo magnetico, accelerazione e velocità angolare il più possibile simili a quelli misurati dai sensori incorporati nella capsula endoscopica.

Contents

Chapter 1

Introduction

The work presented hereby is focused on the design and development of a new localisation algorithm for the Magnetic Flexible Endoscope developed by the STORM Lab UK and STORM Lab USA. This Master Thesis aims to provide the reader a clear description of the innovative robotic platform with main attention toward its magnetic localisation algorithm; to do so, the work has been divided into five chapters.

In *Chapter 1*, an overview of the traditional endoscopy is presented to the reader with particular attention on the colonoscopy procedure and its main drawbacks. Then, the state of the art of the robotic endoscopy is presented as a possible solution to overcome the limitations of the traditional colonoscopy.

Chapter 2 is focused on the Magnetic Flexible Endoscope and provides the reader with a detailed description of the hardware and software environment of the robotic platform. Coherently with the topic of this work, the last section presents in a chronological order all the algorithms for magnetic localisation that have been developed by the STORM Lab. For each of them, the set-up, the algorithm, the results and the limitations are reported in detail. A prior presentation of the state of the art technologies is also reported for completeness.

Chapter 3 explains the need for the development of a ROS simulator of the Magnetic Flexible Endoscope and describes in details the hardware components, the ROS environment and the developed Python code. Given the high degree of similarity with the real platform, this chapter can be useful to deepen the working principles of the magnetic localisation of the MFE.

Chapter 4 shows the preliminary research that has been carried out to choose the most appropriate estimator for the new localisation algorithm. Finally, the impossibility to use an Unscented Kalman Filter is motivated and explained.

In *Chapter 5*, a description of the new localisation algorithm based on a Kalman Filter is provided to the the reader. Moreover, the results of static and dynamic tests are reported for the validation of the new method for the estimation of the pose.

Moving to the *Appendices*, *Appendix A* contains parts of the developed Arduino code for the MFE simulator. As for *Appendix B*, it shows pieces of the code developed for the STM32 Nucleo Board: the *main* function used to initialized the parameters of the IMU and the *endless loop* that is used to calibrate the gyro and publish the data through serial port. Then the Visual Basic Code developed by Derby *et al.* [5] *et al.* and used for the closed form solution of the elliptic integral is shown in *Appendix C*. *Appendix*

D is instead focused on the code developed for the MFE Simulator: although only few functions are reported, a line-by-line description of the Python file is provided to the reader. The python code can be found in the GitHub repository of the STORM Lab. Finally, *Appendix E* presents some functions of the new localisation algorithm based on a Kalman Filter; as before, the whole Python file can be found in the GitHub repository of the STORM Lab.

1.1 Clinical Scenario

The colorectal cancer (CRC) is among the most commonly diagnosed cancer (Siegel et al., 2020) and represents the third cause of cancer death worldwide. However, it is worth noting that more than one-half of these deaths are attributable to additional causes as obesity, smoking and more general unhealthy lifestyles [25]. Based on its prevalence, it is ranked third among the men with cancer and second among the women with cancer. Although its incidence strongly depends on the nation, higher rates are found in developing countries [22]. Individuals aged 50 years or older represent the portion of the population at greater risk. To have a prevision of the incidence of the CRC among the population, RL Siegel *et al.* has developed a method that allows a projection of the total number of new CRC cases and deaths that will occur in 2020 in the United States. Table 1.1 below shows how these numbers are close to 150.000 cases and 53.200 deaths.

	CASES						DEATHS	
AGE, YEARS	COLORECTUM	PERCENT	COLON	PERCENT	RECTUM	PERCENT	COLORECTUM[a]	PERCENT
Birth to 49	17,930	12%	11,540	11%	6,390	15%	3,640	7%
50 to 64	50,010	34%	32,290	31%	17,720	41%	13,380	25%
≥65	80,010	54%	60,780	58%	19,230	44%	36,180	68%
All ages	147,950	100%	104,610	100%	43,340	100%	53,200	100%

Note: Estimates are rounded to the nearest 10 and exclude in situ carcinoma.
[a]Deaths for colon and rectal cancers are combined because a large number of rectal cancer deaths are misclassified as colon.

Figure 1.1: Estimated numbers of CRC cases and deaths in the United States in 2020

Fecal immunochemical test and colonoscopy are among the most common screening procedures: while the former is based on a chemical analysis of the feces and it is recommended every 1-2 years, the latter made use of a semi-rigid endoscope to localise gastrointestinal polyps (Figure 1.2 - Left) in the colon and it is usually done every 10 years. Even thought the exams have the same target, the traditional colonoscopy allows for a more precise diagnosis thanks to the possibility to do a biopsy of the interested area of the GI tract. However, unlike the fecal immunochemical test, this last technique employs an invasive procedure that usually causes patient discomfort. Moreover, due to the method of actuation of the endoscope, this exam sometimes causes damages to the tissues of the GI tract, so it involves a discrete level of risk for the health of the patient.

A more complex scenario is involved when the patient is affected by the Inflammatory Bowel Disease (IBD); this term is used to identify two clinically different conditions named Crohn's Disease and Ulcerative Colitis (Figure 1.2 - Right) that are identified by a chronic inflammation of the GI tract. For patients affected by the IBD, the colonoscopy is extremely painful but crucial for the diagnosis of the disease.

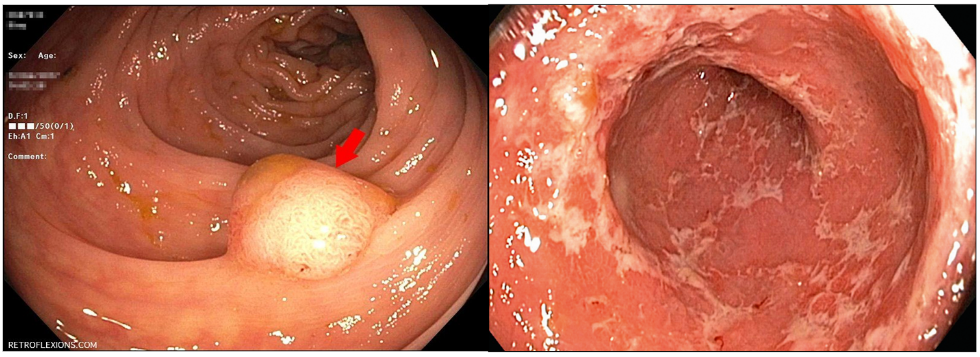


Figure 1.2: Two pictures acquired during a colonoscopy: Colon polyp (Left), Ulcerative Colitis (Right).

1.2 Flexible Endoscopy

The term *endoscopy* refers to a procedure used in medicine to examine the inner part of an hollow organ or cavity of the body by means of an endoscope with diagnostic and therapeutic capabilities. Areas of interest include gastrointestinal tract (GI tract), the respiratory tract, the urinary tract, the female reproductive system and more general closed body cavities.

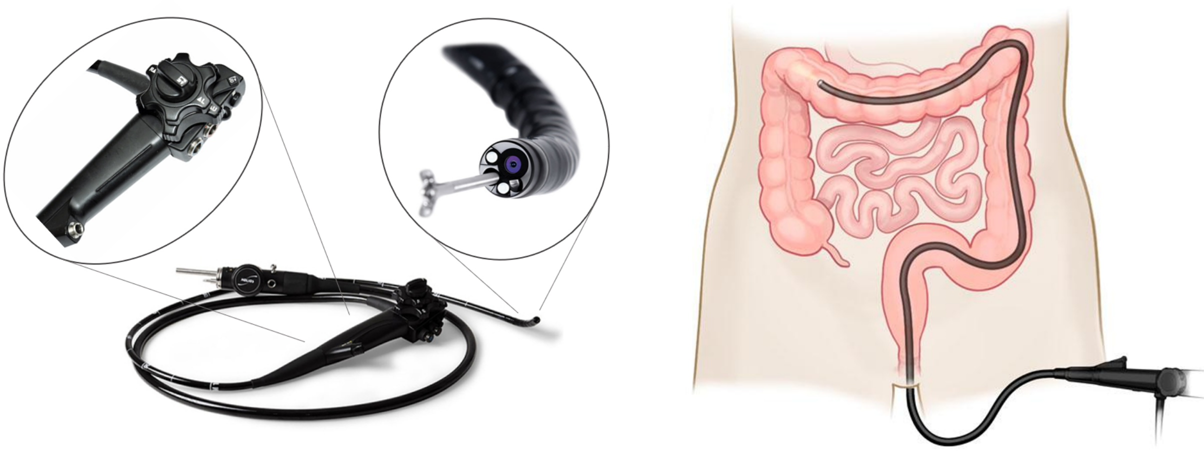


Figure 1.3: Details of a traditional flexible endoscope: control system for the bending section and endoscopic tip (Left). Illustration of a traditional colonoscopy (Right)

Among them, the *colonoscopy* is one of the most common screening procedure which is recommended every 10 years to adults aged 50 years or older to examine the large intestine. The exam is performed by means of an endoscope, i.e. a flexible device made of an insertion tube (length 1235 ÷ 1250 mm; diameter 7.5 ÷ 12.1 mm) whose tip allows for high resolution images of the inner wall of the GI tract in real-time. Depending on the model, the endoscope can host many channels (diameter: 2.0 ÷ 4.8 mm): an illumination channel to highlight the area of interest, an instrumentation channel though which the physician can insert surgery tools (biopsy forceps, needles, balloons etc.) and a water nozzle used to clean the surroundings. This device is usually inserted into a semi-rigid sleeve that allows the physician to tilt and steer its final bending section (i.e. the endoscopic tip) by means of a control box with a dial mounted on it. During the exam, the endoscope is inserted into the rectum of the patient and pushed forward along the large intestine. Thanks to the vision capabilities of the tip, the physician can change the direction of motion by bending the final section of the endoscope.

This widely used technique allows for an efficient navigation of the GI tract but the use of a back-push actuation and the semi-rigid sleeve implies an important drawback. Even though the patient is treated under local anesthesia to allow a painless procedure, the friction between the semi-rigid sleeve and the inner wall of the colon could cause tissues' irritation and damages. Moreover, the discomfort of the patient is increased by the motions of the bowel and its deformations that are induced by the stiff device. It is worth noting that the ability of the physicians to tilt the bending section and push forward the device plays an important role to minimize the discomfort of the patient. For this reason, extensive training sessions and experience are usually needed to master the traditional flexible endoscope.

Taking into account the friction forces between the device and the tissues, Figure 1.4 (Left) shows how the different inclinations of the bending section during the same manoeuvre can significantly affect the pain of the patient: the deformation of the GI tract can be minimized by spreading the same force on a surface as large as possible. Figure 1.4 (Right) shows instead the so called *N-Shape manoeuvre* and highlights how the deformations of the GI tract are induced by the semi-rigid endoscope when the physician performs a not optimal manoeuvre.

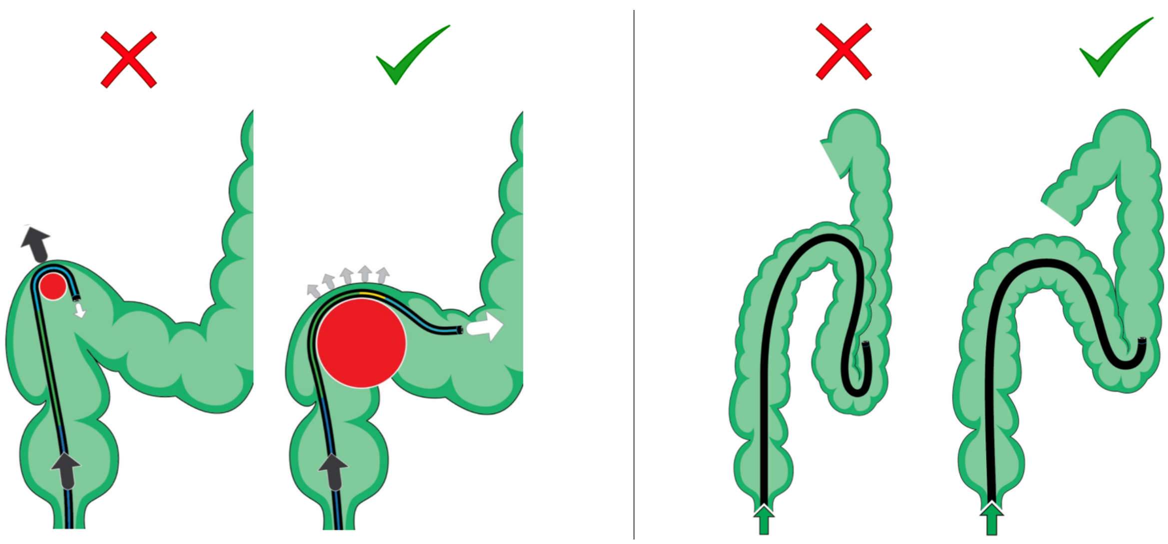


Figure 1.4: Distribution of the forces based on the inclination angle of the bending section (Left). Comparison between a correct and a wrong N-shape manoeuvre (Right). Credit: Ilaria Bondi, Morena Masetti.

In the last 20 years, a wide range of alternative solutions to the traditional endoscopy have been studied and developed to minimize the risks of tissues damages and so to make the patients close to this recommended screening method. Due to the limited amount of essential manoeuvres that are repeated during every colonoscopy, robotics represents a saver and more efficient solution that can also allow the navigation of sections of the GI tract that cannot be reached by traditional endoscopes.

1.3 Robotic colonoscopy

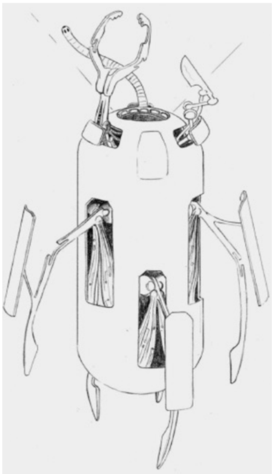
The term *Robotic Endoscopy* refers to three main categories [16]: robot assisted rigid endoscopy, robot assisted flexible endoscopy and active gastrointestinal endoscopy.

The devices of the first category are made of a rigid endoscope whose position can be precisely controlled, changed or maintained fixed by a robotic system. The interface between the physician and the robot consists of a set of pedals and some vocal commands. Among the main advantages, the ability to remove both the tremor of the device and the need of a second operator. These devices are usually employed during surgical operations to have a view of the body cavity but are not intended for GI endoscopy. SOLOASSIST II™ by AKTORmed GmbH and AutoLap™ by Medical Surgical Technologies are among the systems for robot-assisted rigid endoscopy currently on the market.

Figure 1.5: SOLOASSIST II™ by AKTORmed GmbH (Left) and AutoLap™ by Medical Surgical Technologies (Right).

Moving to the category of *robot-assisted flexible endoscopy*, many devices focused on the navigation of the GI tract are encountered. Most of these systems consist of a flexible endoscope mounted on a purpose-built robotic platform (usually a manipulator) that allows the physician to locate and drive the endoscope shaft by means of foot pedals. The manoeuvres of the bending section and the rotation of the semi-rigid sleeve are instead done manually by the physician whose hands are free.

Among the devices of the last category, active capsule endoscopes with locomotion ability represent the most promising solution for GI endoscopy and colonoscopy to reduce the risk of perforation and improve patient tolerance. As widely explained in the work of Piotr R Slawinski *et al.* [26], recent improvements in the actuation schemes of capsule endoscopes suggest that magnetic manipulation is among the main target of the ongoing research and many clinical applications of these technologies are expected in the next years. Based on the work of Piotr R Slawinski *et al.* a description of the most relevant active capsule endoscopes is presented hereby for completeness; the subdivision of these devices is based on the kind of actuation. Mechanical actuation is the first technique of locomotion that has been widely adopted before 2011 for wireless and tethered capsule endoscopes. In these devices, the synchronous motion of embedded mechanical components as legs, anchors, threads, and tails, that are energized by an internal power source, is employed to propel the capsule forward along the GI lumen. Many modes of locomotion have been tested by the researchers both to ensure a faster mobility and to avoid the back slippage problem caused by the interaction of the mechanical components with the mucosa of the GI tract; these include crawling, swimming, paddling-based and inch-worm. Table 1.6 lists the most relevant devices developed starting from 2004.



Ref.	Year	Actuation mode	Highest testing level	Speed (cm/min)
Standard colonoscopy[28]	-	-	-	7
Menciassi *et al*[13]	2004	Legged	*Ex vivo*	NA
Kim *et al*[14]	2004	Inchworm	*Ex vivo*	1.47
Kim *et al*[26]	2005	Inchworm	*Ex vivo*	13.4
Wang *et al*[22]	2006	Inchworm	*In vitro*	NA
Quirini *et al*[8]	2008	Legged	*Ex vivo*	3
Quirini *et al*[16]	2008	Legged	*Ex vivo*	6
Valdastri *et al*[10]	2009	Legged	*Ex vivo*	5
Tortora *et al*[11]	2009	Swimming	*In vivo*	90
Kim *et al*[17]	2010	Paddling-based	*In vivo*	17
Woo *et al*[18,19]	2010	Electrical stimulus	*In vitro*	17.46
Morita *et al*[32]	2010	Swimming	*In vivo*	300
Sliker *et al*[20]	2012	Treads	*In vivo*	18
Lin *et al*[23]	2012	Inchworm	*Ex vivo*	30
Chen *et al*[24]	2013	Inchworm	*In vitro*	NA
Chen *et al*[25]	2014	Inchworm	*Ex vivo*	2.3

NA: Not available.

Figure 1.6: Concept-design of a mechanical actuated capsule (Left). Most relevant mechanical actuated capsule endoscopes developed starting from 2004 (Right). Credit: Picture-Virgilio Mattoli, Table-Piotr R Slawinski.

Since the vast majority of these capsule endoscopes is intended to have a size small enough to be swallowed by the patient or inserted in other cavities, the big amount of volume used to host the mechanical parts strongly reduces the available space for the energy storage. Due to this main limitation, these devices show little promise for clinical trials.

More encouraging results have been obtained by means of magnetically actuated capsule endoscopes. Among the main advantages, the possibility to use a small permanent magnet embedded in the device as an endless energy source. Moreover, the absence of the mechanical components previously employed implies the capability to further reduce the dimensions of the capsule. Once swallowed by the patient, the pose of the device can be controlled by means of a magnetic field generated either by an external permanent magnet or an electromagnetic coil. In this applications, the actuation is ensured by the magnetic coupling between the two magnets and a propulsion of the capsule endoscope is reached whenever a magnetic field gradient is generated. Despite of the simple actuation principle, the attenuation of the magnetic force with the distance must be considered to always ensure the control of the device. However, it is worth noting that the human tissues do not involve a further attenuation of the magnetic filed.

By analysing the developed technologies, two distinct actuation techniques can be identified: *hand-held magnet actuation* and *robot-based control*. Although they are significantly different, both are aimed at controlling the capsule endoscope. In the hand-held magnet actuation, the physician moves the device according to a set of real-time images provided by the embedded camera. This on-sight technique is clearly affected by a low accuracy since it does not provide any information of the magnetic field that actuates the capsule endoscope. More advance applications relay on computer-assisted actuation, more precisely on robot-based control. The motion of the external source of magnetic field (permanent magnet or electromagnetic coil) and so the one of the capsule is controlled by a robotic system that is usually a robotic manipulator. The feedback is ensured by a set of sensors that allow for a precise localisation and path planning. Depending of the application, the sensors can be placed either inside or outside the capsule endoscope. Moreover, the achievement of an extremely precise localisation may allow the navigation of GI tracts that cannot be reached by the traditional flexible endoscopes.

Chapter 2

Motivation and Context

2.1 The Magnetic Flexible Endoscope

The work described in this thesis is based on an innovative robotic platform designed for painless endoscopy by the Science and Technologies Of Robotics in Medicine (STORM) Lab of the University of Leeds - Leeds (UK) and Vanderbilt University - Nashville (TN): the Magnetic Flexible Endoscope (MFE) [8]. The system, which aims to improve patient quality of life and survivability from colorectal cancer, has faced many improvements thanks to the work of engineers and clinicians coming from both universities. Among them, the replacement of the original hardware components with more efficient ones and the development of specific pieces of software that allows the achievement of new complex tasks. Thanks to the wide range of promising results and to the growing interest in the clinical field, the Magnetic Flexible Endoscope has won the KUKA Innovation Award 2019 during the _Healthy Living Challenge organized by KUKA GmbH. The next milestone for this innovative endoscope is scheduled on 2021 when this technology will be used for the first-in-human trials.

The main feature of this innovative robotic platform is the ability to perform a front-pull actuation of the endoscopic tip by means of one magnet placed externally with respect to the body of the patient. This alternative solution to the traditional endoscopy allows to overcome the rear-push mechanical actuation and the use of the semirigid insertion tube that is needed to prevent buckling and trauma owing to tissue stress. Unluckily, the manual operation of magnetic actuation is not intuitive and therefore the usage of computer-assisted operations is essential to assist the operator during both training sessions and complex clinical maneuvers. Moreover, the MFE has been equipped with proprioceptive sensing and software algorithms for the autonomous navigation, retroflexion [27] and diagnostic/therapeutic operations. More specifically, the autonomous capability of the MFE is addressed to polyp detection in colonoscopy due to the repetitive nature of many clinical maneuvers.

Both the description of the system and the developed materials are referred to the latest version of the Magnetic Flexible Endoscope.

2.2 System and Software Environment

The Magnetic Flexible Endoscope consists of four main hardware subsystem: a robotic system placed externally to the patient that acts as actuation source, a tethered endoscopic capsule equipped with many different sensors intended to be moved inside the GI tract by means of magnetic coupling, a processing unit and a purpose-built circuit that enables the communication between the capsule and the software of the MFE. Figure 2.1 shows an overview of the entire system.

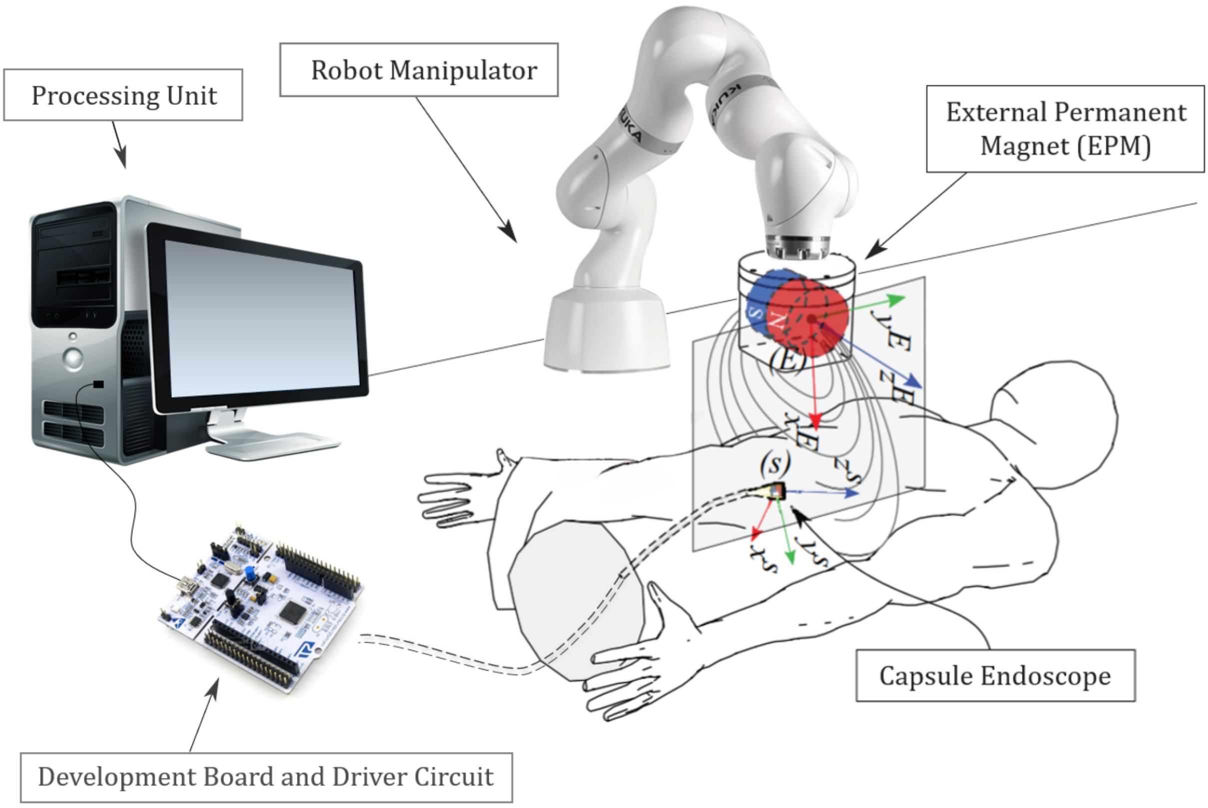


Figure 2.1: The Magnetic Flexible Endoscope developed at the StormLab UK.

The robotic system is composed by a 7-axis collaborative robotic manipulator specifically designed for medical applications by KUKA (KUKA LBR Med, KUKA GmbH, Augsburg, Germany) that can be either controlled by the physician by means of a specific joystick or programmed to follow predefined paths autonomously. A purpose-built magnetic system that allows both localisation and actuation tasks is mounted on its end-effector. More specifically, it includes:

- A neodymium iron boron (NdFeB) cylindrical permanent magnet (N52 grade, 101.6 mm diameter and length; ND_N-10195, Magnet-world AG, Germany) with axial magnetization and 1.48 T remanence held by means of a 3D printed box. Thanks to its strong magnetic field this magnet represents the main actuation source: it drags the capsule in its same moving direction, as long as the distance from the endoscopic tip does not become too large. The higher is the distance, the lower is the magnetic force between the EPM and IPM and whenever the former increases to much, the two magnets are decoupled. This scenario usually occurs when the capsule remains trapped inside a tissue folder of the GI tract. Note that a cylindrical magnet has been selected for this application thanks to the available mathematical models of the generated magnetic field as the Dipole-Moments or the Generalized Complete Elliptic Integral. Even though the former is simpler and can be used for testing purposes, the latter is much more accurate and faithful to the real EPM magnetic field (See Section 2.3.1).

- An electromagnetic coil [23] built using 24 AWG ($R_{int} = 7\Omega$) wire with 160 turns arranged in 2 overlapping layers (180 mm diameter and 40 mm length) held by an additional 3D printed structure. Note that, as it will be explained in Section 2.3.2 the circular shape of the coil allows to exploit precise mathematical model of the generated magnetic field.

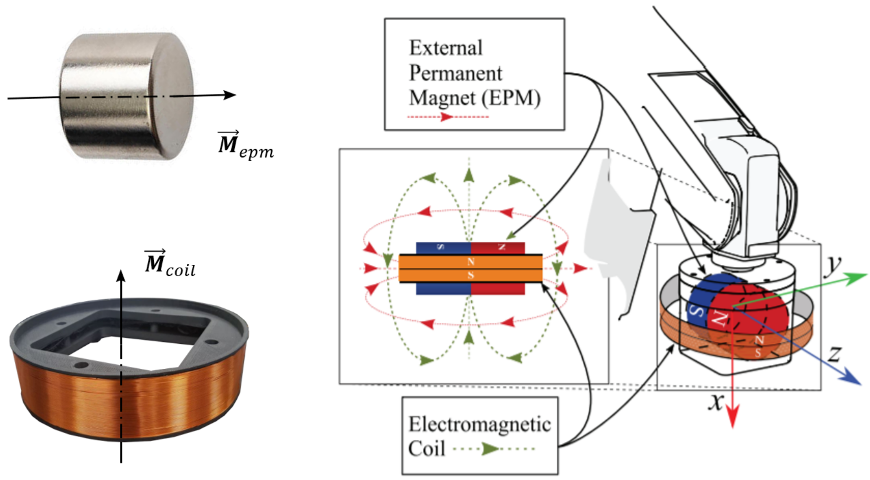


Figure 2.2: External Permanent Magnet (Left) and Coil (Right)

Moving to the description of the endoscopic tip (20 mm diameter and 22 mm length), it has a soft tether that enables functionalities commonly used in the traditional endoscopy and facilitates the data flow between the sensors and the processing unit. The capsule is equipped with a camera for vision, a water nozzle to simultaneously irrigate the inspection area and the camera, a LED for illumination and an instrument channel to enable biopsy procedures. Since the application is based on magnetic direct propulsion, the capsule itself contains a small axially magnetized cylindrical NdFeB permanent magnet (Neodymium 52 Mega-Gauss Oersteds N52-grade, 11.11 mm diameter and length; D77-N52, K&J Magnetics, USA) with a remanence of 1.48 T. From now on it will be referred as the Internal Permanent Magnet (IPM). All around the IPM, six Hall effect sensors (A1391, Allegro Microsystems, USA) are placed around the magnet in order to approximate two triaxial Hall sensors separated by a constant distance. The sensors, that are chosen with a large dynamic range ($0.1\mu T \div 2T$), are placed in specific locations to avoid their saturation by the IPM magnetic field. It is worth mentioning that the biases introduced by the IPM are then removed from the magnetic field measurements so to detect $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$ only.

Moreover, the capsule is equipped with a 6-DoF Inertial Measurement Unit (IMU-LSM330DLC, STMicroelectronics, Switzerland) which features a 3D digital accelerometer (full-scale acceleration range up to ±16 g) and a 3D digital gyroscope (angular rate range up to ±2000 degrees per second). The sensor provides data of linear accelerations and angular velocities that are essential for the capsule localisation. Figure 2.3 below shows the overall arrangement of the sensors inside the endoscopic tip. While the IMU provides in output digital values that can be directly used for the computations, the output of the six Hall effect sensors is in the analogue domain. To enable the conversion in digital, a 16 bit ADC has been added to the setup of the application.

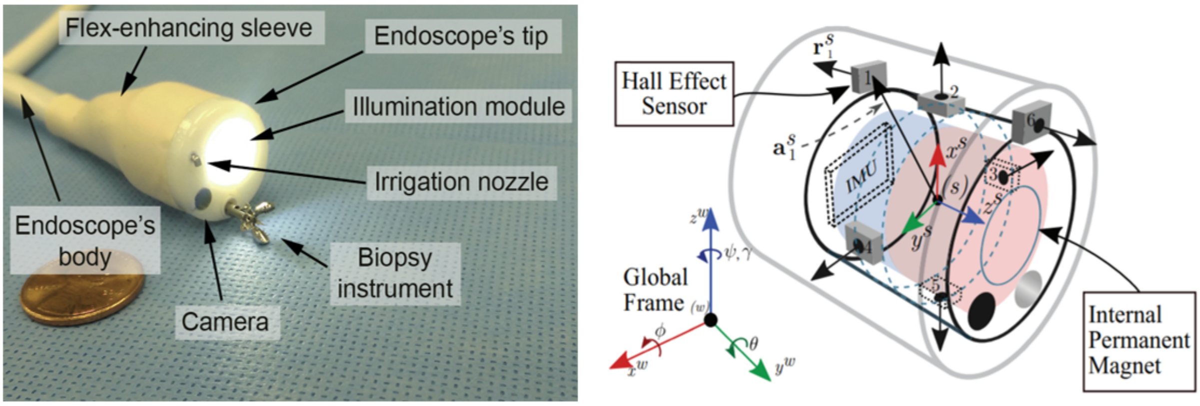


Figure 2.3: The architecture of the endoscopic tip and the position of the sensors

The last part of the hardware architecture of the MFE is a purpose-built electronic circuit based on a STM Nucleo development board (STM32F411RET, ARM Cortex M4) and a driver circuit that is intended to:

- Perform several signal processing techniques with the measurements coming from the Hall effect sensors and IMU. The processed data are then sent to the processing unit via USB cable and used in ROS to perform the magnetic localisation;
- Generate the square wave signal needed for the oscillating magnetic field of the coil.

Before proceeding with the description of the software architecture, a preliminary observations is needed: it must be highlighted that the EPM is used both for actuation and sensing while the magnetic field produced by the coil is employed for sensing only. Although this peculiar feature allows to simply the needed hardware architecture, it involves the development of more complex algorithms and software.

The scheme 2.5 below shows the connections between the previously described hardware components that are crucial to ensure the closed-loop control of the endoscopic tip. In Figure 2.5, the final set-up of the Magnetic Flexible Endoscope is shown.

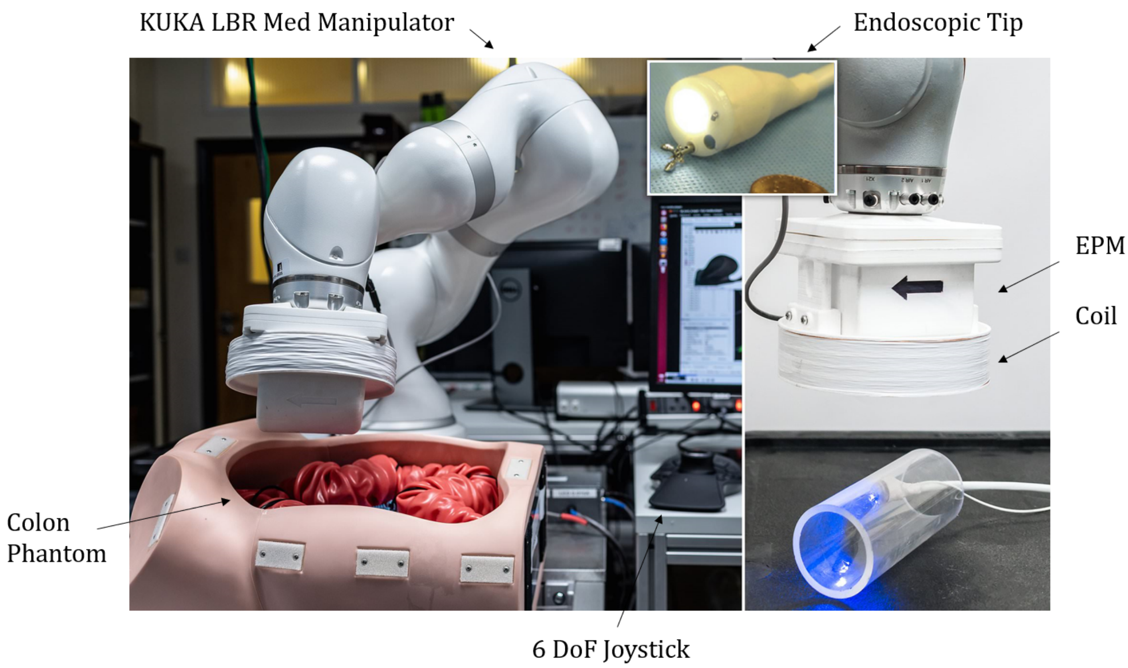


Figure 2.4: Final set-up of the Magnetic Flexible Endoscope.

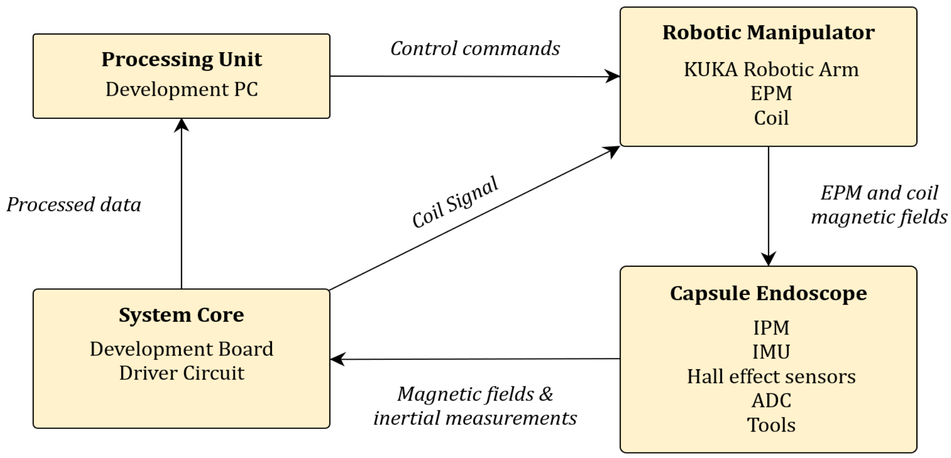


Figure 2.5: Basic block diagram of the MFE.

2.3 Algorithms for Magnetic Localisation

An accurate and precise estimation of the pose of the endoscopic tip is crucial to successfully achieve the closed-loop control of the device. Once the position and orientation of the capsule are known, the physician is able to provide a feedback for its motion, locate the lesions in the GI tract and determine future follow-up treatments. It may be recalled that the motion input of the physician can be used, in conjunction with the pose estimation, to generate a suitable control action.

By analysing the endoscopic devices available in literature and on the market, it can be highlighted how the developed localization strategies are based on a wide range of physical principles. Among them image-based tracking systems, magnetic fields localisation, algorithms based on electric potential and X-rays tracking.

Given the nature of the robotic platform developed by the STORM Lab, this frame of work will provide the reader an overview of the localisation techniques based on magnetic fields and electromagnetic waves. Before starting with the description of the most relevant applications, it is important to highlight the advantages and drawbacks of the use of magnets and energized electromagnetic coil for the capsule localisation.

Advantages:

- Magnetic coupling is one of the few physical phenomena capable of transmitting motion across a physical barrier. In gastrointestinal endoscopy, remote magnetic manipulation has the potential to make screening less invasive and more acceptable to the patients;

- The interactions between static magnetic fields with a strengths of 1T÷2T and tissues are not harmful for human health. In support of that, the work of DW Chakeres *et al.* [3] showed how normal subjects exposed to static magnetic field, with varying strengths of up to 8T, have demonstrated no clinically significant changes in vital signs;

- Concerning static and low frequency magnetic field, the human tissues introduce a null or negligible attenuation of its strength [11]. This observation allows the designer to use existing accurate models of the magnetic field without the need of any additional correction factor;

- Unlike other localisation approaches that work with photoelectric phenomena, the ones based on magnetic fields have an higher degree of flexibility since the Hall-effect sensors do not need to be in the line of sight to detect the capsule.

Despite of the previous advantages, the following drawbacks are encountered:

- The presence of ferromagnetic objects inserted into the workspace causes an unwanted distortion of the magnetic field and brings to the consequent lower accuracy in the localisation. This phenomenon can not be neglected in a clinical scenario which is equipped with surgical tools and active medical devices.

- If a magnetic actuation strategy is implemented, undesired interference with the magnetic localization system may occur.

Since the beginning the MFE project, the team of the STORM Lab has designed and developed three different algorithms for the magnetic localisation of the endoscopic capsule. Even thought they relay on different mathematical tools, they share some common features. Among them, the reference frames and the modelling of the magnetic field generated by the EPM and coil: firstly, $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$ are evaluated offline in the selected workspace and used to create some look-up tables that are then employed online in the localisation algorithm.

Reference Frames

All the computations of the three algorithms are based on the same set of coordinate frame (or reference frame RF) as shown in Figure 2.6.

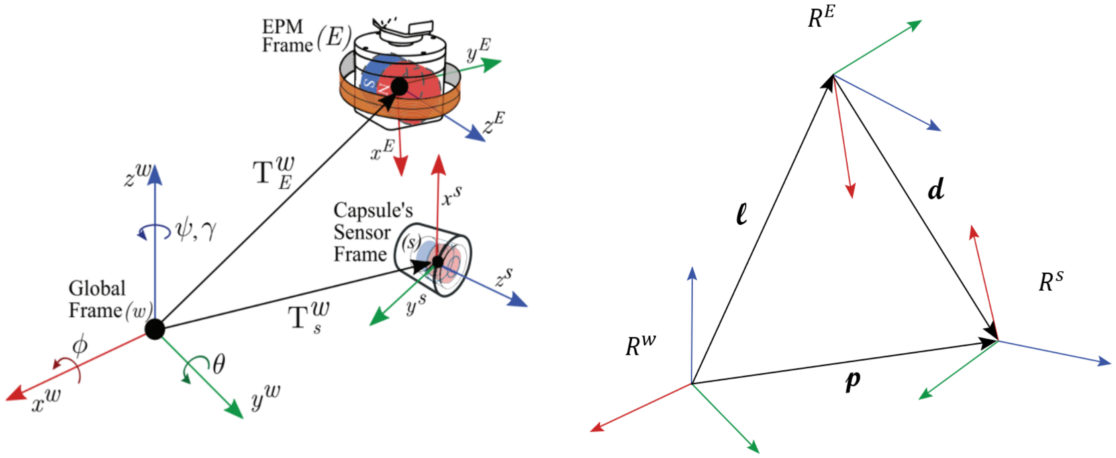


Figure 2.6: Reference frames of the MFE (left). Position vectors between the coordinate frames (right). Credit: Addisu Z Taddese.

The *World (or Global) Frame* R_w is the inertial frame and it is tagged as the "parent frame" of all the others "child frames". Due to the mathematical simplifications, in the Magnetic Flexible Endoscope R_w coincides with the *Base Frame* of the KUKA manipulator. The *EPM Frame* R_E perfectly matches with the center of the External Permanent Magnet that differs from the End-Effector frame. z^E is directed along the axis of polarization of the magnet while x^E is pointing down and lays of the rotation axis of the manipulator's wrist. Finally, the *Capsule Frame* R_s is centered on the IPM with the z^s directed along its polarization axis. The apexes $(^w)$, $(^E)$ and $(^s)$ identify variables evaluated in the Global, EPM and Capsule frames accordingly. ${T_E}^w$ and ${T_s}^w$ represent the transformation matrices (i.e. rotation matrix and translation vector) between the EPM RF - World RF and Capsule RF - World RF. Given the coincidence between R_w and the base frame of the KUKA manipulator, ${T_E}^w$ is always known thanks to a set of internal real-time encoders installed on its revolute joints. On the contrary, ${T_s}^w$ is completely unknown until the localization algorithm retrieves to full capsule pose. It is worth mentioning that the transformation matrix between the EPM and Capsule Frames ${T_s}^E$ can be obtained by simple composition of vectors and matrices.

$$T^E_s = \begin{bmatrix} R^E_w R^w_s & \mathbf{t}^w_s - \mathbf{t}^w_E \\ \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{5.23}$$

As for the position vectors, $\boldsymbol{l}$ and $\boldsymbol{p}$ identifies the position of the EPM and the capsule with respect to the Global Frame respectively. Instead, vector $\boldsymbol{d}$ represents the position of the capsule with respect to the magnet center and it is useful when dealing with the field density map.

Look-up Table of the magnetic field & Singularity Plane

Accurate models of the EPM and Coil magnetic fields ($\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$) are crucial both to minimize the estimation error and maximize the processing speed of the localization algorithm. In literature, there exist many mathematical models of the magnetic field generated by a cylindrical magnet; All of them allow the user to compute the cylindrical coordinates of the magnetic field corresponding to a given position vector.

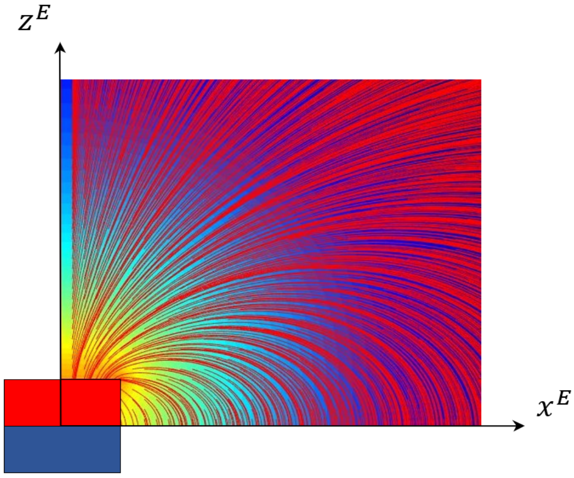


Figure 2.7: Map of the EPM magnetic field on the semi-positive XZ plane.

By repeating this computation for a settable number of points in a chosen workspace, a field density map that links the position with respect to the center of the magnet and the corresponding magnetic field strength can be obtained. Clearly, the higher is the number of selected points, the thicker is the generated grid. With reference to R_E, Figure 2.7 shows the look-up table of $\mathbf{B}_{EPM}$ on the semi-positive XZ plane. Moreover, by means of simple reflections across the planes of symmetry of the cylinder, a 3D map of the magnetic field can be obtained starting from a 2D one. In all three localisation algorithms, the maps of the EPM and Coil magnetic fields are computed offline and used online as look-up tables to minimize the computational effort: given a set of magnetic field measurements, the look-up table is used to search the corresponding (unique) position vector.

Despite of the great utility of the field density map for the computation of the position vector, issues are encountered when the endoscopic device covers specific areas of the workspace around the permanent magnet. The source of this problem is the presence of *Singularity Regions*. In the case of an axially magnetized cylindrical permanent magnet they are reduced to a *Singularity Plane*: plane that is passing through the center of the magnet, orthogonal to its magnetization axis.

Let's consider a 3D map in cylindrical coordinates of the magnetic field generated by an axially magnetized cylindrical EPM. The look-up table is intended to be a bijective function between the set of position vectors and the one of magnetic field strengths, where each element of the former set is paired with exactly one element of the latter one.

Even though a unique magnetic field vector is retreived when the inputs are the coordinates of a point around the magnet, the viceversa does not hold. More precisely, when the inputs are the cylindrical coordinates of the magnetic field corresponding to a point belonging the the Singularity Plane, a not unique solution is retreived. On this plane, the radial coordinate of the magnetic field is null while the axial component is the same for all the infinite points belonging to the same circumference. See Figure 2.8.

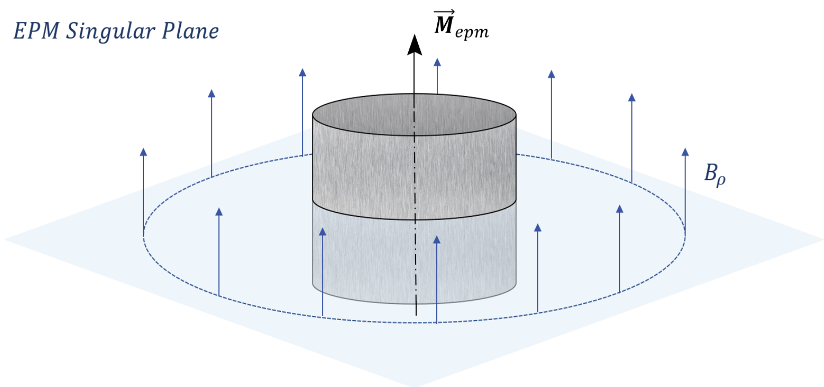


Figure 2.8: Singularity plane of the EPM. An infinite number of points are identified by the same couple of where B_ρ and B_z.

In Sections 2.3.2 and 2.3.3, the three magnetic localization algorithms developed by the team of the STORM Lab are presented in a chronological order. For each of them, a detailed description of the mathematical formulation, results and main limitations are provided to the reader. As it will be explained later on, only the last algorithm successfully solves the problem of the singularity plane.

2.3.1 State of the Art

Magnetic fields **B** are widely used in biomedical procedures for a not-invasive localization of medical devices inside the human body. Well known fields of application are general surgery and endoscopy. As for the latter, many Companies and teams of research relay on magnetic field to address the problem of localizing endoscopic capsules thanks to the negligible attenuation of **B** as it pass through the human tissue. Moreover, the use of a magnet instead of a battery as a source for the localization enables to safe space inside the capsule.

An overview of the existing localization methods designed for endoscopic devices and based on magnetic fields is presented hereby. A more detailed description is presented in the work of Pirotta [23]. Considering that magnetic forces can be also employed for the actuation of the endoscopic capsules, two main categories of algorithms for pose estimation can be identified:

- Magnetic localisation algorithms for *passive capsule endoscope* do not consider the interference with the actuation sources. More precisely, they are addressed to wireless endoscopic devices that are swallowed by the patient and moved inside the GI tract by the gravity force and peristalsis.

- In the magnetic localisation algorithms for *active capsule endoscope*, the same magnetic source is used both for localisation and actuation. This specific family of estimation methods allows a precise pose detection without interfering with the actuation of the device.

Magnetic Localization for passive capsule endoscopes

Among the first applications that use a permanent magnet embedded in the capsule endoscope (IPM), the Superconducting Quantum Interference Device (SQUID) developed by Weitschies *et al.* [33] consists of a complex system made of sensors that is intended to be placed on the abdomen of the patient for the localisation (only position) of a small permanent magnet embedded in a capsule endoscope. The limitation in the estimation of the orientation has been overcome by Schlageter *et al* [24] who replace the SQUID with a 2D array of 16 Hall-Effect sensors. The system ensures a refresh rate of 20Hz within a limited cubic workspace of 20cm side. To increase the available volume for IPM pose detection, Chao *et al.* [11] design a four-planes structure covered by 64 3-axis Hall-effect sensors. Although the precision increases up to 1.8mm and 1.6 °, the overall cost of the system is too high. Similarly, Wu *et al.* propose a wearable vest equipped with Hall-effect sensors grouped in 10 modules. The system is designed to correct the magnetometer measurement by deleting the component of the Earth Magnetic Field.

Dealing with passive capsule endoscopes, localisation systems based on different working principles can be spotted. For example, Nagaoka and Uchiyama [21] replace the IPM inside the wireless capsule with a small coil of ferromagnetic material. Thanks to a bigger energized coil placed outside the patient, the localisation algorithm retrieves the capsule's pose by employing the induced electromotive force.

Magnetic Localization for active capsule endoscopes

As previously mentioned, these algorithms are able to detect the capsule pose with precision without interfering with the actuation of the device that is also based on magnetic coupling. This property reflects the features of the MFE closely.

In 2009, researchers of Olympus based their project on the fact that two magnetic fields at very different frequencies do not interfere one with the other. More precisely, they performed the simultaneous localization and actuation of an untethered capsule by means of high-frequency and low-frequency oscillating magnetic fields respectively. Three coils, that are placed outside the patient along three orthogonal directions, generate a low frequency $\mathbf{B}$ that allows the endoscopic device to move forward and backward. The motion is facilitated by a spiral outer pattern of the capsule's surface [2]. This architecture allows to estimate the pose of the capsule based on the induced magnetic field on its embedded inner coil. A sub-millimetre accuracy can be achieved when the capsule is placed within 12 cm from the outer set of detecting coils.

The system designed by Ciuti *et al.* [4] is the first one in literature that employs a 6-DoF robotic manipulator with a permanent magnet mounted on its end-effector both for localisation and actuation purposes. As for the endoscopic capsule, it is equipped with a 3-axis accelerometer and with four permanent magnets sealed on its surface. To enable a reliable localization procedure, a fine-tuned calibration is needed: when approaching the workspace for the first time, a fast variation of the accelerometer data suggests that the magnetic coupling has taken place. Good results in pose detection can be achieved by considering a constant coupling between the permanent magnet and the capsule during the steering manoeuvre inside the GI tract. Since this hypothesis cannot be guaranteed in a clinical scenario during the real teleoperation of the capsule, poor results are achieved in experimental tests.

Finally, MG Kim *et al.* [15] develop a system based on a 5-DoF planar robotic manipulator whose end-effector grabs a parallelepiped magnet. In addiction, a rotary actuator allows the rotation of the magnet along the orthogonal axis with respect to the dipole direction. As for the capsule, two IPMs enable the magnetic coupling with the EPM, a set of Hall-effect sensors is used for sensing and the spiral outer pattern on its surface is employed to facilitate its motion inside the GI tract. The algorithm for the pose detection of the capsule is based on the generation of a time-varying magnetic field achieved by exploiting the rotary actuator. As a result of this motion, the magnetic flux density detected by the Hall-sensors has a sinusoidal shape. When the flux density reaches either its maximum or its minimum, a simple solution for the inverse kinematic problem can be found and the estimate of the capsule pose is retreived.

2.3.2 Real-Time Pose Detection for Magnetic Medical Devices

The first algorithm for the Magnetic Flexible Endoscope has been developed in 2013 by Christian Di Natali *et al.* [7] at the STORM Lab of the Vanderbilt University (Nashville, USA). This novel approach allows to detect in real time the 6-DoF pose of a magnetic capsule endoscope by taking advantage of the external magnetic field generated by an EPM with axial magnetization. Reminding that magnetic fields are also used to move the endoscopic probe inside the human body, the algorithm is designed for active capsule endoscope, i.e. it can cope with problems related to the interference between the actuation of the device and its localization. The employed set-up (Figure 2.9.A) includes a 6-DoF robotic arm (RV6SDL, Mitsubishi, Inc., USA), the EPM, a wireless endoscopic tip with the Hall-effect sensors and IMU arranged as shown in Figure 2.3, an optical tracker (Micron Tracker, Claron Technology, Inc., USA) and a gimbal (OBJECT 30, Object Ltd., Israel) that works as support of the capsule.

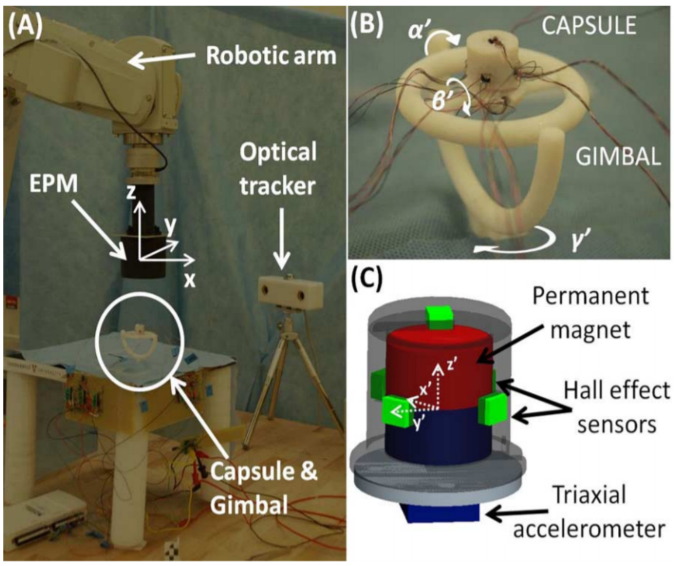


Figure 2.9: (A) View of the platform; (B) Close view of the capsule and the gimbal; (C) Schematic view of the capsule. Credit: STORM Lab.

Magnetic Field Model

This first algorithm relays on a magnetic field density map that covers a 30cm x 30cm x 30cm workspace centered on the EPM. It is characterised by a spatial resolution of 0.2mm along all the three directions (x^E, y^E *and* z^E) that is chosen as a trade-off between computational time and localization accuracy. More specifically, it is obtained by means of the *Magnetic Current Model (MCM)* that allows for a numerical solution by finite element integration. The MCM for an axially magnetized cylindrical permanent magnet is expressed as:

$$\mathbf{B}(\mathbf{d}) = \frac{\mu_0}{4\pi} \oint_{S''} j_m(\mathbf{d}'') \times \frac{(\mathbf{d} - \mathbf{d}'')}{|\mathbf{d} - \mathbf{d}''|^3} dS'' \tag{2.2}$$

where $\mathbf{d}$ identifies a generic point belonging to the workspace, $\mathbf{d}''$ is a point on the magnet surface, j_m is the equivalent current density and S'' is the EPM integration surface.

Procedure

The two cartesian triplet of Hall-effect sensors are used to acquire in two sets of cartesian components of the EPM magnetic field ($\mathbf{B}_1^s = [B_{x1}^s, B_{y1}^s, B_{z1}^s]^T$, $\mathbf{B}_2^s = [B_{x2}^s, B_{y2}^s, B_{z2}^s]^T$). Concurrently, the inertial measurements are used to estimate two out of the three Euler angles (i.e. ϕ^w and θ^w) that identify the attitude of the capsule with respect to the Global Frame. Given these 8 real-time measurements, the following steps are performed:

1. (1^{st} cycle only) An initialization of the unknown Euler angle φ^w is needed for the first iteration of the algorithm. The initialization is performed via the optimal tracker and the gimbal. The value φ_0^w is used as baseline and will be replaced by the output of the localisation for the next iterations.

2. Thanks to the knowledge of the remaining Euler angle φ^w, the measurements of the magnetic field $\mathbf{B}_1^s$ and $\mathbf{B}_2^s$ are moved from the Capsule to the EPM Coordinate Frame:

$$\mathbf{B}_i^E = R_E^w R_s^w (R_w^E)^T \ \ \mathbf{B}_i^s \tag{2.1}$$

   where R_E^w is the rotation matrix between the EPM and Global frame that is known thanks to the real-time encoders while R_s^w [1] identify the attitude of the capsule with respect to the Global Frame.

[1]Tait-Bryan 321 R_{321} rotation matrix

$$R_s^w = \begin{bmatrix} cos(\theta)cos(\varphi) & sin(\theta)sin(\phi)cos(\varphi) - cos(\phi)sin(\varphi) & cos(\phi)sin(\theta)cos(\varphi) + sin(\phi)sin(\varphi) \\ cos(\theta)sin(\varphi) & cos(\phi)cos(\varphi) - sin(\varphi)sin(\phi)sin(\theta) & sin(\theta)sin(\varphi)cos(\phi) - cos(\varphi)sin(\phi) \\ -sin(\theta) & cos(\theta)sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \tag{2.2}$$

3. Once the measurements are moved in the EPM Frame, they are used in conjunction with a numerical map that associates to every position vector $\mathbf{d}$ the corresponding EPM magnetic field $\mathbf{B}^E(\mathbf{d})$. As previously mentioned, the field density map is generated offline and treated online as look-up table. A search is performed in the map to find the values of (d_x, d_y, d_z) related to the magnetic field vectors that are the closest to the measured values $\mathbf{B}_1^E$ and $\mathbf{B}_2^E$. In the case of an axially magnetized cylindrical permanent magnet, the magnetic field can be expressed in cylindrical coordinates. Therefore, the employed search function coincides with the following system of equations.

Magnetic field in cylindrical coordinates:
$$\begin{cases} \mathbf{B}^E = B_\rho \hat{\mathbf{r}} + B_z \hat{\mathbf{z}} = \sqrt{(B_x^2 + B_y^2)}\hat{\mathbf{r}} + B_z \hat{\mathbf{z}} \\ \delta = arctan(\frac{B_y}{B_x}) \end{cases}$$

where $B_\rho \hat{\mathbf{r}}$ and $B_z \hat{\mathbf{z}}$ are the radial and axial components of $\mathbf{B}^E$ respectively. δ is instead the azimuthal angle. Thanks to the cylindrical symmetry of the EPM and to the knowledge that the capsule remains below the magnet in a clinical scenario (z < 0 with reference to Figure 2.9), the magnetic field map can be reduced to a slice of 15cm x 15cm domain. Each point of this slice is expressed by two cylindrical coordinates ($B_\rho(\mathbf{d})\hat{\mathbf{r}}$ and $B_z(\mathbf{d})\hat{\mathbf{z}}$) that allow a univocal association to a position vector $\mathbf{d}$.

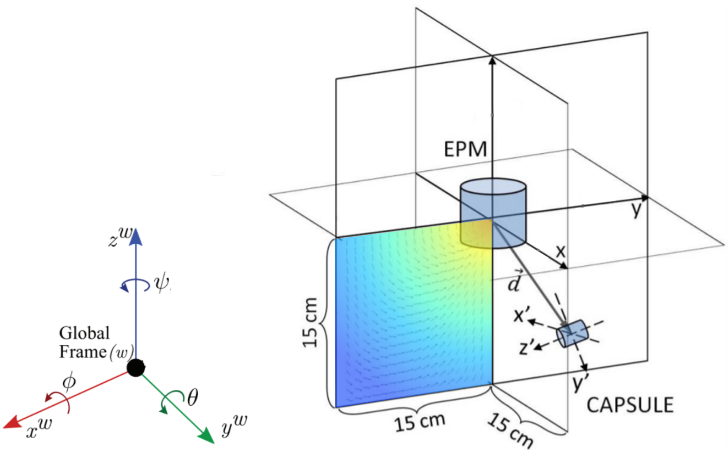


Figure 2.10: Field density map reduced to a slice of 15 cm x 15 cm domain.

For each of the two magnetic field vectors ($\mathbf{B}_1^E$ and $\mathbf{B}_2^E$) a sequential search is performed over the reduced bi-dimensional map to find $d_r = \sqrt{d_x^2 + d_y^2}$ and d_z. As for the azimuthal angle, it is evaluated from the values of B_x and B_y. Finally, the position $\mathbf{d}$ of the capsule in the EPM Frame is computed by averaging the just obtained $\mathbf{d}_1$ and $\mathbf{d}_2$; Knowing the transformation matrix T_E^w, the position vector $\mathbf{p}$ is so evaluated. The estimate of φ is instead achieved as the angle between the x-axis and the line intercepting the projections of $\mathbf{d}_1$ and $\mathbf{d}_2$ on the horizontal plane.

The capsule pose is fully identified with respect to the Global Frame. A new iteration of the algorithm begins, this time adopting the actual ϕ in the computation of R_s^w.

Performances Evaluation and Results

For the validation of the algorithm, Di Natali *et al.* fixed the endoscopic capsule to the gimbal whose coordinate are $\mathbf{p}^{w}_{Gimbal} = [0.0, 0.0, -15.0]^T cm$. The optical tracker is used as validation benchmark for ϕ^w, θ^w and ψ^w. As for the robotic arm, it is used to scan a 30cm x 30cm area on the XY plane above the capsule. The procedure is then repeated for nine different vertical positions between the EPM and the capsule (From -8.0cm to 0.0cm with a step of 1.0cm). As it is expected, the bigger is the distance between the capsule and the EPM, the higher is the estimation error: when $\mathbf{d}$ increases, the contributions of electromagnetic disturbances, soft and hard iron sources are no more negligible and affect the readings of the Hall-effect sensors. The estimation error e and the standard deviation σ are reported in Table 2.1 for $|\mathbf{d}| = 10cm$ and $|\mathbf{d}| < 15cm$.

	$e_{d_x} \pm \sigma_{d_x}$	$e_{d_y} \pm \sigma_{d_y}$	$e_{d_z} \pm \sigma_{d_z}$	$e_\psi \pm \sigma_\psi$
$\lvert\mathbf{d}\rvert = 10cm$	-4.3 ± 2.1 mm	-4.5 ± 1.9 mm	-3.9 ± 1.8 mm	$-12° \pm 29°$
$\lvert\mathbf{d}\rvert < 15cm$	-3.4 ± 3.2 mm	-3.8 ± 6.2 mm	3.4 ± 7.3 mm	$-19° \pm 50°$

Table 2.1: Experimental estimation errors and standard deviations.

Concerning the computational time, the sensor data acquisition and the localization algorithm require 6.5 ms and 16 ms per loop. The block diagram of the just analysed algorithm is shown in Figure 2.11 below. The estimated parameters that completely identify the pose of the capsule are highlighted in blue.

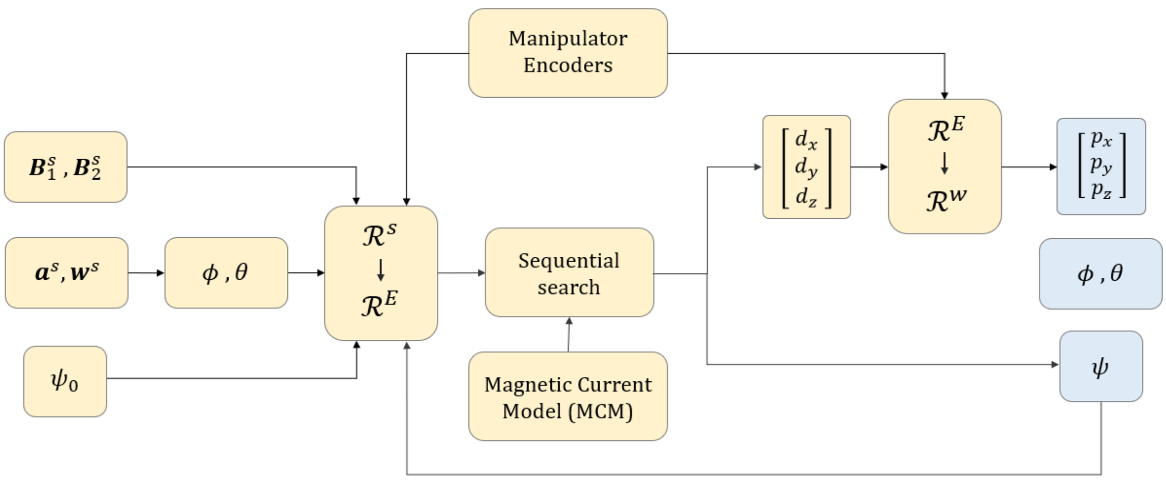


Figure 2.11: Block diagram of the *Real-Time Pose Detection Algorithm for Magnetic Medical Device*

Drawbacks and Limitations

Two main drawbacks affect this first localisation algorithm. Firstly, the yaw angle ψ_0 initialization introduced a strong limitation to the clinical application of the MFE:

although ψ could be initialized at the beginning of the endoscopic procedure, the same operation cannot be performed once the capsule is inside the body of the patient. It means that whenever the pose of the endoscopic tip is lost, the capsule must be pulled out and re-initialized.

Since the algorithm has been tested with the capsule fixed on the gimbal, the problem of regions of singularity of $\mathbf{B}_{EPM}$ has been neglected. Because of the use of a single permanent magnet (i.e. EPM) the algorithm would retrieve a not-unique solution whenever the capsule lays on the singularity plane. By considering the magnetic coupling between the EPM and the IPM in a clinical scenario, this condition is repeated frequently.

2.3.3 Jacobian-Based Iterative Algorithm

In 2016, Christian Di Natali *et al.* of the STORM Lab USA [6] developed a second algorithm for the magnetic localisation in Robotic Capsule Endoscopy. Unlike the first version, that performed an absolute search for the capsule position employing a look-up table of the magnetic field, this new algorithm allows the estimation of the pose in an iterative way. More specifically, a computationally faster localization is designed based on a closed-form expression for the Jacobian of the magnetic field relative to changes in the capsule pose. The algorithm uses the dipole assumption for the modelling of $\mathbf{B}_{EPM}$ and exploits the integration of IMU measurements to computed the orientation of the capsule. However, the integration of inertial data usually brings to unwanted drift that could become an issue over time. For this reason, the new algorithm is intended to be used in conjunction with an absolute localization technique that provides, at a slower rate, more precise initialization values for this faster algorithm.

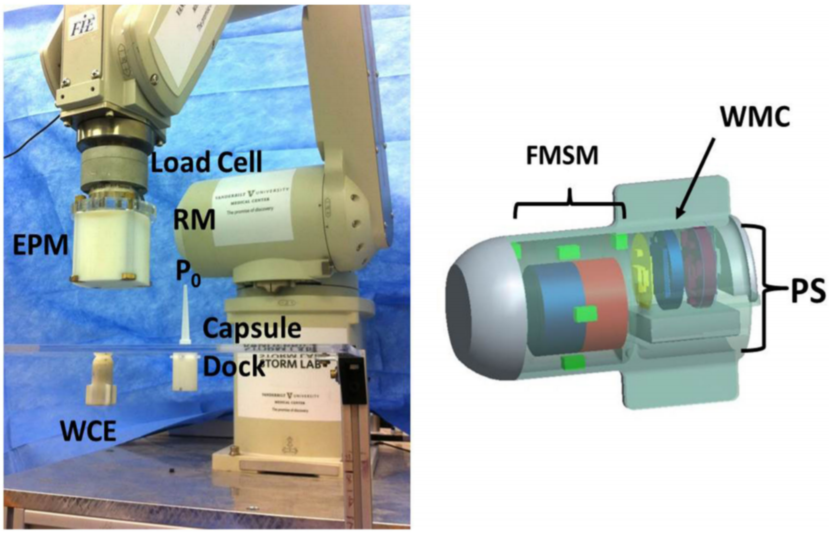


Figure 2.12: Experimental Hardware Set-Up (Left). Visual rendering of the Wireless Capsule Endoscope (Right). Credit: STORM Lab.

The employed set-up (Figure 2.12) shows few differences with respect to the one used in the previous algorithm. It includes a 6-DoF robotic arm (RV6SDL, Mitsubishi, Inc., USA) with a NdFeB cylindrical permanent magnet EPM (N52, remanence 1.48T, diameter and length of 50 mm) mounted on its end-effector and a wireless capsule which hosts the *force and motion sensing module(FMSM)*, a *wireless microcontroller (WMC)* and the *power supply (PS)*. The outer shell of the capsule is a 3D-printed prototype with four lateral wings used for calibration purposes. The positions of the six Hall-effect sensors and the IMU remain unchanged. Unlike before, no gimbal is used in the experimental set-up.

Preliminary observations

Regardless of the chosen magnetic field model, the Magnetic Direct Relationship (MDR) $\mathbf{p}_i \longrightarrow \mathbf{B}_i$ and so the Magnetic Inverse Relationship (MIR) $\mathbf{B}_i \longrightarrow \mathbf{p}_i$ consist of a system of nonlinear equations. Because of the high computational effort that is usually needed to solve nonlinear systems as the one employed in this algorithm, a first-order linearization can be applied to simply the computation of the solution.

More specifically, a Jacobian-based method (also known as Resolved Rates Method) has been selected for the new localization approach thanks to its ability in solving systems of nonlinear equations subject to the limitations of first-order linearization. To minimize the linearization error, it has been assumed that the acquisition rate of the capsule's pose is fast enough so that small displacements, both in position and in attitude, can occur between two subsequential measurements.

The second hypothesis consists in the knowledge of the capsule orientation: this result can be achieved by means of well known sensor fusion algorithm that uses the inertial measurements coming from an IMU to compute online the attitude (Euler angles or quaternion) of the sensor. The employed one is described below in more detail.

Capsule Orientation Algorithm

To move the magnetic field measurements from the *Capsule* to the *EPM Frame*, the capsule orientation knowledge (i.e. the Euler angles) is required to compute the rotation matrix R_s^w as follow: the cartesian components (a_x, a_y, a_z) of the Earth gravity vector g, that are retreived in output by the accelerometer, are used to compute the roll ϕ and pitch θ angles as follows:

$$\phi = atan2(a_y, \sqrt{a_x^2 + a_z^2}) \tag{2.5}$$

$$\theta = atan2(a_x, \sqrt{a_y^2 + a_y^2}) \tag{2.6}$$

As for the evaluation of the remaining yaw angle ψ, the adopted approach involves the application of the *Axis-Angle Method for rotation matrices* to the gyroscope outputs. At each iteration (i.e. after each Δt), the gyro data are used to compute the rotation matrix $\Delta R(\Delta\phi^s, \Delta\theta^s, \Delta\psi^s)$ corresponding to the instantaneous variations of the capsule orientation. These values can be evaluated in the *Capsule Frame* as follow:

$$\Delta\phi^s = g_x\Delta t, \quad \Delta\theta^s = g_y\Delta t, \quad \Delta\psi^s = g_z\Delta t \tag{2.3}$$

At the time instant t, the matrix ΔR^s is computed and moved into the corresponding *Axial-Angle Representation.* The angle of rotation φ and the instantaneous rotation axis w are defined as:

$$\varphi = arccos(\frac{trace(\Delta R) - 1}{2}) \tag{2.4}$$

$$w = \frac{1}{2sin(\varphi)} \sum_{j=1}^{3} (\hat{\mathbf{e}}_{j,i} \times \hat{\mathbf{e}}_{j,i+1}) \tag{2.5}$$

Then, the Axis-Angle Representation (φ, w) is moved back in the *Rotation Matrix Representation* with respect to the *Global Frame* at the previous time instant *(t-1)*: R^{t-1}.

Once $\Delta\psi$ has been evaluated from the third component of the Rotation Matrix Representation, the capsule absolute orientation ψ about z^w is achieved by summing $\Delta\psi$ at each iteration.

Procedure

The analytical formulation of the time-invariant nonlinear relation between the position vector $\mathbf{d} = [d_x, d_y, d_z]$ and the corresponding magnetic field $\mathbf{B^E} = [B_x^E, B_y^E, B_z^E]$ is crucial to implement a Jacobian-based iterative algorithm for magnetic localisation. This relation, also known as MDR, can be expressed by:

$$\mathbf{B}_i = f(\mathbf{d}_i), \quad f(\mathbf{d}_i) : \mathbb{R}^3 \to \mathbb{R}^3 \tag{2.6}$$

Whenever the capsule moves from $\mathbf{d}_i$ to $\mathbf{d}_{i+1}$, the displacement in position Δd_i corresponds to a variation of the magnetic field ΔB_i according to the MDR. The relation between the two quantity is given by the partial derivative of B_i with respect to Δd_i:

$$\frac{\partial \mathbf{B}_i}{\partial \mathbf{d}} = \nabla_d f(\mathbf{d}_i) = \begin{bmatrix} \frac{\partial B_x}{\partial d_x} & \frac{\partial B_x}{\partial d_y} & \frac{\partial B_x}{\partial d_z} \\ \frac{\partial B_y}{\partial d_x} & \frac{\partial B_y}{\partial d_y} & \frac{\partial B_y}{\partial d_z} \\ \frac{\partial B_z}{\partial d_x} & \frac{\partial B_z}{\partial d_y} & \frac{\partial B_z}{\partial d_z} \end{bmatrix} \tag{2.7}$$

The term $\nabla_d f(\mathbf{d}_i)$ represents the gradient of $\mathbf{B}$ with respect to $\mathbf{d}$ and can be used to compute the magnetic field vector $\mathbf{B}_{i+1}$ corresponding to the new position $\mathbf{d}_{i+1}$. By using a first-order Taylor series approximation, the relation becomes:

$$\mathbf{B}_{i+1} = \mathbf{B}_i + \frac{\partial \mathbf{B}_i}{\partial \mathbf{d}} \Delta \mathbf{d}_i = \mathbf{B}_i + \nabla_d f(\mathbf{d}_i) \Delta \mathbf{d_i} \tag{2.8}$$

By inverting the equation with respect to the position, the MIR can be achieved. The obtained relation can be further simplified by considering that the gradient of a vectorial function coincides with the transpose of the Jacobian.

$$\mathbf{d}_{i+1} = \mathbf{d}_i + \nabla_d f^{-1}(\mathbf{d}_i)\Delta\mathbf{B}_i = \mathbf{d}_i + J_d^{-1}\Delta\mathbf{B}_i \tag{2.9}$$

Relaying on the knowledge of the capsule orientation (so on R_s^E), whenever a new set of magnetic field measurements $\mathbf{B}_i^s$ is available, $\Delta\mathbf{B}_i$ is computed and the estimated capsule position $\mathbf{d}_{i+1}$ is retreived.

Magnetic Field Model

An analytical expression of $\mathbf{B}_{EPM_i} = f(\mathbf{d}_i)$ is essential for the computation of the Jacobian and so for the entire algorithm. It is worth noting that complex models of the magnetic field, as the *Magnetic Current Model (MCM)*, are not suitable for this algorithm because of the impossibility to evaluate their Jacobian in a closed form. On the other side, the new algorithm aims to overcome the limitations of the simplistic *Dipole Model*. A more suitable approach relays on least-squares interpolation to generate functions that can fit with a previously computed dataset of the EPM magnetic field. The steps listed below are followed to evaluate the interpolation functions:

1. Considering the magnetic field of a cylindrical axially magnetized EPM, cylindrical coordinates can be used to establish the relation Ψ defined as:

$$\Psi : (\rho, z) \to (B_\rho^E, B_z^E) \tag{2.10}$$

In can be useful to express Ψ by means of two scalar mathematical functions 2.33 33 to separately represent the radial and axial component of the generated magnetic field.

$$B_\rho^E = \Psi_\rho(\rho, z) : \quad \mathbb{R}^2 \to \mathbb{R} \tag{2.33}$$

$$B_z^E = \Psi_z(\rho, z) : \quad \mathbb{R}^2 \to \mathbb{R} \tag{33}$$

As for the first localisation algorithm [7], the numerical solution of 2.33 and 33 are casted into two matrices (Φ_ρ and Φ_z) that represent the magnetic field numerical solutions for any given position vector $\mathbf{d}$ laying of a restricted 2D plane (Figure 2.10). To reach the wanted interpolations functions, the datasets of Φ_ρ and Φ_z are approximated by means of the equivalent modal representation. Thus, Φ_ρ and Φ_z become:

$$\Phi_\rho = \mathbf{w}(\rho)^T A_\rho \gamma(z) \tag{2.31}$$

$$\Phi_z = \mathbf{w}(\rho)^T A_z \gamma(z) \tag{333}$$

where A_ρ and A_z are is the characteristic matrices of coefficients for the specific magnetic field shape that, together with the two orthogonal bases $\mathbf{w}$ and γ, represents the interpolation functions that best approximate the transformation Φ_ρ and Φ_z over the domain of interest (i.e. the 2D plane). Possible interpolation functions can be searched among the polynomial functions, Chebyshev polynomials, Fourier harmonic basis and composition of these. The best one has been chosen as the one that minimizes the least-squares error between the reference measures of magnetic field given by the matrices Φ_ρ and Φ_z and the approximated values. Thanks to the just obtained function, the MDR is known and the Jacobian of $\mathbf{B}_{EPM}$ can be evaluated. Given J_d, $\Delta\mathbf{B}$ and the old position d_i, the new pose is estimated.

Note that the two equations 2.31 333 can be rearranged in the more compact form below, where Φ collects Φ_ρ and Φ_z, $\mathbf{A}$ includes A_ρ and A_z, while Ω and Γ are the modal basis matrices that constitute the collection of the orthogonal basis $\mathbf{w}$ and γ:

$$\Phi = \Omega \quad \mathbf{A} \quad \Gamma \tag{223}$$

A block diagram (Figure 2.13) of the entire Jacobian-Based Iterative Algorithm is presented below for completeness.

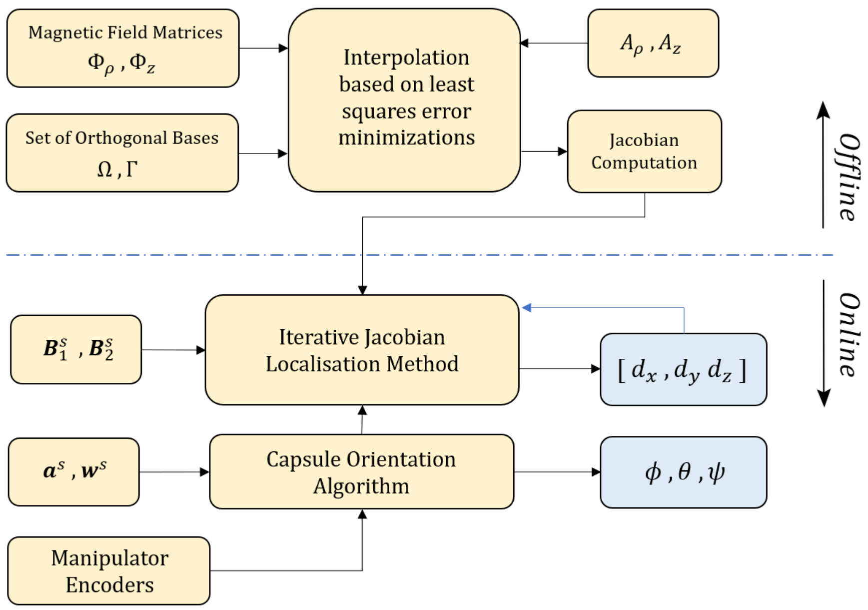


Figure 2.13: Block diagram of the Jacobian-Based Iterative Algorithm. Note that the estimated position $\mathbf{d}_i$ is feedbacked for the next iteration (blue arrow).

Performances Evaluation and Results

To validate the overall algorithm, several different tests are performed by Di Natali *et al.*. Among them, *Capsule Orientation Algorithm Assessment*, *Steady-State Positional Drift Evaluation*, *Robustness to Initialization Errors* and *Robustness to Position Lag*. Although the detailed description of these tests is left to the reader, the results obtained from the *General Assessment* are reported below. During this final experiment, that aims to validate the localisation algorithm for a generic trajectory of the EPM, the wireless capsule is maintained fixed while the external magnet coordinates span in the range of -10cm÷10cm along x^w and y^w, and from 6cm÷12cm along z^w away from the capsule position.

The average pose estimation errors (both absolute and relative) e and the corresponding standard deviations σ presented in the radial component d_ρ, axial component d_z and azimuth component d_δ are reported in Table 2.2

	$e_{d_\rho} \pm \sigma_{d_\rho}$	$e_{d_z} \pm \sigma_{d_z}$	$e_{d_\delta} \pm \sigma_{d_\delta}$
Absolute Error	6.2 ± 4.4 mm	6.9 ± 3.9 mm	5.4 ± 7.9 °
Relative Error	5.7 ± 7.6 %	7.0 ± 4.9 %	Not specified

Table 2.2: Experimental estimation errors and standard deviations.

Drawbacks and Limitations

The presented localisation algorithm features three main drawbacks and limitations. The first one concerns the use of the least-squares interpolation: in some specific regions of the magnetic field map Ω, the relative error in the estimate capsule position $\hat{\mathbf{d}}$ is grater than 20% and so causes an imprecise localisation. The second limitation is strictly linked to the integrative method that is used both for position and yaw angle's estimation. By summing noisy measurements, *drift* may become an issue over time and affects the precision of the estimation. As previously mentioned, the proposed algorithm could be integrated with an absolute localization strategy to avoid this phenomenon. As for the first algorithm by Di Natali *et al.*, the singularity regions of the magnetic field represent a strong drawback: considering the endoscopic capsule moving along a circumference on the singularity plane of the MFE, the components of the sensed magnetic field do not change in time. So, even though the position changes from $\mathbf{d}_i$ to $\mathbf{d}_{i+1}$, no variations $\Delta\mathbf{B}_i$ are perceived and the algorithm returns the previous position $\mathbf{d}_i$.

2.3.4 Particle Filter based Pose Estimation Algorithm

The algorithms previously mentioned face significant challenges in their path to clinical adoption: among them, the presence of regions of magnetic field singularity and the need of accurate initialization of the capsule's pose. In 2018, Addisu Z Taddese *et al.* proposed the final version of the localization algorithm that is actually used in Magnetic Flexible Endoscope. In this system the capsule's pose is updated every 10 ms, thus requiring an operating frequency of 100 Hz.

The motivation behind this new estimation method relays on a study, carried on by the STORM Lab UK and STORM Lab USA, that employs the point-dipole magnetic field model to show that singular regions exist in areas of the workspace where the capsule in *nominally* located during magnetic actuation. The cause of this limiting phenomenon has been addressed to the use of a single source of magnetic field. The novel hybrid approach for the pose estimation subsystem is able to solve the problems of magnetic singularities and attitude's initialization thanks to an addictional electromagnetic coil. Moreover, it allows advances in closed-loop control of a tethered capsule endoscope by means of a particle filter based algorithm.

The description of all the hardware components is available in Section 2.2 while a detailed description of the used set-up, software and firmware environment is presented hereby.

Electromagnetic Coil and Singularity regions of the magnetic field

Magnetic singularities occur when the same components of the magnetic field are associated to different points of the selected workspace (Section 2.3). For the EPM this set of points identifies a plane orthogonal to its magnetization axis, passing through its center. The idea behind this novel approach is to add a source of magnetic field to reduce the singularity plane to a line; From the mathematical point of view, a second source of magnetic field introduces three additional equations that can be used to solve the Inverse Magnetic Relationship ($\mathbf{B}_i \rightarrow \mathbf{d}_i$) for the points of the singularity plane. Note that this result can be achieved if and only if the two magnetic fields can be decoupled and clearly distinguished. So, given the DC magnetic field generated by the EPM, the second source must produce a not-constant $\mathbf{B}$.

Let's suppose to place a further cylindrical magnet, with its own singularity plane, in a different direction with respect to the EPM. Thanks to $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$, the vast majority of the points of the workspace is univocally identified by a specific set of magnetic field components. The only exception is represented by the singularity line given by the intersection of the two planes: the same resultant magnetic field identifies two symmetrical points (red points in Figure 2.14) with respect to the center of the magnets.

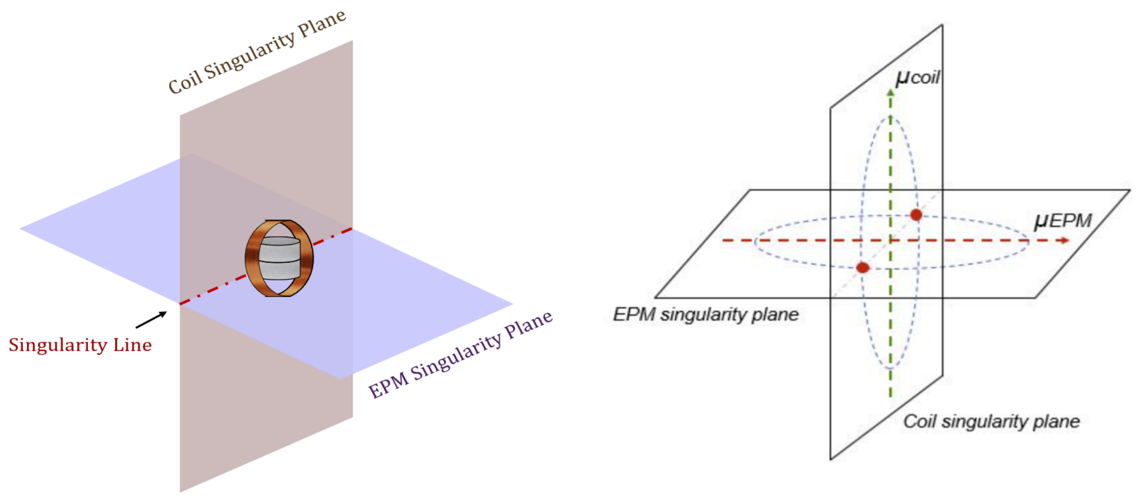


Figure 2.14: Singularity line given by the intersection of the two singularity plane (Left). On the singularity line, two symmetric points are identified by the same magnetic field (Right).

Given the infeasibility to physically place two cylindrical magnets with the same center in the same volume, an electromagnetic coil has been designed by Marco Pirotta *et al.*

and placed around the EPM with its equivalent magnetization axes orthogonal to the one of the EPM. This arrangement makes the magnetic field of the coil always orthogonal to the one generated by the EPM. In the final set-up, a mechanical enclosure for the coil is 3D-printed and designed to slide along the outer edges of the EPM: this solution allows to achieve the smallest volume for the whole assembly while minimizing the risk for collision with the manipulator itself. For this reason, the EPM and coil may not be centered in the same location: the center of the coil is shifted of 45mm away from the center of the EPM along x^E. However, it is worth noting that this distance has not a relevant impact on the performances of the pose estimation system.

However, a further design step is needed to fully exploit the advantages introduced by the coil. In fact, if the coil was energized by a DC voltage, it would generate a constant magnetic field. Thus, the Hall-effect sensors embedded in the capsule would detect the resultant magnetic field given by the superposition principle $\mathbf{B}_{tot} = \mathbf{B}_{EPM} + \mathbf{B}_{coil}$. So, the use of a constant current source to energize the coil implies the impossibility to separately detect $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$ and so precludes the availability of the three additional equations. A square wave voltage at a frequency of 300Hz, together with a signal processing technique (Goertzel Algorithm) has been employed to overcome this limitation. A more detailed explanation of their roles is presented later in the section.

To completely remove the singularity regions, a second electromagnetic coil could be added. In order to be effective, it should be oriented orthogonally with respect to the others magnetic sources while the frequency of the new oscillating field should be different from 300Hz. Even though the problem could be solved by the introduction of this third source of magnetic field, this choice would result in a waste of resources and in a too complex device. A more reasonable solution is instead achieved through a purpose-built piece of code: since the localization algorithm fails whenever the capsule moves toward the singularity line, a control of proximity is activated when the endoscopic tip enters in a predefined area around the same line of intersection. If enabled, the algorithm estimates the position of the capsule as the mean of the possible positions in the surroundings. As for the yaw angle, it is kept constant and equal to the last reliable estimated value.

Moreover, the use of an electromagnetic coil implies an important advantage in terms of modelling: thanks to simple mathematical manipulations, an equivalent residual magnetization of the coil can be evaluated. Thus, the same model of the magnetic field can be used both for the EPM and the coil.

Magnetic Field Model

Even though the point-dipole magnetic field model has been used to study the effect of singular regions for clinical adoption, the model developed by Norman Derby *et al.* [5] has been chosen to achieve a more precise and accurate localisation. The model relays on the physical principle that wire-wound solenoids and cylindrical magnet can be approximated to ideal, azimuthally symmetric solenoids. More specifically, it presents an exact solution for the magnetic field of an ideal solenoid based on the following Generalized Complete Elliptic Integral C:

$$C(kc,q,c,s) = \int_0^{\frac{\pi}{2}} \frac{(c*cos^2\varphi + s*sin^2\varphi)}{(cos^2\varphi + q*sin^2\varphi)\sqrt{cos^2\varphi + k_c^2*sin^2\varphi}} d\varphi \tag{2.11}$$

Where kc, q, c and s are functions depending on the EPM design parameters and on the cartesian coordinates of the point where the magnetic field is measured. φ is instead the azimuthal angle that defines the position of the point in space in cylindrical coordinates. The Visual BASIC function shown in Appendix C has been developed by Derby *et al.* [5] and guarantees an efficient computation of C. The function can be easily implemented in any other programming languages, including MatLab and Python.

More specifically, this algorithm allows to compute two offline lookup tables that map the magnetic field (B_ρ and B_z) with the position of the point mass. With reference to Figure 2.15, given the cylindrical coordinates of a point around the magnet ($\rho = d_\rho = \sqrt{d_x^2 + d_y^2}$ and d_z), the cylindrical coordinates of $\mathbf{B}_{EPM}$ are computed by means of the set of formula below. Equations (5.23) up to (5.23) are referred to a cylinder magnet (or coil) of length *2b* and radius *a*.

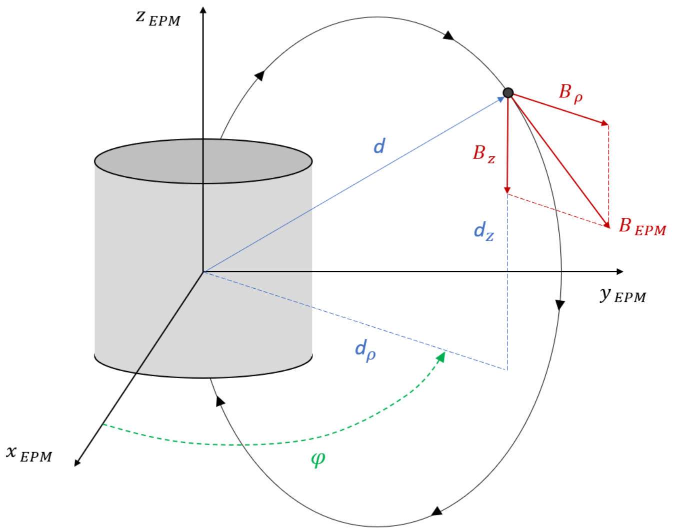


Figure 2.15: Cylindrical Coordinates of the EPM magnetic field.

$$B_\rho = B_0 * [\alpha_+ * C(k_+, 1, 1, -1) - \alpha_- * C(k_-, 1, 1, -1)] \tag{5.23}$$

$$B_z = \frac{B_0 * a}{a + \rho} * [\beta_+ * C(k_+, \gamma^2, 1, \gamma) - \beta_- * C(k_-, \gamma^2, 1, \gamma)] \tag{5.23}$$

$$B_0 = Remanence/\pi \tag{2.12}$$

$$z_\pm = z \pm b \tag{2.13}$$

$$\alpha_{\pm} = \frac{a}{\sqrt{z_{\pm}^2 + (\rho + a)^2}} \tag{2.14}$$

$$\beta_{\pm} = \frac{z_{\pm}}{\sqrt{z_{\pm}^2 + (\rho + a)^2}} \tag{2.15}$$

$$\gamma = \frac{a - \rho}{a + \rho} \tag{2.16}$$

$$k_{\pm} = \sqrt{\frac{z_{\pm}^2 + (a - \rho)^2}{z_{\pm}^2 + (a + \rho)^2}} \tag{2.17}$$

The same model can be applied to an electromagnetic coil with n turns per unit length, carrying a current of amplitude I. To do so, $\mathbf{B}_0$ must be re-defined as shown below. However, since the code of Appendix C is referred to a magnet with magnetization axis along $\mathbf{z}^E$, particular attention must be paid during the modelling of the magnetic field generated by the coil whose magnetization axis is directed along $\mathbf{x}^E$.

$$B_0 = \frac{\mu_0}{\pi} nI \tag{2.18}$$

As in the previous localisation algorithms, this model is employed to generate magnetic field density maps of the cylindrical coordinates both for $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$. The mapped workspace is chosen coherently with the one used in a clinical scenario.

Generation of the Oscillating Magnetic Field

A purpose-built circuit, composed by an STM32 Nucleo board and a custom-made Driver Circuit, is employed for the generation of the oscillating magnetic field of the coil. As shown in Figure 2.16, the Driver circuit is based on a low pass filter, an amplification stage, a voltage regulator and a H-Bridge. The latter needs two constant and three Pulse-width modulation (PWM) signals coming from the Nucleo board.

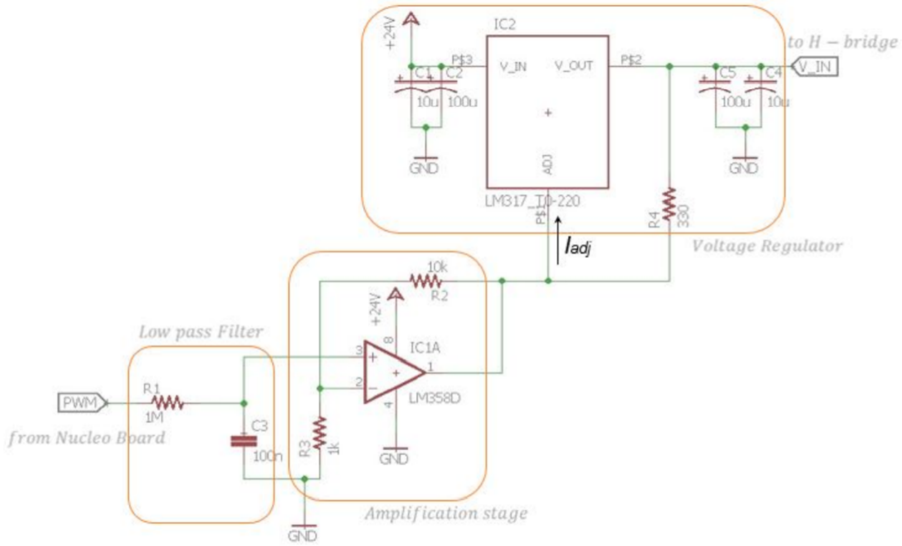


Figure 2.16: Electrical circuit used to generate the square wave voltage on the coil ends . Credit: Marco Pirotta.

As for the five used signals:

- Two *HIGH* signals ($V_{DC} = 5V$) are needed for the activation of the two sides of the H-Bridge mounted on the Driver Circuit;
- Two PWMs signals at the frequency of $f = 300Hz$ and duty cycle of $D.C. = 50\%$ are used to switch the polarity of the output voltage applied to a coil, so to select the wanted *frequency* of 300Hz. It is worth mentioning that the two PWMs must be shifted in phase of 180° to ensure the correct working principle of the H-Bridge;
- The remaining PWM at the frequency of $f = 1kHz$ is sent in input to an analogical stage with filtering and amplification purposes that guarantees the correct DC input voltage to the H-Bridge and so the *amplitude* of the square wave voltage between the coil ends. While its frequency is selected for the design of a low pass filter, the D.C. allows the regulation of the amplification of the stage. The output DC voltage is the input signal of the H-Bridge that directly regulates the amplitude of the oscillating magnetic field. An additional Voltage Regulator is placed after the amplification stage to guarantee a steady output voltage on the coil ends. An amplitude of 17.3V has been choosen to ensure (in the selected workspace) values of magnetic field always higher than the electromagnetic noise.

Experimental tests are performed to evaluate the correctness of the designed circuit and show the presence of unwanted oscillations in the time instants when the switches of the H-Bridge take place. The measured voltage across the coil ends is shown in Figure 2.17.

For a more detailed explanation of the electrical circuit design, please refer to the work of Pirotta *et al.* [23].

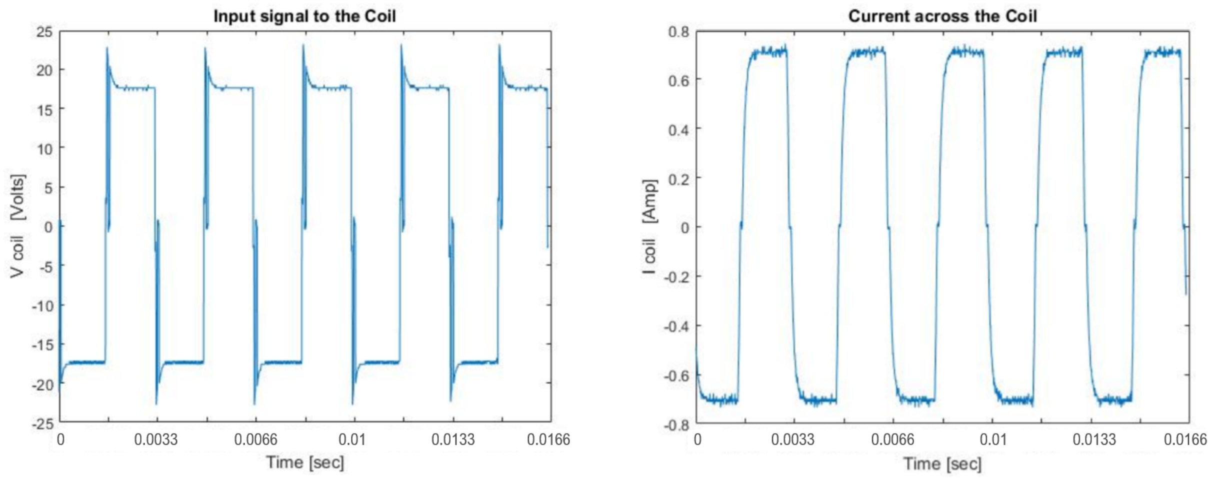


Figure 2.17: Voltage and current across the coil ends measured by means of an oscilloscope.

Procedure

The new localisation algorithm, developed by Addisu Taddese *et al.* [28], is based on the same set of input measurements of the previous versions; It implies linear accelerations $[a_x, a_y, a_z]^T$, angular velocities $[w_x, w_y, w_z]^T$ and a pair of magnetic field measurements $\mathbf{B}_1^E$ and $\mathbf{B}_2^E$. It is worth mentioning that the sensed magnetic field is the resultant of $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$. This method allows the estimation of the capsule pose $[p_x, p_y, p_z, \phi, \theta, \psi]$ at the refresh rate of 100Hz. The employed reference frames are the ones shown in Figure 2.6.

As previously done by Christian Di Natali *et al.*, the first step of the algorithm is intended to compute two out of the three angles that completely define the capsule orientation with respect to the Global Frame. This result is achieved thanks to the Mahoney Filter by Mahoney *et al.* [19] that retrieves the roll ϕ and pitch θ angles by integrating the gyroscope readings and compensating the long term gyro drift by means of the acceleration measurements.

Procedure: Goertzel Algorithm

As it will be clarified later, the estimation algorithm needs separate measurements of $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$. So, recalling the superposition principle between the static $\mathbf{B}_{EPM}$ and the oscillating (i.e. AC) $\mathbf{B}_{coil}$, the measurements coming from the Hall-effect sensors cannot be directly sent as input of the filter. Thanks to the different frequencies of the two fields, the Goertzel Algorithm (Goertzel, 1958; Turner, 2003) [10] can be applied to separate the static component from the AC one. Moreover, this filter allows the computation of the equivalent DC value of the AC component. Unlike other algorithms for tone detection, this filter allows to *concurrently* perform a frequency analysis, needed to distinguish $\mathbf{B}_{EPM}$ from $\mathbf{B}_{coil}$, and to measure the magnitude of a signal in correspondence of a single sampling frequency (i.e. 300Hz). To make the Goertzel Algorithm compliant with the MFE, it is designed to retreive in output the equivalent separate magnetic field measurements at 100Hz.

For the application of this filter, the *sampling rate*, the *block size* and the *bin width* must be defined. These three parameters are not independent.

- The sampling rate $f_{sampling}$ is selected by considering the Nyquist Theorem. Given the oscillating magnetic field at 300 Hz, $f_{sampling} = 600Hz$ represents the lower bound. To obtain an higher number of samples and so a more precise tone detection, the Hall-effect sensors are sampled at a rate of 18 kHz via 16-bit analog-to-digital converter embedded in the endoscopic tip;
- The *bin width* is equivalent to the target frequency with respect to which the tone detection must be performed. In this application, it coincides with $f_{target} = 100Hz$;
- The *block size* N can be easily derived from the two previous values since it coincides with the number of samples that are available at each iteration of the algorithm. For this application, $N = \frac{f_{sampling}}{f_{target}} = 180$.

The Goertzel Algorithm is implemented in the STM32 Nucleo board and involves a two steps procedure: an *intermediate process* is performed whenever a sample of the measured magnetic field is received (i.e. each 3.33 ms); once the *block size* is reached, the tone detection takes place. Two instances of this algorithm are run with 10 ms and 30 ms windows, respectively. Although the output from the 10 ms instance is less reliable, it provides the desired update rate for the real-time pose estimation. The mathematical formulation of the algorithm is reported below:

1. The following set of parameters is defined before the beginning of the algorithm.

$$k = 0.5 + \frac{N f_{target}}{f_{sampling}} \tag{2.19}$$

$$w = \frac{2\pi}{N} k \tag{2.20}$$

$$ss = sin(w) \tag{2.21}$$

$$cc = cos(w) \tag{2.22}$$

$$h = 2cc \tag{2.23}$$

2. Whenever a magnetic field sample is received by the Nucleo board, the intermediate process begins. s_0 and s_1 are initialized to zero at the beginning of each tone detection (i.e. each 10ms).

$$s_0 = hs_1 - s_2 + new_{sample} \tag{2.24}$$

$$s_2 = s_1 \tag{2.25}$$

$$s_1 = s_0 \tag{2.26}$$

3. When the block size N=180 is reached, the DC magnitude corresponding to 300 Hz is evaluated as follow:

$$Re = s_1 - s_2 cc \tag{2.27}$$

$$Im = s_2 ss \tag{2.28}$$

$$Magnitude = \sqrt{Re^2 + Im^2} \tag{2.29}$$

$$Phaseshift = \frac{Im}{Re} \tag{2.30}$$

   As for the sign, it can be computed by evaluating the phase between the input voltage of the coil and the measured oscillating magnetic field. Taking into consideration that the two signals are square waves, if the measured *phase shift* is lower than 90°, it means that the two signals are in phase, thus the measured magnetic field will have a negative sign. Viceversa, when the two signals are out of phase (i.e *phase* $> 90°$) a positive sign will be associated to the reconstructed value. From the experimental point of view, this result can be reached every 10ms by concurrently measuring the oscillating field and re-initializing the two PWMs used to control the H-Bridge: this last operation allows to generate a voltage at the coil ends that always starts with a positive sign. When a new set of magnetic field measurements are received by the Nucleo Board, s_0 and s_1 are re-initialized to zero and the algorithm is repeated from step 2.

Procedure: Particle Filter

After these preliminary steps, which allow to reduce the order of the system (i.e. the unknown parameters) and decouple the magnetic field measurements, an iterative process is performed to estimate the remaining position vector and yaw angle ψ. These identify the state of the system: $\mathbf{x} = [p_x, p_y, p_z, \psi]^T \in \mathbb{R}^{4\times1}$. The iterative algorithm is based on the Sampling Importance Resampling (SIR) variant of the Particle Filter *(Gordon et al., 1993)*. In this method, the posterior distribution $p(\mathbf{x}_k|\mathbf{z}_{1:k})$ of the state $\mathbf{x}_k$ at time k conditioned on a time series of measurements $\mathbf{z}_{1:k} = \{\ \mathbf{z}_i, i = 1, 2, ..k\ \}$ is represented by a set of particles; A specific weight $w_{k,i}$ is assigned to each particle.

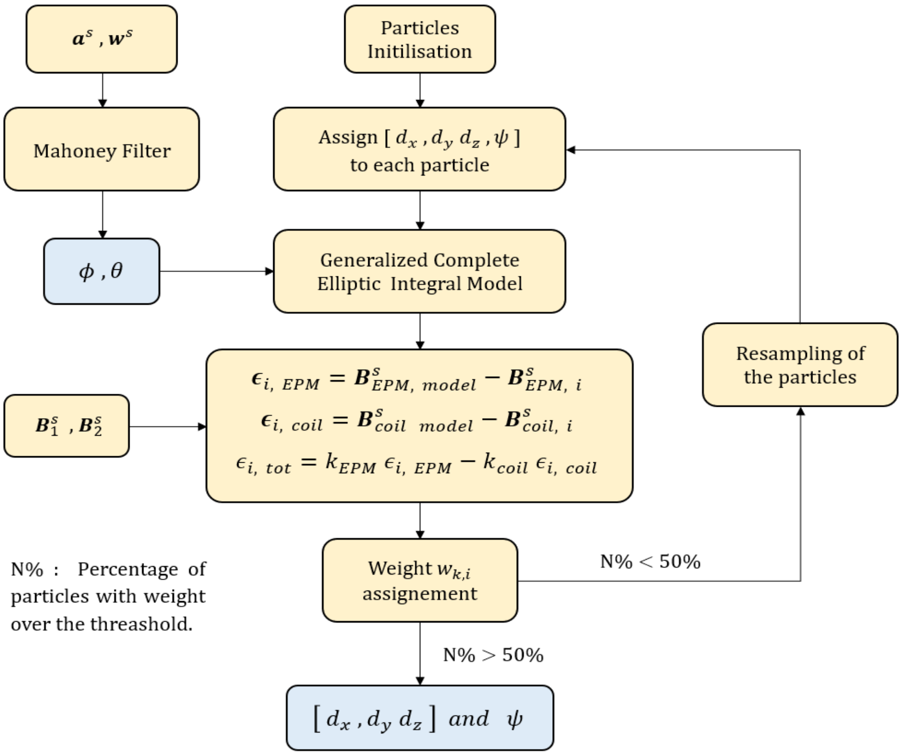


Figure 2.18: Block diagram of the real-time pose estimation algorithm based on the Particle-Filter.

At each iteration, the SIR algorithm performs a prediction which consists of creating a new set of particles based on the prior density (i.e. the weights $w_{k,i}$) and the process model. For this specific application, the algorithm needs as input the computed roll ϕ and pitch θ angles and the *separate* measurements of the EPM and coil magnetic fields. Once the sensed magnetic fields have been decoupled, the Particle-Filter algorithm begins to obtain an estimate of the state: $\hat{\mathbf{x}} = [\hat{p}_x, \hat{p}_y, \hat{p}_z,\ \hat{\psi}]$. Step 1 to XX are repeated each 10ms (i.e. 100Hz). The block diagram above sums up the steps of the whole algorithm.

1. A predefined number of particles are generated by the filter; for this application, 1200 particles ensure a good compromise between performances and computational effort. Each particle is intended to simulate a possible pose of capsule inside the whole workspace. For the MFE, random values of p_x, p_y, p_z and ψ are assigned to each particles; this solution allows to cover uniformly the cubic workspace around the EPM. As for the roll ϕ and pitch θ angles, they are assumed known and equal to the computed ones.

2. To each particle, the corresponding values of EPM and coil magnetic field are assigned: while the position of the particle is used to compute the cylindrical coordinates in the EPM Frame of $\mathbf{B}_{EPM,model}$ and $\mathbf{B}_{coil,model}$ by means of the generalized Complete Elliptic Integral, its orientation allows to move these measurements in the Capsule Frame. By following this approach, specific values of $\mathbf{B}_{EPM,model}$ and $\mathbf{B}_{coil,model}$ are spread in the whole workspace as if they were sensed by the capsule with that particular pose.

3. To assign a weight $w_{k,i}$ to each particle and so to enable the resampling process, the measurements of the separate magnetic fields $\mathbf{B}_{EPM,i}i$ and $\mathbf{B}_{coil,i}$ (i.e. the output of the Goertzel Algorithm) are considered. Knowing both the ideal and the measured values, the differences are computed for each particle as:

$$\epsilon_{i,EPM} = \mathbf{B}_{EPM,model} - \mathbf{B}_{EPM,i} \tag{2.31}$$

$$\epsilon_{i,coil} = \mathbf{B}_{coil,model} - \mathbf{B}_{coil,i} \tag{2.32}$$

For each particle, these two vector are used to evaluate the total error:

$$\epsilon_{i,tot} = k_{EPM}\epsilon_{i,EPM} + k_{coil}\epsilon_{i,coil} \tag{2.33}$$

where $\epsilon_{i,EPM}$ and $\epsilon_{i,coil}$ are the norm of the two vectors computed above. k_{EPM} and k_{coil} are instead real numbers chosen to properly weight the two contributions: the weight given to $\epsilon_{i,EPM}$ is ten times higher than k_{coil} due to the much stronger strength of the EPM magnetic field with respect to the one of the coil. Based on the final error, a simple consideration can be done: the higher is $\epsilon_{i,tot}$, the worst is the estimate provided by the *i-th* particle.

4. Based on $\epsilon_{i,tot}$, a specific weight $w_{k,i}$ is given to each particle: high weights are associated to low errors, so to particles identified by a pose similar to the one of the endoscopic tip. Viceversa, the bigger is the total error $\epsilon_{i,tot}$, the lower is the weight. Once a weight is assigned to all the 1200 particles, the resampling process takes place: all the N particles whose weight is below a predefined threashold value, are deleted and re-sampled in the region of the workspace identified by the highest weights.

5. Steps 1-4 are repeated until more than the 50% of the particles (N=601 particles) has a weight over the threashold. When this last condition is satisfied, the estimate of the yaw $\hat{\psi}$ is chosen equal to the one of the most weighted particle while the estimate position $\mathbf{p}$ of the capsule is computed by averaging the cartesian components $p_{x,i}, p_{y,i}, p_{z,i}$ of the selected particles:

$$\hat{\mathbf{p}} = \left[\frac{\sum_{i=1}^{N} p_{x,i}}{N}, \frac{\sum_{i=1}^{N} p_{y,i}}{N}, \frac{\sum_{i=1}^{N} p_{z,i}}{N} \right] \tag{2.34}$$

Performances Evaluation and Results

Tests in static and dynamic condition are performed by Taddese *et al.* to evaluate the performances of the developed algorithm and to enable a comparison with the previous version.

Static Tests

The first set of tests in static condition sees the endoscopic tip inserted in a custom-made 3D printed enclosure and secured in a known position by means of a secondary robotic manipulator. As shown in Figure 2.19, the EPM is moved in 25 preselected points on a spiral trajectory belonging to an hemisphere: it allows to test the algorithm for increasing distances. The pose estimates is recorder for 30 seconds in each point. Moreover the same test is performed varying the radius of the hemisphere from 150 to 200 mm. The average accuracy (mean ± standard deviation) of position estimates for these static tests is reported in Table 2.3 below; note that the average accuracy of the roll and pitch angle depends on the Mahoney Filter and not on the Particle Filter's implementation.

$\Delta x \pm \sigma_x$ (mm)	$\Delta y \pm \sigma_y$ (mm)	$\Delta z \pm \sigma_z$ (mm)	$\Delta \phi \pm \sigma_\phi$ $(°)$	$\Delta \theta \pm \sigma_\theta$ $(°)$	$\Delta \psi \pm \sigma_\psi$ $(°)$
1.57 ± 1.41	4.05 ± 1.67	2.24 ± 0.95	1.02 ± 0.59	-0.94 ± 0.65	-5.39 ± 0.18

Table 2.3: Static tests, spiral trajectory: experimental estimation errors and standard deviations.

The second set of static testes is intended to evaluate the performances of the algorithm both in the singularity plane of the EPM. For this tests, the capsule is maintained fixed in a position while the EPM is moves on a grid 200 × $50mm^2$ of 25 coplanar points belonging to the singularity plane. The pose estimates is recorder for 30 seconds in each point. Figure 2.19 shows the configuration for the test. The average accuracy (mean ± standard deviation) of position estimates for these static tests is reported in Table 2.4.

$\Delta x \pm \sigma_x$ (mm)	$\Delta y \pm \sigma_y$ (mm)	$\Delta z \pm \sigma_z$ (mm)	$\Delta \phi \pm \sigma_\phi$ $(°)$	$\Delta \theta \pm \sigma_\theta$ $(°)$	$\Delta \psi \pm \sigma_\psi$ $(°)$
2.85 ± 0.80	3.74 ± 1.53	1.67 ± 0.88	0.73 ± 0.60	-1.69 ± 0.15	-3.76 ± 0.12

Table 2.4: Static tests, EPM singularity plane: experimental estimation errors and standard deviations.

The last set of static tests are addressed to check the performances of the localisation algorithm on the singularity line identified by the intersection of the two singular planes. A 30s test is performed on 10 points on this line. The results in terms of average error and standard deviation are reported in Table 2.5.

$\Delta x \pm \sigma_x$ (mm)	$\Delta y \pm \sigma_y$ (mm)	$\Delta z \pm \sigma_z$ (mm)	$\Delta\phi \pm \sigma_\phi$ $(^\circ)$	$\Delta\theta \pm \sigma_\theta$ $(^\circ)$	$\Delta\psi \pm \sigma_\psi$ $(^\circ)$
1.21 ± 0.18	4.85 ± 1.34	5.10 ± 0.68	0.75 ± 0.10	-2.05 ± 0.13	1.08 ± 0.06

Table 2.5: Static tests, singularity line: experimental estimation errors and standard deviations.

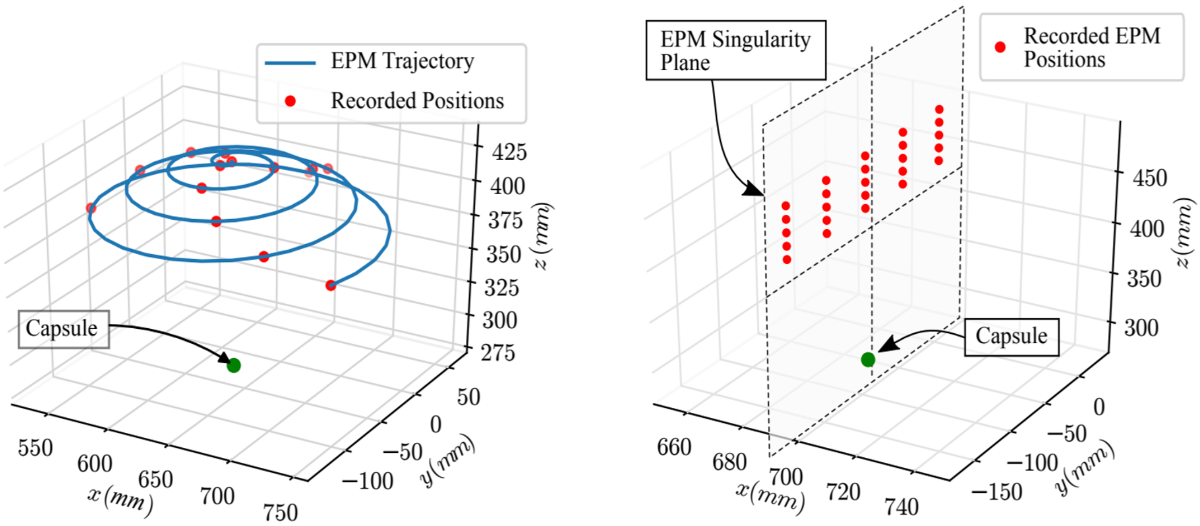


Figure 2.19: Static tests: static test performed in 25 point along a spiral trajectory (Left). Static test performed in 25 points on the Singularity plane of the EPM (Right). Credit: Addisu Taddese.

The results of the last two set of static tests shows that the system performs well even in the singularity regions of the workspace and so, allows a more robust localisation with respect to previous versions. Moreover, the overall accuracy of the system for these set of tests was equivalent or better than the algorithm described in Di Natali *et al.* (2013) [7] and Di Natali *et al.* (2016) [6].

Dynamic Tests

To evaluate the performance in dynamic conditions, two tests are conducted: the first one in *Static-Dynamic* conditions, the second in *Dynamic-Dynamic* conditions.

As for the *Static-Dynamic* tests, the EPM is maintained fixed while the capsule is moved by a secondary robotic manipulator along a predefined trajectory with a speed ranging from 10 mm/s to 50 mm/s. The test is mainly focused in testing the ability of the Particle Filter to estimate the capsule pose by means of the Random Walk process model (Explained in details in Section 4.2-Process model). The results show how the estimation error increases with the relative speed between the endoscopic tip and the EPM; However, this drawback does not limit the use of the localisation algorithm due to the low speed used in a clinical scenario.

The final set of tests in *Dynamic-Dynamic* conditions are the ones that reflect the most a real clinical application of the MFE. More specifically, both the EPM and the capsule are moved along two identical trajectories at a relative distance of 20cm. This experimental set-up is endured by the secondary robotic manipulator and simulates a perfect magnetic coupling between the EPM and the endoscopic tip. As before, the same experiment is repeated at the speed of 10 mm/s , 25 mm/s and 50 mm/s; However, while in the previous set of tests the absolute speed of the capsule coincides with the relative speed with respect to the EPM, in the *Dynamic-Dynamic* tests, the relative speed is approximately null. As shown in Figure 2.20, the chosen trajectory in the *Dynamic-Dynamic* conditions aims to mimic the shape of the human colon. Even though both tests have been performed at different speeds, the result obtained with 10 mm/s are reported only due to the similarity with a traditional colonoscopy: Table 2.6 for Static-Dynamic conditions, Table 2.7 for Dynamic-Dynamic conditions. Finally, the algorithm shows good performance during the closed loop control of the endoscopic tip in a colon phantom.

$\Delta x \pm \sigma_x$ (mm)	$\Delta y \pm \sigma_y$ (mm)	$\Delta z \pm \sigma_z$ (mm)	$\Delta\phi \pm \sigma_\phi$ $(^\circ)$	$\Delta\theta \pm \sigma_\theta$ $(^\circ)$	$\Delta\psi \pm \sigma_\psi$ $(^\circ3$
-3.39 ± 6.76	-4.84 ± 5.23	4.06 ± 1.91	-0.96 ± 2.30	0.29 ± 1.73	-0.37 ± 2.84

Table 2.6: Static-Dynamic test at 10mm/s: estimation errors and standard deviations.

$\Delta x \pm \sigma_x$ (mm)	$\Delta y \pm \sigma_y$ (mm)	$\Delta z \pm \sigma_z$ (mm)	$\Delta\phi \pm \sigma_\phi$ $(^\circ)$	$\Delta\theta \pm \sigma_\theta$ $(^\circ)$	$\Delta\psi \pm \sigma_\psi$ $(^\circ)$
-2.05 ± 5.00	-1.60 ± 3.76	1.60 ± 0.57	-1.76 ± 1.15	0.07 ± 1.80	-0.27 ± 2.65

Table 2.7: Dynamic-Dynamic test at 10mm/s: estimation errors and standard deviations.

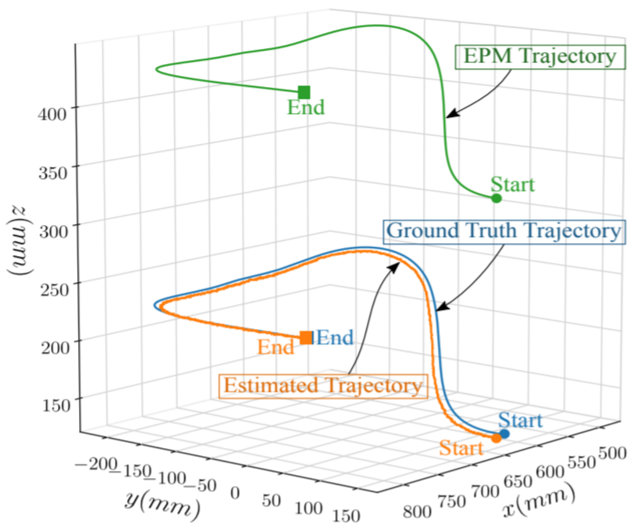


Figure 2.20: Dynamic tests: trajectory designed to mimic the human colon used for the dynamic-dynamic test. Credit: Addisu Taddese.

2.4 Motivations for a new Localisation Algorithm

The results of the three localisation algorithms clearly highlight that the use of the Particle Filter (Taddese *et al.*, 2018) allows for a more accurate localisation and the added electromagnetic coil almost completely solves the problems of magnetic singularities and attitude initialization. So, the introduction of this second source of magnetic field has opened new opportunities for the design of computationally less expensive but equally robust localisation algorithms. Moreover, the high computational cost of the Particle Filters (or sequential Monte Carlo methods) represents a relevant limitation for real-time application, as the MFE, and requires the computational power of a PC. These observations further motivate the design of a computational less expensive magnetic localisation algorithm that could bring benefits in the design of the platform: all the computations needed for the estimation of the pose could be ported in an embedded system (i.e. development board) that, thanks to its smaller dimensions, could replace an ordinary PC and enables a more compact hardware environment of the device. Moreover, a new localisation algorithm that runs on an additional embedded system could be also helpful if introduced as a *redundant system*: it can be employed to duplicate both the hardware and the magnetic localisation algorithm with the intention of increasing the system safety in case of fail of the main computational unit. This last feature would enable the MFE to be more compliant with the regulation that concerns the development of medical devices.

Even though this work aims to achieve a precise localisation while concurrently reducing the computational effort required by the Particle Filter, the benefits of using an estimator to filter out the measurements noise must not be neglected. As already suggested by Christian Di Natali *et al.* (2013), real-time filtering techniques, as Kalman Filters, may be a viable option to guarantee good accuracy of pose detection.

The estimation method developed in this work represents a proof of concept for the application of Kalman Filters to achieve robust performances. This feature is essential for the MFE to allow a precise localisation of the endoscopic tip despite of the presence of the electromagnetic noise that affects the measurements of the six Hall-Effect sensors embedded in the real capsule endoscope. The new algorithm aims to obtain promising results with a significantly lower computational effort than the one actually required by the Particle Filter based method.

Chapter 3

The Magnetic Flexible Endoscope Simulator

As it has been previously mentioned, the project is focused on the study and design of a new localisation algorithm for the Magnetic Flexible Endoscope. As the one currently implemented on the platform, the new numerical method must provide as output an estimation of the position and orientation (i.e. pose) of the endoscopic capsule starting from the data of acceleration, angular velocity and magnetic field measured by the sensors mounted on the capsule. By considering the wide range of devices that could surround the Magnetic Flexible Endoscope in a clinical scenario.

The physical robotic platform and the endoscopic capsule are essential to develop the new localisation algorithm. Unluckily, the start date of this Final Project coincides with the beginning of the COVID-19 pandemic. For this reason, an unplanned ROS Simulator of the Magnetic Flexible Endoscope has been designed and developed to overcome the COVID-19 Containment measures that do not allow visiting students to get access to the STORM Lab. The Simulator aims to accurately mimic all the features of the robotic platform including the magnetic field generated by the EPM, the one produced by the coil, the LBR iiwa manipulator, the endoscopic capsule and the arrangement of the sensors inside it.

The hardware and software features of the MFE Simulator are presented and explained in details in this chapter. It is worth mentioning that the work described here represents an essential step for design of the new localisation algorithm described in Chapter 4. An highly detailed analysis of the code is available in Appendix D.

3.1 The ROS Environment

The Kinetic distribution of the Robot Operating System (ROS) has been choosen for the development of MFE Simulator to make it consistent with the ROS nodes that are used with the physical system. Ubuntu 16.04 (Xenial) release has been selected as platform to fully exploit the capabilities of ROS Kinetic. In the following description of the project it is assumed that the reader is familiar with the concept of ROS node, ROS message and ROS topic.

The ROS node *macros_localization_node.cpp* developed by Addisu Taddese *et al.* [28] for the magnetic localisation of the endoscopic capsule is considered to understand the features of the MFE Simulator. By analysing the code, it can be shown how the algorithm relays on the set of ROS messages published on the ROS Topics listed below.

ROS Topic Name	ROS Message Type
/MAC/imu/raw	sensor_msgs/Imu
/MAC/epm/mfs1/raw	sensor_msgs/MagneticField
/MAC/epm/mfs2/raw	sensor_msgs/MagneticField
/MAC/coil/mfs1/raw	sensor_msgs/MagneticField
/MAC/coil/mfs2/raw	sensor_msgs/MagneticField
/MAC/coil10/mfs1/raw	sensor_msgs/MagneticField
/MAC/coil10/mfs2/raw	sensor_msgs/MagneticField

Table 3.1: Table x

This observation is crucial to understand the needed output of the simulator: a new localisation algorithm can be design if and only if the same set of messages are generated and published on the same topics. The publication frequency of 100Hz and arrangement of the sensors inside the capsule must be considered to obtain a perfect match between the simulator and the real platform.

As for the input, the user should have the possibility to freely move the simulated capsule and change its orientation in the space surrounding the EPM. This requirement allows to have the same degrees of freedom (i.e. 6 dof) of the real endoscope.

During the first step of the project, the ROS node developed by *trainman419* [30] has been modified to allow the complete keyboard tele-operation of the simulated capsule. Table 3.2 shows the keyboard-key initially used to control the linear and angular velocities of the simulated endoscope tip.

Key	Action
t	Capsule moves along +x
b	Capsule moves along -x
i	Capsule moves along +z
,	Capsule moves along -z
u	Capsule pitch angle θ increases
o	Capsule pitch angle θ decreases
j	Capsule yaw angle ψ increases
l	Capsule yaw angle ψ decreases
k	Capsule stops

Table 3.2: Table xx

According to the pressed keyboard keys, the pose of the capsule changes and the corresponding inertial values are generated in the capsule reference frame. The simulated device can be moved in the space according to a uniformly accelerated motion and a maximum acceleration of 2 cm/s. Considering a real scenario where the endoscopic device is slowly propelled forwards through the GI track, the values of linear acceleration are mainly affected by the gravitational acceleration that is modeled in the World Frame as $g = [0.0; 0.0; -9.81] m/s^2$.

Although the requirement of the 6 Dof is satisfied, the main drawback of this approach is the lack of measurements noise which is typical of a real Inertial Measurements Unit sensor.

It follows that:

1. Since the simulated values of linear accelerations and angular velocities are ideal, a localisation algorithm based on these data is not robust against sensor noise and could face instability problems if applied to the physical system;

2. The core of the new localisation algorithm is an Unscented Kalman Filter that receives in input measurements affected by uncertainty and retrieves in output filtered values. So, the presence of noisy measurements is essential to justify the choice of the new algorithm.

To solve the problems listed above, a new version of the simulator that relays on several hardware components and sensor fusion algorithm has been studied and developed. Unlike before, the attitude and angular velocity of the simulated capsule depends on an IMU sensor while the keyboard is still used to control its position and linear velocity. Figure 3.1 shows the benefits produced by the introduction of the IMU sensor (MPU9250).

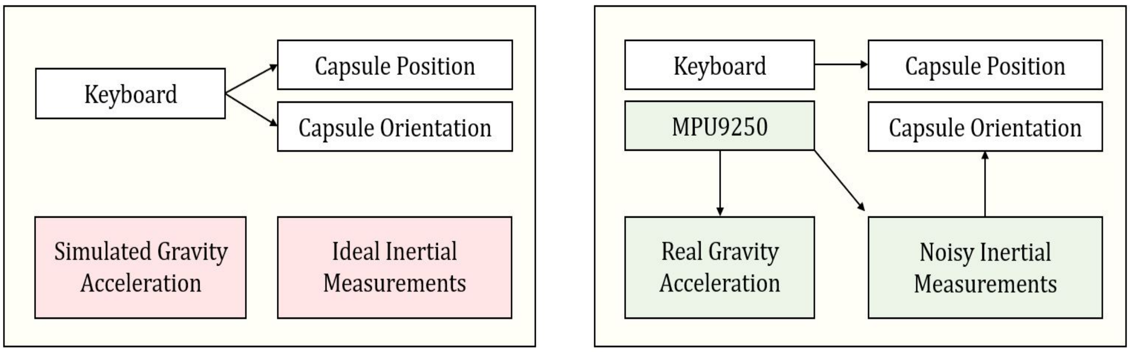


Figure 3.1: Simulator block diagram before (Left) and after (Right) the introduction of the IMU sensor.

However, the orientation of the capsule can not be directly obtained from the IMU data and a more complex approach that involves sensor fusion algorithm must be followed.

3.2 System Components

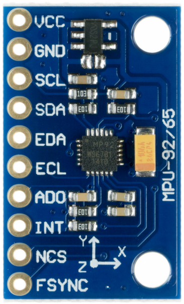
This section gives the reader a comprehensive description of the choosen hardware components for the implementation of the MFE Simulator. It is worth mentioning that different design approaches have been taken during the entire project. As a result, the developed material allows the user to implement the simulator by means of not unique development boards and sensors.

3.2.1 MPU9250 IMU Sensor

As previously mentioned in Chapter 2.2, a LSM330DLC Inertial Measurements Unit by STMicroelectronics is mounted inside the capsule right behind the Internal Permanent Magnet. The inertial measurements in the Capsule Frame are published on the topic /MAC/imu/raw as sensor_msgs/Imu messages and play an important role in the localisation algorithm.

Although the LSM330DLC is a 6 axes device, a 9 axes MPU9250 has been selected for this project. While the former provides as output the cartesian coordinates of acceleration and angular velocity only, the latter can also measures the strength of the magnetic field. As it will be clarified in Section 3.4, this additional set of values is used to make the MFE simulator closer to the real system where the Earth magnetic field can not be neglected. Note that the magnetometer is not mandatory for the proper operation of the MFE Simulator.

The main features of the choosen module are shown in Table 3.3.



Gyroscope Fullscale Range	± 250, ± 500, ± 1000, ± 2000 deg/sec
Accelerometer Fullscale Range	± 2g, ± 4g, ± 8g, ± 16g m/s^2
Magnetometer Fullscale Range	± 4800 μT
Communication Protocols	I^2C, SPI

Table 3.3: MPU9250 Product Specifications

The Product Specifications shows how this module perfectly fits with our application and provides sufficiently high safety margin for all the measurements. In a clinical scenario the capsule is mainly subjected to the Gravitation Acceleration ($\cong 9.81\ m/s^2$) and to low angular velocities ($\cong$ 30 deg/sec). Moreover, the maximum magnetic field that the capsule can face in the workspace is around 1.5 T according to the EPM and Coil specifications.

As for the communication protocol, the I^2C (Inter-Integrated Circuit) serial communication bus has been selected as the most appropriate one to exchange data with the Development Board. Before proceeding, it is useful to highlight the *Orientation of Axes of Sensitivity* and *Polarity of Rotation* for accelerometer, gyroscope and magnetometer to avoid misunderstandings during the analysis of the Python code of the Simulator. See Figure 3.2. More detailed information about the module are available in the *MPU-9250 Product Specification Revision 1.1* [12].

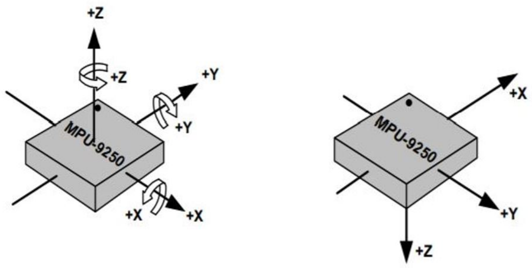


Figure 3.2: Accelerometer and gyroscope RF (Left). Magnetometer RF (Right).

Once the appropriate Inertial Measurements Unit is selected, the choice of the Development Board covers an essential role to achieve the following three functionalities:

1. First, the board must be able to wake-up the IMU and to start reading the raw data via either I^2C or SPI at the wanted frequency of 100Hz. The former communication protocol is preferred for this application due to its simpler implementation and because of the presence of a unique slave device.

2. Once the raw data have been read by the development board, a further step is needed to scale them in the wanted range and to set the correct measurement unit. An additional calibration can be performed to increase the reliability of the data.

3. Finally, the board is programmed to enable the serial communication with the development PC. In this specific application, the data are sent via USB cable.

Two different development boards have been exploited during the project: Arduino Uno by Arduino and STM32F411RE Nucleo by STMicroelectronics. Sections 3.2.2 and 3.2.3 explain in details how these board are programmed in their own Integrated Development Environment (IDE). Moreover, some schematics shows the needed wiring.

3.2.2 Arduino Development Board

In the first stage of the project an Arduino Uno board by Arduino has been selected to achieve the functionalities listed above. Before analysing the developed purpose-built code shown in Appendix A, the reader is invited to follow these preliminary steps to download and set-up the needed libraries in the Arduino IDE: Download the Bolderflight/MPU9250 file [9] and add it to the Arduino libraries. Then, open the file Basic_I2C.ino and replace the original code with the one shown in Appendix A.

Concerning the hardware environment, four wire are needed to connect the Arduino Uno Board to the MPU9250 as shown in Figure 3.3. The module is powered by 3.3V and the I^2C serial protocol is enabled by connecting the Serial Clock (SCL) to PIN A4 and the Serial Data (SDA) to PIN A5.

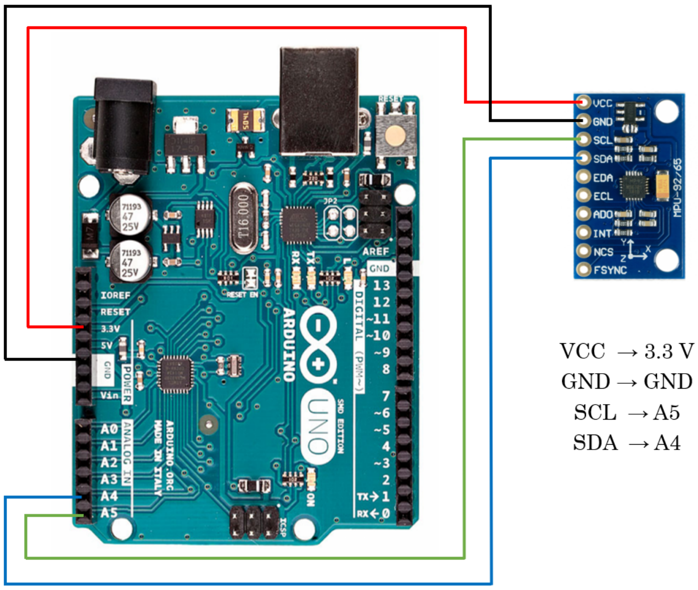


Figure 3.3: Connections between Arduino Uno and MPU9250.

During the analysis of the code of Appendix A, two main sections can be highlighted:

1. Lines 1-20: The needed libraries are included. The following important observation concerns lines 17-18.

   Pull-up resistors are often needed when working with I^2C to reach the Minimum Input Voltage V_{IH} to be considered a HIGH: while in the Arduino board there are not embedded pull-up resistors, some of them are available on the MPU9250 module. When the library Wire.h is used, two 10kΩ resistors are automatically enabled [1] to pull up the HIGH-level voltage on the SDA line close to 3.54V. By analyzing the Electrical Characteristics of the MPU9250 in Table 3.4, it can be seen how the V_{IH} is dangerously close to the maximum allowed value of 3.6V and can damage the module over time. Lines 17-18 are needed to disable the two pull-up resistors and bring the HIGH-level VDDIO to 3.25V. Although the new value is

[1] Line 76-77 of the file \Arduino\hardware\arduino\avr\libraries\Wire\src\utility\twi.c

lower that the initial one, it is still higher than the minimum threashold of V_{IH} = 1,75V and ensures a higher safety margin.

2. Lines 21-41: An endless loop begins once the initialization phase is completed. Acceleration, angular velocity and magnetic field data are printed to the serial port as human-readable ASCII text. A delay of 10 ms has been set to respect the working frequency of 100 Hz. Note that the printing order and the sign of the values depends on the different *Orientation of the Axes of Sensitivity* as previously highlighted in Figure 3.2. Although the magnetic field measurements are considered and printed, their role will be replace by a simpler algorithm in the final version of the MFE Simulator.

VDD	2.4V ÷3.6V
VDDIO	1.71V ÷3.6V
Min V_{IH}	1,75V

Table 3.4: I^2C Electrical Characteristics

Although the main functionalities of the code are achieved, it can be shown how the Serial Print operation can not reach 100Hz.

3.2.3 STM32F411RE Development Board

With reference to Chapter 2.2, the generation of the input signal for the coil and the processing of data in the real platform are achieved through a custom-built electronic system [23] that consists of the STM32 Nucleo Board and a driver circuit. In the second stage of the project the original Arduino Uno has been replaced by an STM32F411RE to make the simulator closer to the MFE and to exploit the wide range of available debug features of the STM IDE.

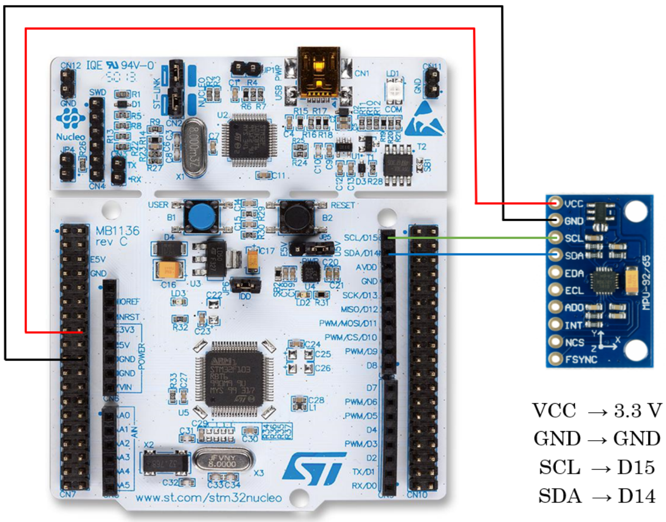


Figure 3.4: Connections between STM32F411RE and MPU9250.

Concerning the needed connections between the two devices, a set-up similar to the one previously shown with Arduino Uno is used. As before, the MPU9250 is powered by 3.3V and the I^2C serial protocol is enabled by connecting the Serial Clock (SCL) to PIN D15 and the Serial Data (SDA) to PIN D14. The Figure 3.4 above shows the needed connections.

Although the mounted circuit is similar to the one used with Arduino, a more complex purpose-built code has been developed through the STM32CubeIDE to exploit the higher performances of the microcontroller. It is worth mentioning that the Nucleo board can be also programmed by means of the Arduino IDE. In particular, the Library *STM32 Cores* allows to adapt and upload the Arduino code shown in Appendix A to many different boards including the STM32F411RE [1]. The main disadvantage of this approach is given by the impossibility to use the useful features and tools of the STM32CubeIDE.

Among the advantages of this IDE by STMicroelectronics, there is the possibility to program the needed registers of the STM32 Arm Cortex MCU by means of an integrated Software Tool. The final MCU configuration, that has been used in the project to achieved the wanted functionalies, is shown in Figure 3.5. A deeper explanation of how the MCU can be programmed in STM32CubeIDE is left to the reader since it falls outside the purpose of this description.

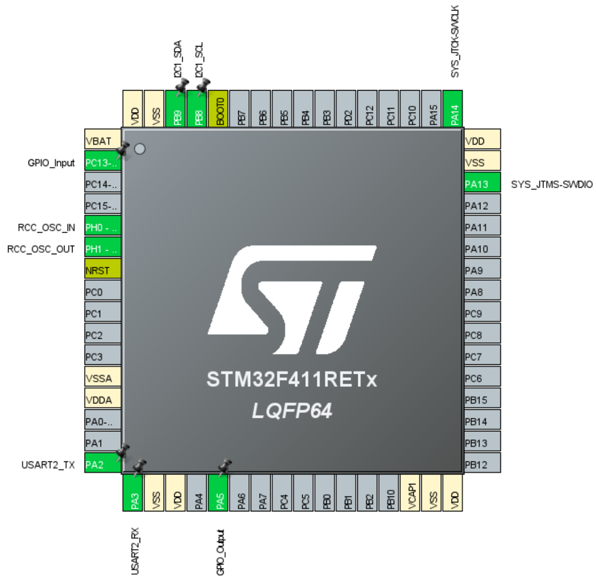


Figure 3.5: Final configuration of the STM32 Arm Cortex MCU.

The functions of the STM library *MPU6050 6 Axis Module* by T.Jaber [29] are used in this project to initialized and read the measurements data coming from the IMU. Note that the piece of code in Appendix B is only the section of the file main.c that has been designed to initialize, calibrate and work with the IMU; the auto-generated code is not included. Referring to Appendix B:

[1] Download the library. Then select these options: Tools → Board → Nucleo 64; Board part number → Nucleo F411RE; Upload method → STM32CubeProgrammer(SWD). Finally, upload the code to the STM development board

1. Lines 1-35: All the needed variables are defined. Function *_write* and *uprintf* are needed to print on the serial monitor the data;

2. Lines 36-92: The MPU9250 is woken-up and initialized after that the microcontroller and all the peripherals are configured. The parameters shown in Table 3.5 has been set by taking into account the worst scenario that the capsule could face. These values allow to scale the raw data coming from the IMU and set the right measurements units.

Accelerometer Full Scale Range	2g m/s^2
Gyro Full Scale Range	500 deg/sec
Clock Source	8 MHz

Table 3.5: Selected Full Scale Range for the scaled data.

As it will be explained later on in Section 3.4, angular velocity measurements are used to estimate the roll and pitch angle of the capsule with respect to the World reference frame. For this reason, the calibration of the gyroscope (Line 61-80) is needed to delete the unwanted offset values that cause a wrong estimate of the capsule's orientation. Once the MPU9250 is powered on, the calibration starts when the push button embedded on the STM board is pressed for the first time. While 600 samples are acquired and used to compute the offset value along each axis of the gyroscope, a warning message suggests the user not to move the device during all the calibration phase. Figure 3.6 shows the gyroscope signals before and after the calibration phase with the corresponding offset values.

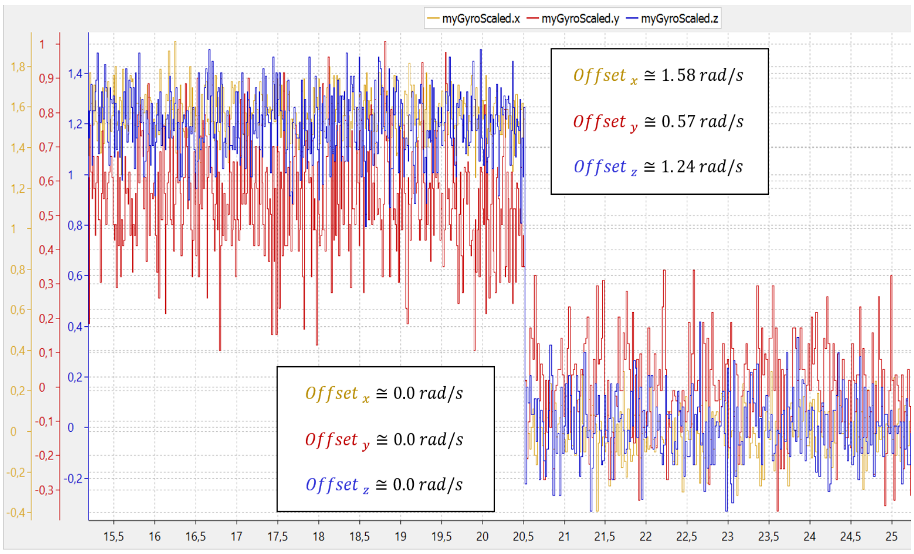


Figure 3.6: Calibration of the gyroscope signals.

Finally, the calibrated acceleration and angular velocity data are saved inside a buffer and printed to the serial port as human-readable ASCII text. The baud rate is set to 115200 bps. As for the magnetic field data, they are no more considered: in the final version of the MFE Simulator a simpler algorithm will be implemented to simulate the electromagnetic disturbance. Unlike with Arduino, no additional observation about pull-up resistors are needed.

3.3 Sensor Fusion: Madgwick Filter

Once the data are printed on the serial port either by Arduino or by the STM32 Board, they are read by means of the Python Serial Library and used inside the ROS environment for many purposes. The first one is the estimation of two out of the three angles that completely define the orientation of the capsule: the Roll and the Pitch. In general, an algorithm that use data coming from different sensors is defined with the prefix *Sensor Fusion*. Among them, the Madgwick Filter [17], the Mahony Filter [18] and the Complementary Filter [19] are widely used in robotic applications, as MAVs ,UAV and SLAM, to estimate the attitude of a body starting from measurements coming from an IMU. These algorithms are also defined with the acronym AHRS (Attitude and Heading Reference System).

Among the ones listed above, the Madgwick Filter has been selected for the MFE Simulator thanks to its slightly better accuracy in the attitude estimation and to the additional learning materials available in the Adafruit website (https://learn.adafruit.com/how-to-fuse-motion-sensor-data-into-ahrs-orientation-euler-quaternions). Although the mathematical explanation of the filter falls outside the purpose of this work, a quick description of the two versions of this algorithm is given below:

1. The first version of the filter takes in input the accelerations and angular velocities measured by the IMU along its reference frame. Basically, the algorithm integrates the gyroscope readings to compute a first estimation of the attitude that is sensitive to fast rotations of the body and is reliable in the short term. Then the linear accelerations are use to evaluate the direction of the gravity vector that is used to compensate and correct the long term gyro drift obtained by integration. See Figure 3.7. Madgwick filters use a proportional controller to correct the gyroscope bias and a quaternion representation of the attitude to avoid Gimbal Lock: a singularity that appears when two axes of the object have parallel orientation and causes the loss of one degree of freedom and the following measurement inaccuracy.

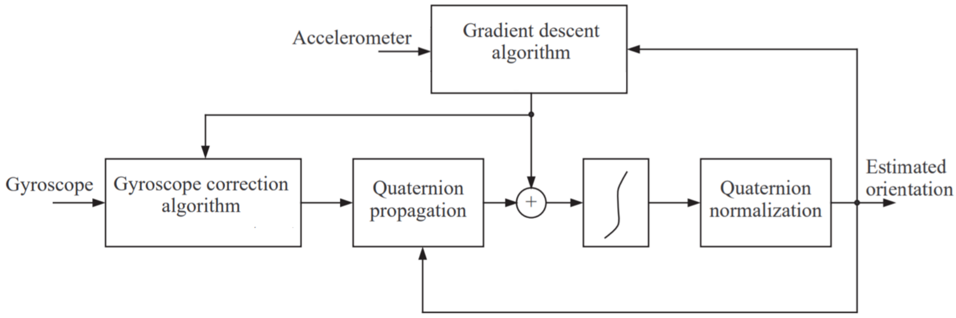


Figure 3.7: Madgwick Block Diagram.

2. The second version uses the measurements of the embedded magnetometer of the 9-axes IMU as additional input and retrieves in output the complete attitude of the body. As before, accelerations and angular velocities data are used to evaluate the roll and pitch angles while the direction of the Earth magnetic field is used to compute the yaw. It is worth mentioning that the direction of $\mathbf{B}_{Earth}$ must be specified in the algorithm since it depends on the specific place of the planet where the algorithm runs. Moreover, this technique can be chosen if and only if there are not other sources of magnetic field that affect the magnetometer: this is not the case of the MFE where the External Permanent Magnet (EPM) generates a $\mathbf{B}_{EPM}$ approximately 10^5 times higher than $\mathbf{B}_{Earth}$.

Although both versions have been implemented in the ROS simulator, functions *mahony_accel_gyro* and *mahony_accel_gyro_magn* in Appendix D respectively, only the first one will be used for the computation of the roll and the pitch.

Note that the unreliable yaw angle computed through the Madgwick Filter is firstly considered reliable to obtain a complete capsule pose $[p_x, p_y, p_z, \varphi, \theta, \psi]$. Then, these values are used as reference for the Kalman based localisation algorithm where the yaw angle is treated as an unknown state variable. Moreover, these data are published as ROS messages of type *sensormsgs/Imu* on the ROS topic */MAC/imu/raw*. This approach allows us both to simulate the full set of data sensed by the real capsule and to design a new localisation where the yaw is unknown.

3.4 Magnets Modelling

The External Permanent Magnet and the Coil play an essential role for the actuation and localization of the endoscopic capsule of the MFE. So, accurate models of the magnetic fields sensed by the six Hall-Effect sensors mounted on the capsule are crucial for a reliable simulator of the robotic platform. To accomplish this task and to allow the user to choose among different implementation methods, similar models of the needed magnetic fields have been developed by means of MatLab and Python. It is worth mentioning that the simulator is designed to perfectly fit with the existing code developed by the STORM Lab UK and STORM Lab USA; For this reason, the orientation of the six Hall-Effect sensors can be simulated by rearranging the printing order of the cartesian coordinates of the magnetic field.

A further preliminary observation about magnetometer noise is needed to design a simulator that mimics as much as possible the experimental conditions of the MFE. The magnetic localisation algorithm by Addisu Taddese [28] uses the calibrated [1] values of $\mathbf{B}_{EPM}$ and $\mathbf{B}_{coil}$ to compute the position of the capsule. Even though the Hall Effect sensors should retrieve constant values of magnetic field when the device is fixed, sensor noise affects the measurements and so the position. More realistic values are achieved by choosing a 9-axes MPU9250 instead of a 6-axes IMU since the measurement noise that affects the magnetometer data is added to the simulated values of the magnetic field. For the final version of the simulator, the measurement noise that affects the Hall Effect

[1] Two kinds of distortions can affect the Earth magnetic field measured by a magnetometer. Hard-iron disturbances arise from permanent magnets or magnetized steel while soft-iron disturbances arise from the interaction of the Earth magnetic field with magnetically soft material surrounding the sensor.

Sensors is simply modelled with a random process. This last option allows to mimic the desired behaviour and to simplify the code.

3.4.1 External Permanent Magnet Model

The model of the magnetic field based on the Generalized Complete Elliptic Integral C [5] has been selected as the most appropriate choice for the generation of the field density maps for the simulator. To perfectly mimic the reference frames used in the real application, the z-axis is chosen as axis of polarization of the EPM.

During the first stage of the project the MatLab library *Magnetic Field Modeling* by Federico Masiero [20] is used to generate two matrices that are used as look-up tables for B_ρ and B_z. The MatLab algorithm has been modified to generate two maps of the magnetic field only on the semi-positive plane XZ of the EPM (Figure 3.8). Thanks to the symmetry of the magnet, the entire workspace around the EPM can be mapped by means of a rotation around the z-axis followed by a reflection with respect to the XY plane. Finally, the Matlab matrices are saved as *.mat* files and used inside the Python code after some conversions.

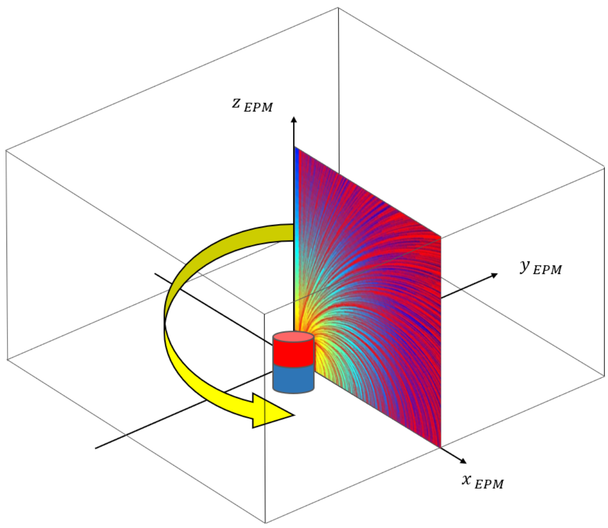


Figure 3.8: MatLab look-up table of the EPM magnetic field on the plane XZ. EPM axis of polarization: z-axis.

For the final version of the MFE Simulator, a similar look-up table is generated by means of the Python function *AxialCylindricalEllipticMap* developed by Addisu Taddese *et al.*. Although the computational approach is still based on the magnetic field model of Derby [5], the new function allows to simply the code and fits with the model that is used within the current localisation algorithm. Thanks to the knowledge of the full pose of the simulated capsule, the cylindrical components of the magnetic field are moved in the capsule reference frame and published on the ROS topics */MAC/epm/mfs1/raw* and */MAC/epm/mfs2/raw*.

It is important to highlight that the lookup table is computed offline both in Matlab and in Python: it means that the user must choose the volume around the EPM to be mapped in advanced. If the capsule is placed outside of the selected volume, the output values of magnetic field will be wrong.

For future purposes, it is essential to underline that the map establishes a unidirectional relationship: the magnetic field can be evaluated given the position but the vice-versa does not hold. The reason behind this limitation is the impossibility to use the function shown in Appendix C when the cartesian coordinates of the position vector are the unknowns of the problem. As it will be explained later on, this problem strongly limits the design of a new localisation algorithm based on an Unscented Kalman Filter.

3.4.2 Coil Model

The coil that is used in the MFE is composed by 2 layers (78 turns each one) of AWG-24 wire for a total length and width of 4cm and 18.2cm respectively. A current with amplitude of 800 mA allows to generate the needed magnetic field. To avoid discrepancies between the EPM and the Coil models, the Generalized Complete Elliptic Integral has been used. By calling n the number of turns per unit length and I the current amplitude, B_0 is defined as:

$$B_0 = \frac{\mu_0}{\pi} nI \tag{3.1}$$

The previous Python function is used to generate the lookup tables of the coil magnetic field. However, taking into account the different orientation of the Coil with respect to the EPM, an additional transformation is needed to allow a correct use of the maps. Firstly, a look-up table is created for the semi-positive plane XZ. Then, the whole workspace is covered by means of a rotation around the x-axis and a reflection with respect to the YZ plane.

Knowing the full pose of the simulated capsule, the cylindrical components of the coil magnetic field are moved in the capsule reference frame and published on the ROS topics */MAC/coil/mfs1/raw* and */MAC/coil/mfs2/raw*. Moreover, these data are continuously summed and, whenever 10 samples are reached, the average value is computed and published on the ROS topics */MAC/coil10/mfs1/raw* and */MAC/coil10/mfs2/raw*. In such a way, the set of ROS messages shown in Table 3.1 is completed and the Simulator is ready to be used for the development of the new localisation algorithm.

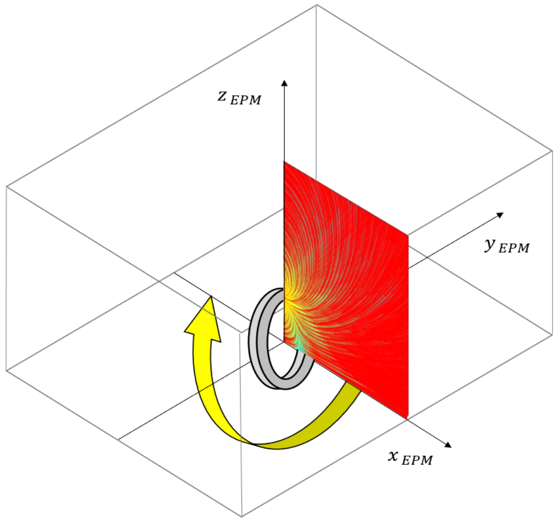


Figure 3.9: MatLab look-up table of the Coil magnetic field on the plane XZ. Coil axis of polarization: x-axis.

3.4.3 Goertzel Algorithm

The measurements of the simulated EPM and Coil magnetic fields are valid under the hypothesis of *decoupled magnetic fields*, i.e. they do not affect each others and the Hall-Effect sensors can measure $B_{\mathbf{EPM}}$ and $B_{\mathbf{Coil}}$ separately. Clearly, this condition can be only applied to the Simulator.

As previously mentioned in Section 2.3.2, in the Magnetic Flexible Endoscope the decoupling problem is solved by the Goertzel's Algorithm (Goertzel, 1958; Turner, 2003): a signal processing technique that runs on a separate thread at 18 kHz and relays on the 300 Hz oscillating magnetic field of the coil. Because of the impossibility to faithfully reproduce the same algorithm in the simulator, a simplified version has been designed to mimic as much as possible the real system. Even though it is mathematically correct, it is not included in the final simulator where the two magnetic fields are simulated separately.

3.5 Rviz: Robotic Manipulator Model and Markers

The ROS 3D visualization software Rviz has been used during the entire design of the simulator as additional tool to check graphically the correctness of the obtained results and to clarify conceptual mistakes. More specifically:

1. **Reference Frames RF**:
   When ROS is running, fourteen coordinate frames compose the structure of the simulator: *Map* and *World* are the parents of all the other RFs; *iiwa_link_0* to *iiwa_link_7* identify the current configuration of the manipulator; *capsule* and *imu_link* are coincident and centered on the capsule; finally *EPM* coincide with the center of the external permanent magnet. All of reference frames have been chosen and set according to the ones used in the real robotic platform. By means of the ROS command *rosrun rqt_tf_tree rqt_tf_tree*, the diagram shown in Figure 3.10 is obtained.

With reference to the real platform, the freedom of motion of the capsule with respect to the KUKA-iiwa makes the manipulator and the endoscopic tip two separate entities. It means that variations of the magnetic field are sensed whenever the capsule pose or the robot configuration changes. The same degree of freedom can be reached in the ROS Simulator if the capsule R.F. is not the child frame of the EPM R.F. or viceversa. However, the transformation matrix between these two frames can be directly retrieved when needed by means of the ROS command *rosrun tf tf_echo EPM capsule*.

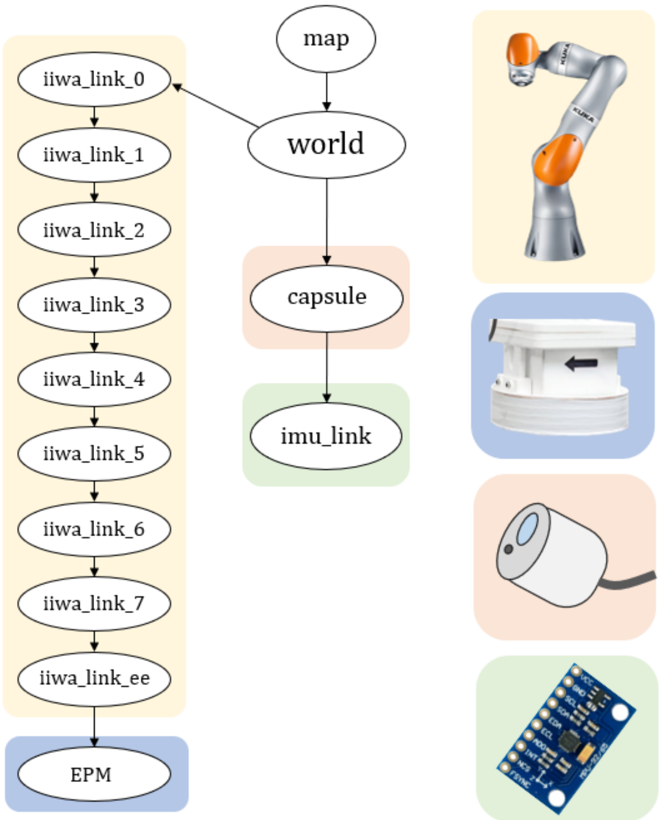


Figure 3.10: Tree structure of the Coordinate Frames of the MFE Simulator.

2. **Manipulator Model and Markers**:
   The ROS package *iiwa_description* by Ahoarau [1] is downloaded and the XACRO description of the manipulator KUKA iiwa 7 R800 is included in the 3D visualization tool RViz. The main advantage of this additional feature is the possibility to test the simulation when the capsule is fixed while the end-effector of the robotic arm moves and rotates. The configuration of the manipulator can be changed by means of the ROS node *joint_state_publisher* that allows the user to separately control all the joints of the kinematic chain through a simple graphical user interface (gui). By making the EPM coordinate frame coincident with the end-effector of the KUKA manipulator, the simulator reaches a good level of similarity with respect to the Magnetic Flexible Endoscope. Figure 3.12 below shows how the final version of the simulator looks in the 3D visualization tool Rviz: the coil and EPM are represented

by cylindrical markers on the end-effector of the manipulator while the capsule is shown on the bottom-left corner of the picture. Simple geometrical markers are used in RViz to help the user visualise the position and orientation of the EPM, coil and capsule during a simulation.

Figure 3.11: Magnetic Flexible Endoscope.

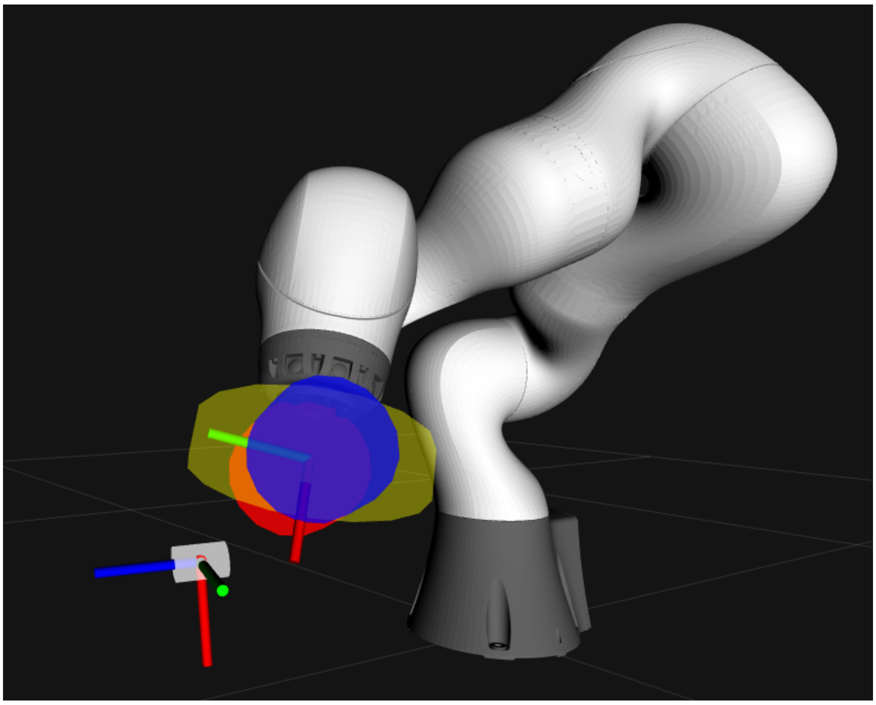


Figure 3.12: Final configuration of the MFE Simulator in RViz.

3.6 Block diagram of the MFE Simulator

The block diagram shown in Figure 3.13 summarises all the properties of the MFE simulator that have been previously analysed and highlights how the different parts are linked together. Recalling Table 3.1, all the requirements concerning the inputs and outputs of the simulator are satisfied.

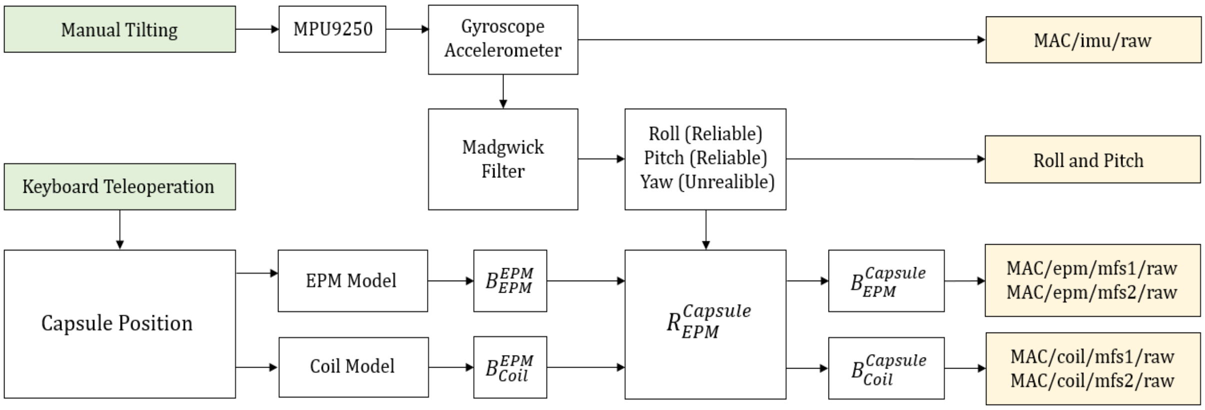


Figure 3.13: Complete block diagram of the MFE simulator.

Chapter 4

Pose estimation subsystem

In the previous chapter, the simulator of the Magnetic Flexible Endoscope is described in details with main focus on the needed output data. Thanks to the satisfactory results that are reached in terms of similarities with the real robotic platform, the development of a new robust localisation algorithm is no more unfeasible.

Recalling the block diagram of Section 3.6, the available input data that can be used for the new capsule localisation are:

1. The *inertial measurements* sensed by the simulated capsule that are published on the ROS Topic MAC/imu/raw. These values are referred to the Capsule reference frame;

2. The *Estimated Roll* and *Pitch* angles of the capsule with respect to the World reference frame;

3. The cartesian coordinates of the magnetic field generated by the External Permanent Magnet. These data are referred to the Capsule reference frame and are published of the topics MAC/epm/mfs1/raw and MAC/epm/mfs2/raw;

4. The cartesian coordinates of the magnetic field generated by the Coil. These data are referred to the Capsule reference frame and are published of the topics MAC/-coil/mfs1/raw and MAC/coil/mfs2/raw.

To estimate the full pose of a rigid body in the 3D space, three cartesian coordinates are needed for the position $[p_x, p_y, p_z]$ while its orientation can be expressed either by the Euler Angles $[\varphi,\ \theta,\ \psi]$ or by a quaternion $[q_0, q_1, q_2, q_3]$. Although these representations bring to equivalent results, the latter is more robust since it avoids the so called Gimbal Lock [1].

By taking into consideration a real clinical scenario where the MFE is surrounded by many medical devices, robustness against electromagnetic disturbance is among the most important features that the new localisation algorithm must have. A mathematical tool that concurrently estimates the capsule pose and filters out the sensors noise is thus

[1] Gimbal lock is the loss of one degree of freedom in a three-dimensional, three-gimbal mechanism that occurs when the axes of two of the three gimbals are driven into a parallel configuration, locking the system into rotation in a degenerate two-dimensional space.

needed to satisfy this constraint. Moreover, the highly not-linear nature of the magnetic field equations must be considered as an important requirement to be met.

As previously mentioned in Section 2.4, the main target of this work is the research and implementation of an alternative algorithm that can replace the existing Particle Filter. Among the wide range of estimators, Kalman Filters represent a robust solution for the design of the new localisation algorithm. An overview of the available Kalman filters and a more detailed explanation of their features is presented in Section 4.1.

4.1 Kalman Filters

Kalman Filters provide an efficient and recursive set of mathematical equations needed to estimate the state of a process while only having access to noisy and/or inaccurate measurements. Although different versions of these filters have been developed to cover a wider range of applications, only the **Kalman Filter** (1960, R.E. Kalman) [14] is an optimal estimator for linear systems. Although a more detailed mathematical formulation of the Kalman Filter is left to the reader, the following concepts are crucial to understand the work shown in this Chapter.

The following notation will be used: normal letters (a) denote scalars, bold ($\mathbf{a}$) denotes vectors and uppercase (A) denotes matrices. Subscripts (x_a) denote discrete time. Conditional subscripts ($x_{a|b}$) denote the variable x at time a, given measurements up to time b. Letters with an hat ($\hat{a}$) are estimated values that are affected by a certain degree of uncertainty. Letter k identifies the discrete time instant at which that variable is considered/acquired.

Both linear and non-linear Kalman Filters need a *state-space* representation of the process, i.e. a mathematical model of the physical system as a set of input $\mathbf{u}$, output $\mathbf{y}$, measurements $\mathbf{z}$ and state variables $\mathbf{x}$ related by a first-order differential equations. The *state* of the system can be represented as a vector $\mathbf{x}$ within that space. In particular, the state-space model is composed by:

- The *process model* which describes how the state $\mathbf{x}_{k+1}$ propagates in time based on the previous state $\mathbf{x}_k$, external inputs u_k and disturbances $\mathbf{w}_k$. It is usually represented by the function *f*.

$$\mathbf{x}_{k+1} = f(\mathbf{x_k}, \mathbf{u}_k, \mathbf{w}_k) \tag{4.1}$$

- The *measurement model* which describes how the available sensed data $\mathbf{z}_k$ are linked with the states $\mathbf{x}_k$, typically simulating measurements noise $\mathbf{v}_k$. The model is usually represented by the function h.

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \tag{4.2}$$

f and h are usually based upon a set of discretized differential equations that describes the dynamic of the system. The general structure of a state space model is shown in Figure 4.1. Depending on the nature of these two functions, the state space model can be either linear, if both f and h are linear in the state variables, or non-linear if the previous condition is not satisfied. Even though the Kalman Filter ensures an optimal solution only for linear systems, the Extended Kalman Filter and the Unscented Kalman Filter are similar estimators for non-linear models. Regardless the mathematical formulation of these filters, all of them aim to estimate the state.

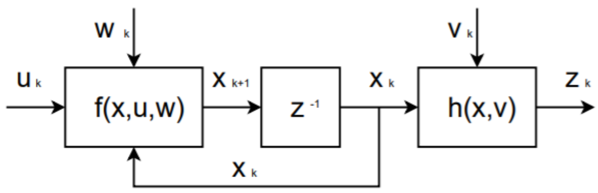


Figure 4.1: General state space model. z^{-1} : unit time delay.

State estimation concerns the computation of the probability density function (pdf), i.e. the mean value and the covariance, of a process which is not directly observable. To solve the problem, a two-steps algorithm is needed. This involves a prediction step where the future state is estimated based upon the current one followed by an updating step where the predicted state is corrected by means of the acquired noisy measurements. It is worth mentioning that, dealing with estimators as Kalman Filters, a Gaussian distribution with a specific mean $\hat{\mathbf{x}}$ and covariance P is assigned to each state. Figure 4.2 shows gives an idea of the two-steps algorithm.

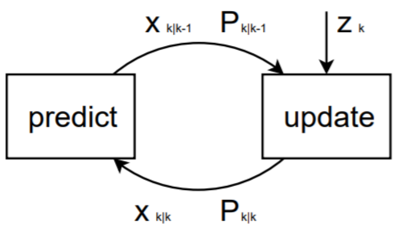


Figure 4.2: Kalman Filter two steps algorithm.

The **Extended Kalman Filter (EKF)** solves the problem of the non linearity through the computation of the Jacobian [1] of f and h around the mean of the Gaussian distribution of the estimated state. These two matrices are evaluated at each discrete time instant k and are identified with F_k and H_k.

$$F_k = \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{x}}|_{(\hat{\mathbf{x}}_{k|k}, \mathbf{u_k}, 0)} \tag{4.3}$$

$$H_k = \frac{\partial h(\mathbf{x}, \mathbf{v})}{\partial \mathbf{x}}|_{(\hat{\mathbf{x}}_{k|k-1}, 0)} \tag{4.4}$$

[1] The Jacobian matrix of a function in several variables is the matrix of all its first-order partial derivatives.

Thus, the EKF propagates the mean and the covariance through the linearized model. Figure 4.3 (left) gives a graphical representation of the propagation of the Gaussian distribution when $y = f(x)$: the dotted non linear function is linearized around the mean of $f(x)$.

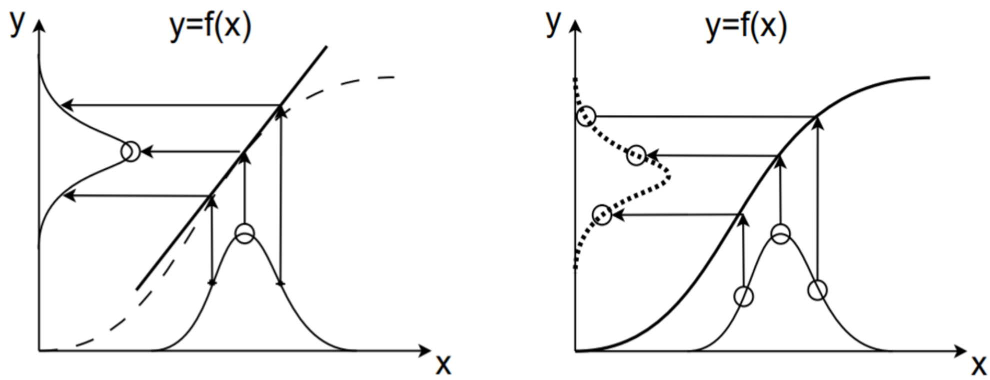


Figure 4.3: Extended Kalman Filter: Linearization of the non-linear function and propagation of the Gaussian Distribution (Left). Unscented Kalman Filter: Propagation of the sigma points with N=1 (Right).

The problem of the propagation of the Gaussian random variables through the non-linear functions $f(x)$ and $h(x)$ can be addressed with another technique: the **Unscented Kalman Filter (UKF)** [13]. Unlike the EKF which relays on the linearization of the functions $f(x)$ and $h(x)$, this new Filter approximates the Gaussian distribution of the state variables with a set of weighted *2N+1 sigma points* χ, where N is the number of the states. The points are selected such that their mean, covariance, and possibly also higher order moments, match with the Gaussian random variable. Then, the sigma points are propagated through the non linear functions $f(x)$ and $h(x)$; finally, the mean and the standard deviation of the weighted projected points is computed to approximate the new Gaussian density function at the output as shown in Figure 4.3 (right), yielding a more accurate result compared to ordinary function linearization. The reader is invited to compare the two pictures of Figure 4.4 to catch graphically the more accurate propagation of the state ensured by the UKF when the model is highly non-linear [31].

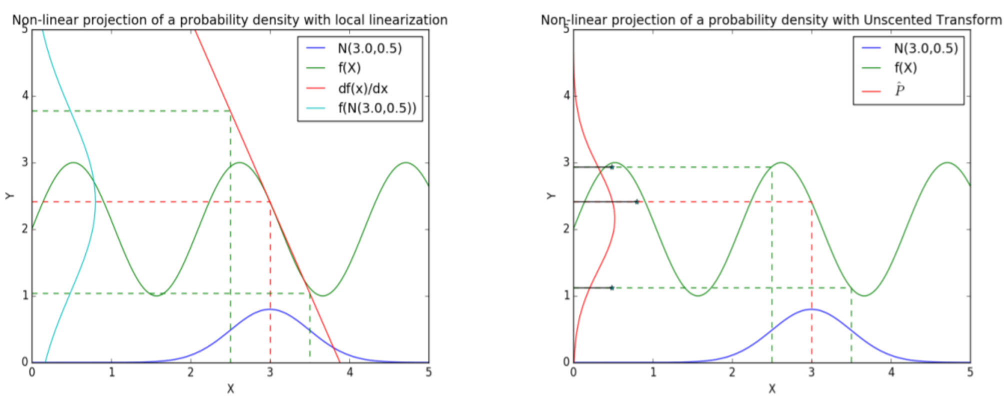


Figure 4.4: Propagated Gaussian distribution with EKF (Left) and UKF (Right)

4.2 The Unscented Kalman Filter

The Unscented Kalman Filter is selected as the most valuable alternative to the Particle Filter thanks to its ability to provide accurate state estimation despite of the non-linear nature of the state-space model. To apply this algorithm to the Magnetic Flexible Endoscope, the following variables and models must be designed and/or computed a priory. It is worth mentioning that the design of the new localisation relays on many concepts and models of the localisation algorithm designed by Addisu Taddese *et al.* [28]. The reader is invited to review Figure 2.6 to avoid misunderstanding with the nomenclature of vectors.

- **State variables**
  The pose of the endoscope tip is univocally identified by a $3x1$ position vector and three Euler Angles. Since the Roll and the Pitch are the only known parameters (from the Madgwick Filter), the $4x1$ state vector is:

$$\mathbf{x} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ \psi \end{bmatrix} \in R^{4x1} \tag{4.5}$$

  Note that inside the algorithm the elements of $\mathbf{x}$ will be treated as the mean values of the corresponding parameters. N = 4 identifies the dimension of the state-space model.

- **Covariance matrices**
  Due to the probabilistic nature of the Kalman Filters, each variable is seen as a Gaussian process identified by a mean value and a covariance (or standard deviation). The following matrices have been empirically chosen by Addisu Taddese et al. as a trade-off between convergence speed and jitter of the pose estimate. Note that the higher is the covariance, the bigger is the uncertainty that affects the corresponding state.

  Initial Process Covariance Matrix $P_0 = diag([0.001, 0.001, 0.001, 0.001]) \in R^{4x4}$

  This matrix is required for the design of Kalman filters and gives an information about the initial uncertainty between the real states and the estimated ones. The latter are usually initialized to zero. This matrix is updated at each iteration by the Kalman algorithm.

  Process Disturbance Covariance Matrix $Q = diag([0.0015, 0.0015, 0.0015, 0.01]) \in R^{4x4}$

  It provide information about the uncertainty that affects the propagation of the states. The elements of this matrix do not change and suggest the range of values within the new state could lay. The choice of these values is strictly connected with the real application: the fact that the last element is ten times higher with respect to the others suggests that the yaw angle could change faster that the position of the capsule. This concept will be clarified in the next point by explaining the Random Walk Process Model.

Measurements Covariance Matrix $R = diag([0.001, 0.001, 0.001, 0.001]) \in R^{4x4}$

The diagonal elements of this matrix bring information about the uncertainties that affect the acquired measurements. So, these values depend on the sensors noise and the mathematical computations that are performed to achieve the measurements. Note that these numbers are referred to the "measured" state and not to the simulated magnetic field measurements.

- **Process model**
  Referring to equation 4.1, it can be highlighted how a good estimation of the new state needs the knowledge of the applied input. In the case of the MFE, it coincides with the set of forces that allow actuation of the endoscopic tip, i.e. the magnetic coupling between the EPM and IPM and the frictional force between the capsule and the wall of the GI tract. Nevertheless, a precise mathematical formulation of these forces at each time instant is complex and extremely sensitive to the shape of the GI tract. Many unpredictable environmental factors as tissue folds of the GI tract or peristalsis could make the motion of the endoscopic tip significantly different with respect to the one which is expected from the robot actuation. See Figure 4.5 for clarification.

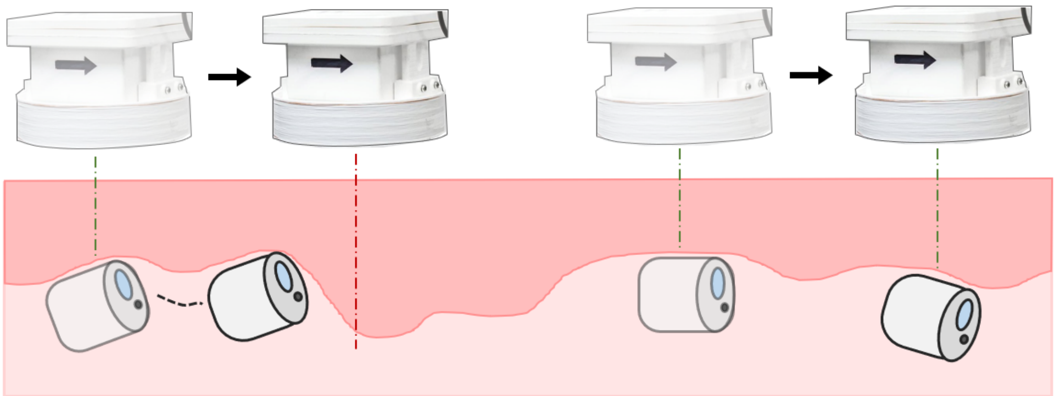


Figure 4.5: The same EPM displacement can bring to different actuation of the capsule due to environmental factors as tissue folds.

To avoid an unnecessary increase of complexity, the Random Walk Process Model 4.6 4.7 has been selected for the Kalman based localisation. Moreover, this choice is justified by the good results obtained by Addisu Taddese.

$$f(\mathbf{x_k}, \mathbf{w}_k) = \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{4.6}$$

$$\mathbf{w}_{k-1} \sim \sigma(0, Q) \tag{4.7}$$

where w is a set of four Gaussian processes σ with zero mean and Covariance Q.

- **Measurement model**
As already done in the current localisation algorithm, the position of the endoscopic tip and the yaw angle are evaluated starting from the EPM ($\mathbf{B}_{epm}^{capsule}$) and Coil ($\mathbf{B}_{coil}^{capsule}$) magnetic fields sensed by the Hall Effect sensors. By means of the Generalized Complete Elliptic Integral [5] and the knowledge of the rotation matrices between the reference frames, six non-linear equations can be used to obtain the "measured" state vector. The procedure that has been followed to achieve the needed equations is shown below; note that the definitions of B_ρ (5.23) and B_z (5.23) are the ones of Section 3.4.1 and that the unknowns p_x, p_y, p_z and ψ are highlighted in blue.

  1. The measurements of the EPM and coil magnetic fields are moved from the *Capsule R.F.* to the *EPM R.F.*:

$$\mathbf{B}_{epm}^{epm} = R_{world}^{epm} * Rot(\phi, \theta, \psi) * \mathbf{B}_{epm}^{capsule} \tag{4.8}$$

$$\mathbf{B}_{coil}^{epm} = R_{world}^{epm} * Rot(\phi, \theta, \psi) * \mathbf{B}_{coil}^{capsule} \tag{4.9}$$

  2. Compute the expressions of the cartesian coordinates of the EPM and Coil magnetic fields by means of the Generalized Complete Elliptic Integral. The model needs the position vector of the point with respect to the magnet center $\mathbf{d} = [d_x, d_y, d_z]^T$. However, the measurements model must be a function of the state variables $\mathbf{p}$ and ψ. Given the position of the EPM with respect the Global frame $\mathbf{l} = [l_x, l_y, l_z]^T$ (known thanks to the real-time encoders) the equations can be written as a function of the state variables:

$$\mathbf{d} = [\ p_x - l_x,\quad p_y - l_y,\quad p_z - l_z\ ]^T \tag{4.10}$$

  For the EPM:

$$B_{x_{epm}}^{epm} = B_\rho^{epm}(d_x, d_y, d_z) * cos(atan2(d_y, d_x)) \tag{4.11}$$

$$B_{y_{epm}}^{epm} = B_\rho^{epm}(d_x, d_y, d_z) * sin(atan2(d_y, d_x)) \tag{4.12}$$

$$B_{z_{epm}}^{epm} = B_z^{epm}(d_x, d_y, d_z) \tag{4.13}$$

  For the Coil

$$B_{x_{coil}}^{epm} = B_\rho^{coil}(d_x, d_y, d_z) * cos(atan2(d_y, -d_z)) \tag{4.14}$$

$$B_{y_{coil}}^{epm} = B_\rho^{coil}(d_x, d_y, d_z) * sin(atan2(d_y, -d_z)) \tag{4.15}$$

$$B_{z_{coil}}^{epm} = B_z^{coil}(d_x, d_y, d_z) \tag{4.16}$$

3. Thus, six non-linear equations in an implicit form can be obtained by forcing to zero the difference between the just computed terms. The measurements model $h(\mathbf{x}) = 0$ is in the form of the Gauss Helmert (GHM) where it is not possible to separate observations from parameters. However, only four out of six equations are necessary to solve the system, so to define the measurements model for the UKF.

Now that the needed parameters and models for the Unscented Kalman Filter have been defined, an high level description of the algorithm is provided below. However, a deeper analysis of the mathematical formulation of the filter is left to the reader since it is out of the scope of the present work. By using N to identify the system order, the algorithm can be divided into two steps: prediction correction.

Prediction step before measurements acquisition:

- Generation of the sigma points: χ_{k-1}.

  The Gaussian distribution identified by the initial mean vector x_{k-1} and covariance P_{k-1} matrix P is approximated by means of $2N + 1$ sigma points. Even though there exist multiple solutions for the choice of the sigma points, the first one (R^{4x1}) usually coincides with the mean vector, while the remaining 8 are chosen symmetrical around the mean. In the definition of χ, a parameter allows the user to select the spatial distribution of the points while the square root of the covariance matrix P is performed by the Cholesky matrix square root L.

- Projection ahead of the sigma points through the process model f: $\chi_k^* = f(\chi_{k-1})$.

  Given the set of the sigma points χ and the process model f defined in an explicit form with respect to the state variables (See 4.6), the projection of χ through f is performed.

- Computation of the weights for sigma points and covariance.

  Once the $2N + 1$ sigma points have been computed and projected ahead through f, specific weights are assigned to all of them according to their distance with respect to the mean: the weight for the 1^{st} sigma point is the highest one. As before, some parameters allow the user to customize this set of values.

- Computation of the predicted mean and covariance.

  The sets of sigma points and weights are then used to reconstruct a new Gaussian distribution. From it, the predicted mean $\hat{x}_k$ and covariance $\hat{P}_k$ are retrieved. Note that the Process Disturbance covariance matrix Q_k is used for the computation of $\hat{P}_k$.

Correction step after measurements acquisition:

- Evaluation of the new sigma points $\hat{\chi}_k$ by means of the predicted mean $\hat{x}_k$ and covariance $\hat{P}_k$.

  A new set of sigma points $\hat{\chi}_k$ is computed based on the predicted mean and covariance.

- Projection of the sigma points through the measurements model h: $\hat{\chi}_{zk} = f(\hat{\chi}_k)$.

  The set of the sigma points $\hat{\chi}_k$ are projected through the measurement model h defined in an explicit form with respect to the state variables (See 4.2).

- Computation of the new predicted mean $\hat{\mathbf{z}}_{\mathbf{k}}$, innovation covariance S_k and Kalman filter gain K_k.

  The Gaussian distribution that is reconstructed from the set of sigma points $\hat{\chi}_{zk}$ is used to compute the new predicted mean, the innovation covariance S_k and the Kalman filter gain K_k.

- Correction of the predicted mean by means of the measurements vector m_k.

  Finally, the measurements vector m_k and the previously obtained terms are used for the computation of the corrected mean $\hat{\mathbf{x}}_{k_{corrected}}$ and covariance $\hat{P}_{k_{corrected}}$. These will be used as input for the next iteration of the algorithm.

4.3 Problem: Explicit form of the measurement model

The measurement model is composed by four out of the six equations previously shown that link the measurements of the Hall Effect sensors with the Generalized Complete Elliptic formulation of the EPM and coil magnetic field. The reduced number of needed equations is based on the order of the system (i.e. N = 4).

By analysing the general formulation of the measurement model of the Unscented Kalman Filter, it can be highlighted how an explicit form of h 4.2 in the state variables is mandatory to make the algorithm work correctly. So, the chosen four equations should be written in an explicit form in p_x, p_y, p_z and ψ. If it is not possible to reach this result, a Kalman based algorithm can not be used for the new magnetic localisation. The following steps have been followed to find a solution for this mathematical constraint.

1. **Generalized Complete Elliptic Integral**

   As previously mentioned, the Generalized Complete Elliptic Integral is the most accurate and precise model for the EPM and coil magnetic fields and whichever variation from this model would bring to unwanted discrepancies with respect to the real fields. So, an explicit form of the model has been investigated by using the symbolic toolbox of Python and Matlab. Although no satisfactory results are obtained from this attempt, an important observation about the Generalized Complete Elliptic Integral is highlighted: the efficient algorithm developed by Derby *et al.* [5] provides a solution of the Elliptic Integral in a closed-form only when the input is the position vector so that the magnetic field must be evaluated in a precise point of the space. When trying to solve the problem in the opposite direction (i.e. the inputs are the cartesian coordinates of the magnetic field) no solution is retrieved. In particular, the mathematical limitation which avoid the inverse problem to be solved is the inequality at Line 34 C: when using the symbolic toolbox, the comparison between real numbers (*g*errtol*) and symbolic variables (*abs(g-k)*) do not have a mathematical meaning. Similar results are achieved without making use of the algorithm of Derby *et al.* for the implementation of the Elliptic Integral. As a result, an explicit form of the system can not be reached by means of the Generalized Complete Elliptic Integral.

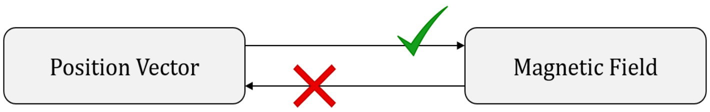


Figure 4.6: Unidirectional validity of the algorithm by Derby *et al.*

2. **Dipole Model**

   The magnetic fields generated by a cylindrical permanents magnet or by an electromagnetic coil can be approximated by the $\mathbf{B}$ of the equivalent magnetic dipoles. Although these models brings to bigger discrepancies with respect to the Generalized Complete Elliptic Integral, a simplified system of equations is obtained. By calling $\mathbf{m_E}$ and $\mathbf{m_C}$ the dipole moments of the EPM and Coil, $\mathbf{\hat{m}_E} = \mathbf{\hat{z}} = [0,0,1]^T$ and $\mathbf{\hat{m}_C} = \mathbf{\hat{x}} = [1,0,0]^T$ the unit vectors along their direction, $\mathbf{d}$ the position vector and $\mathbf{\hat{d}}$ the unit vector along its direction, the magnetic fields generated by the EPM and Coil are given by:

$$\mathbf{B_E}(\mathbf{d}) = \frac{\mu_0 \|\mathbf{m_E}\|}{4\pi \|\mathbf{d}\|^3}(3\mathbf{\hat{d}\hat{d}^T\hat{m}_E} - \mathbf{\hat{m}_E}) \tag{4.17}$$

$$\mathbf{B_C}(\mathbf{d}) = \frac{\mu_0 \|\mathbf{m_C}\|}{4\pi \|\mathbf{d}\|^3}(3\mathbf{\hat{d}\hat{d}^T\hat{m}_C} - \mathbf{\hat{m}_C}) \tag{4.18}$$

   After having moved the measurements of the EPM and coil magnetic fields from the *Capsule R.F.* to the *EPM R.F.* , six non-linear equations can be obtained. By expressing $\mathbf{d}$ as a function of $\mathbf{p}$ and $\mathbf{l}$ (Figure 2.6), the system is in in implicit form in the state variables and four explicit expressions of p_x, p_y, p_z and ψ are then needed. Again, Matlab and Python symbolic toolbox are used to solve the symbolic system of equations but none of the two software is able to find an explicit solution.

3. **UKF or EKF able to deal with implicit measurement model**

   Although the Unscented Kalman filter and the Extended Kalman filter are widely used with non-linear systems, the vast majority of the documented applications relays on explicit measurement model. As for implicit models in Gauss Helmert form *h(*$\mathbf{x}$*) = 0*, only few papers have been spotted during a research in literature. Vogel *et al.* [32] developed an Iterated Extended Kalman Filter with Implicit Measurement Equations of type GHM. The algorithm is based on the Jacobian matrix of the implicit measurement model h with respect to the state variables $\mathbf{x}$ and relays on an iterated optimization problem. The INTerval LABoratory (INTLAB) Toolbox for Matlab has been used to perform numerical differentiation.

   Although the algorithm can manage implicit measurement model, it has been excluded for the new localisation of the MFE due to its mathematical complexity and to the difficulties of its Python implementation.

This overview highlights the impossibility to obtain an explicit formulation (with respect to the state variables $\mathbf{x}$) of the EPM and Coil magnetic field and the difficulties to implement a UKF able to deal with implicit measurement model.

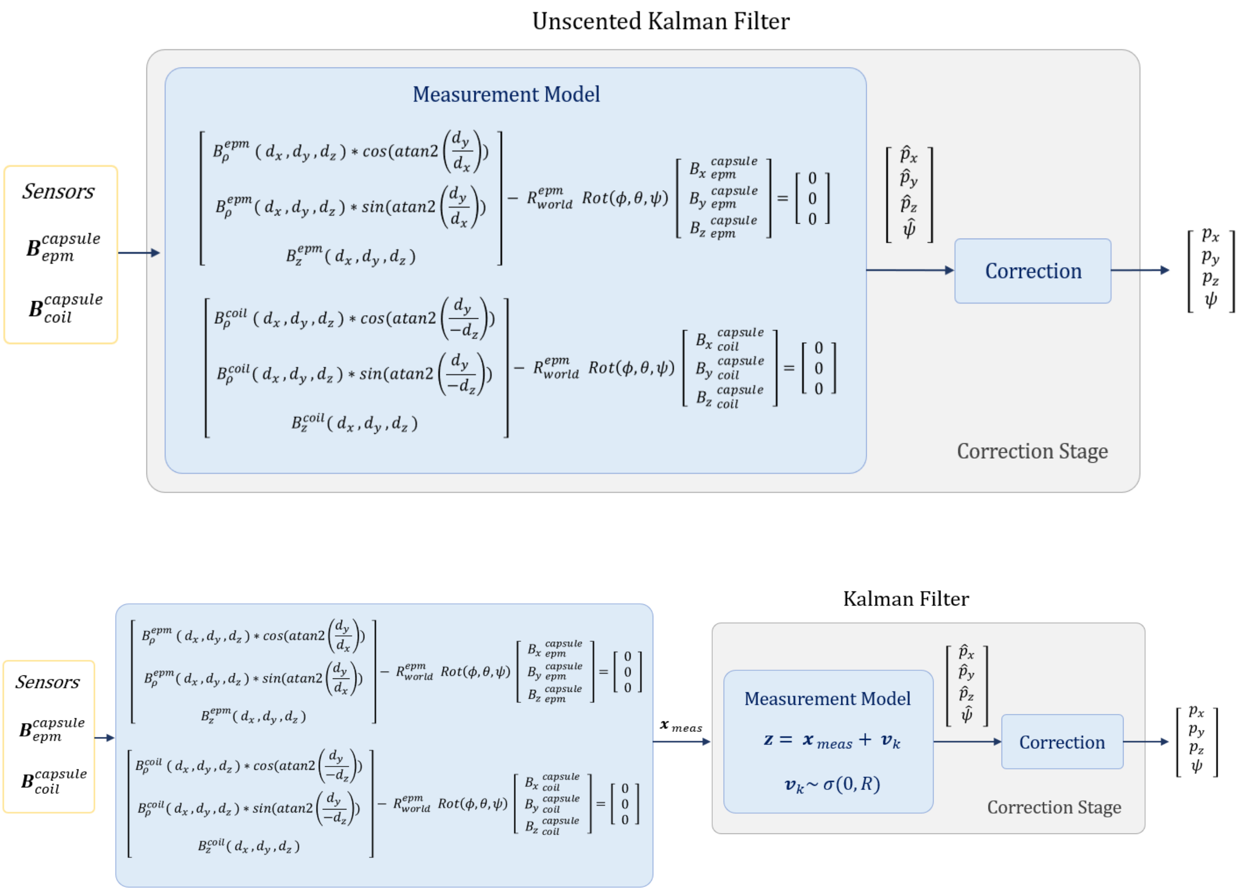
As a result, several modifications of the entire estimation algorithm have been introduced. The set of four implicit equations are solved outside the filter by means of the Python function *root()* that is based on an iterative optimization problem: given the initial conditions as input parameter, the algorithm computes the state vector $\mathbf{x}$ that guarantees the closest local minimum. This first modification allows the user to use the Generalized Complete Elliptic Integral that guarantees a more accurate model of the magnetic field with respect to the one based on the dipole moment. Moreover, if the measurement vector is computed aside and used as input of the estimation, the problem is no more non-linear from the Kalman Filter point of view.

The new *measurement model* is:

$$\mathbf{z_k} = h(\mathbf{x_k}, \mathbf{w}_k) = \mathbf{x}_{Measured} + \mathbf{v}_k \tag{4.19}$$

$$\mathbf{v}_k \sim \sigma(0, R) \tag{4.20}$$

Thanks to this new approach and to the linear measurement model, the initial Unscented Kalman Filter can be replaced by a simpler Kalman Filter. It is important to underline how this variation implies modification of the measurement model only, while the state variables, covariance matrices and process model remain the ones previously described in Section 4.2.



Note that $\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} - \begin{bmatrix} l_x \\ l_y \\ l_z \end{bmatrix}$, where l_x , l_y and l_z are known thanks to the real-time encoders.

Figure 4.7: Comparison between the first implementation with the Unscented Kalman Filter (UP) and the second one with the Kalman Filter (DOWN).

Chapter 5

Kalman Filter based Localisation Algorithm and Results

A detailed description of the developed algorithm based on the Kalman Filter is presented in this chapter. To enable the communication with the MFE Simulator an additional ROS node has been developed for the new magnetic localisation.As before, the programming language Python is used for the implementation of the algorithm. It is worth mentioning that the target of this work is to provide a proof of concept of a Kalman Filter based localisation; if it was needed to make the code faster for a real-time application, the code would be re-written in C++.

Once the simulator is running, the following steps are performed:

1. Firstly, the Kalman Filter is initialized: the initial estimate is set by convention equal to the zero vector $\mathbf{x}_0 = \mathbf{0} \in R^{4x1}$ while the covariance matrices are the ones shown in Section 4.2.

2. The input data of the new magnetic localisation are the *MagneticField* messages published on the ROS topics of Table 3.1. Since both the EPM and the Coil measurements are essential for the algorithm, a messages synchronization is performed in advance by means of the ROS function *ApproximateTimeSynchronizer*. Note that the running frequency of the new ROS node is imposed by the publication rate of these messages.

3. Once all the data are available, the prediction step of the Kalman Filter is performed: starting from $\hat{\mathbf{x}}_{k-1}$ and $\hat{P}_{k-1}$, the future state and process covariance is estimated depending on the process model.

4. Before proceeding with the correction step of the Kalman Filter, the measured (indirectly) state vector must be retrieved from the measurements of the EPM and coil magnetic fields. For this purpose, the Python function *scipy.optimize.root()* (or *scipy.optimize.fsolve()*) has been used to solve the system of non-linear equations shown in Figure 4.7: as the vast majority of the solvers, this function needs an initial condition $\mathbf{x_0}_{solver}$. According to it, different operations are performed depending on the cycle of repetitions of the algorithm:

    - Cycle = 1: The first localisation of the capsule is the hardest and most crucial step of the whole algorithm. Since the pose of the endoscopic tip is unknown at the beginning, the entire workspace around the EPM must be sampled and tested to find the $\mathbf{x_0}_{solver}$.

- Cycle > 1: For all the following iterations, $\mathbf{x_0}_{solver}$ is set equal to the output of the Kalman Filter, i.e. the estimated state vector. This condition allows the solver to be faster since the initial condition is close to the solution or coincident with it. Note how this step introduces an additional level of robustness to the algorithm since it assumes that the capsule displacement is limited to few millimeters each iteration.

5. Once the measurement vector has been computed, the correction step of the Kalman Filter is performed. Finally, the corrected state $\hat{\mathbf{x}}_{k_{corrected}}$ and process covariance $\hat{P}_{k_{corrected}}$ are used as estimate of the capsule pose. Moreover, they will be used as *old parameters* for the next iteration.

The flow chart 5.1 below is intended to graphically help the reader during the explanation of the new magnetic localization.

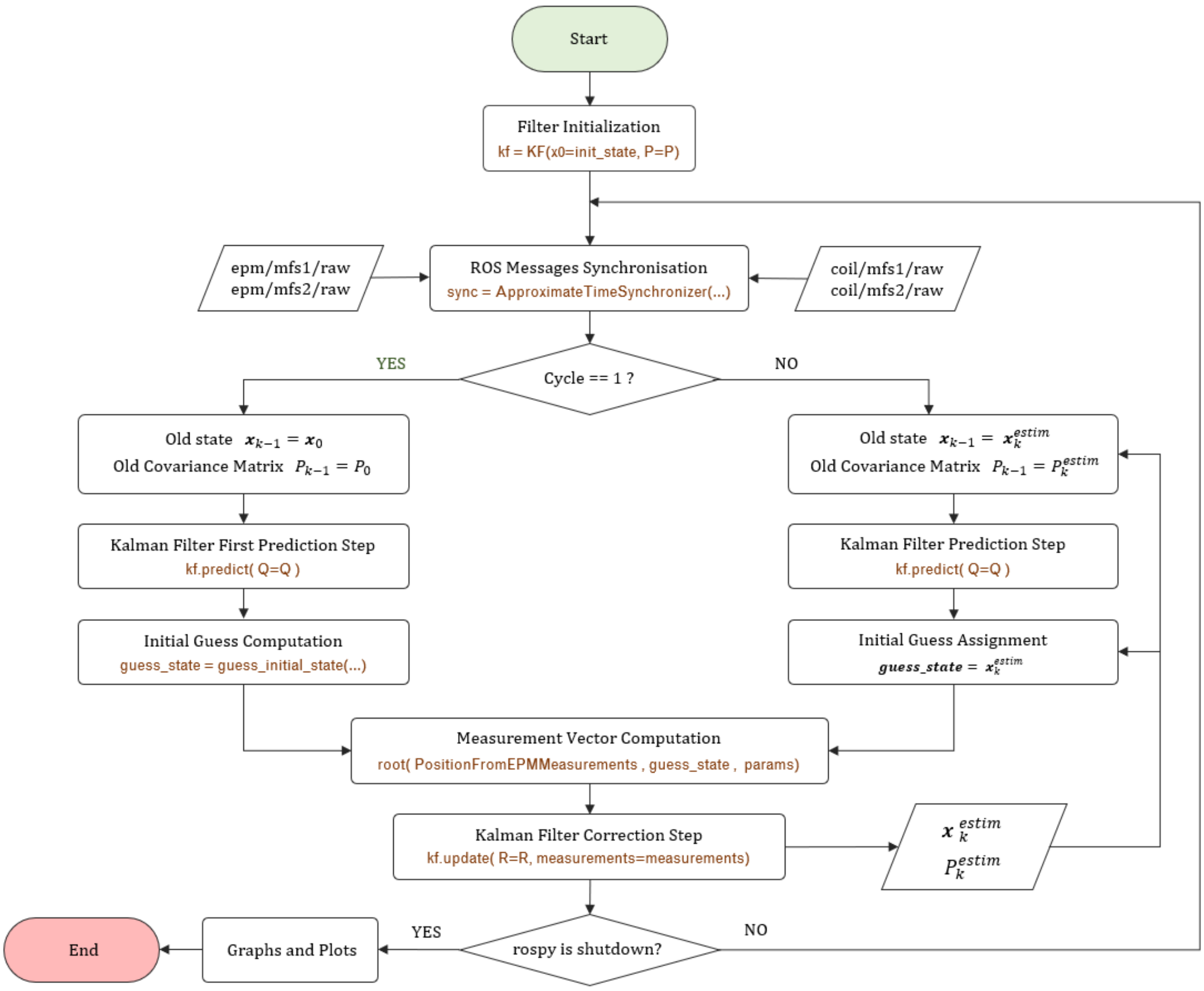


Figure 5.1: Flow chart of the new magnetic localisation algorithm.

5.1 Results

A set of tests have been carried out to evaluate the correctness and the performances of the developed localisation algorithm. Moreover, the set of experiments has been carefully chosen to allow a comparison with the algorithms described in Section 2.3 (Di Natali *et al.* 2013, Di Natali *et al.* 2016, Taddese *et al.* 2018). In particular, the algorithm has been tested in two different simulated scenarios:

- *Static-static condition*: the EPM is maintained fixed in a known position while the simulated endoscopic capsule is spawned around it. This experiment tests the ability of the algorithm to detect the capsule without the need of an initialization of the its pose.

- *Static-dynamic condition*: as before, the simulated EPM is maintained in a fixed position while the endoscopic tip is forced to follow a trajectory that involves variations of all the four state variables (i.e. p_x, p_y, p_z, ψ). This test in performed to evaluate the capabilities of the algorithm to track the pose of the moving capsule.

The results have been evaluated in terms of average error between the pose given by the simulator (selected as the real one) and the estimate given by the localisation algorithm. Moreover, the 3D simulation environment RViz has been employed to have an immediate, but less accurate, graphical feedback of the results.

Concerning the first set of tests (i.e. *Static-Static* condition), the capsule is spawned in a volume below the EPM that is chosen coherently with the one employed in a clinical scenario. As previously mentioned, the main limitation is associated to the choice of the *initial guess* during the first iteration of the algorithm that computes the estimated state $\hat{\mathbf{x}} = [\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}]$ as the closest local minimum of the measurements model (i.e. the system made of four out of the six non-linear equations 4.11 - 4.16). This task is accomplished in an iterative way by the Python function *scipy.optimize.fsolve()* which requires an *initial guess* to perform the computations. The problem of localising the capsule during the first iteration of the algorithm has been solved by selecting 27 points, equally spread within the workspace. For each of these points, six iterations of the localisation algorithm are run and the initial guess is chosen as the point that allows for the convergence to the closest solution in term of distance. For the next iterations, the initial guess is chosen equal to the estimated state allowing for a pose detection that converges to the real capsule pose. The following test procedure has been performed: as shown in Figure 5.2 12 tests subdivided into three categories depending on the distance between the center of the EPM and the tested point (i.e. *12cm*, *24cm* and *35cm*). Each of this test is done by running the localisation algorithm for 60 seconds. At each iteration, the errors between the estimate states and the real ones are computed and saved in an array that is used at the end of the simulations for the computation of the average error (Δx, Δy, Δz, $\Delta \psi$) and the standard variation (σ_x, σ_y, σ_z, σ_ψ). See Table 5.1. As for the attitude of the capsule, it is changed randomly after each test. For each distance (i.e. *12cm*, *24cm* and *35cm*) the average error and the standard deviation are computed by averaging the obtained results:

$$\Delta state_d = \frac{\sum_{i=1}^{4} \Delta state^d}{4} \tag{5.1}$$

where d = *12cm, 24cm and* 35cm while the *state* = x, y, z *and* ψ.

d [mm]	$e_{d_x} \pm \sigma_{d_x} [mm]$	$e_{d_y} \pm \sigma_{d_y} [mm]$	$e_{d_z} \pm \sigma_{d_z} [mm]$	$e_{d_\Delta} \pm \sigma_{d_\Delta} [mm]$
120	5.2 ± 2.06	16.53 ± 2.31	4.43 ± 2.54	8.69 ± 0.94
240	5.4 ± 2.87	5.99 ± 8.11	2.24 ± 2.90	8.14 ± 2.44
350	4.21 ± 5.59	22.67 ± 18.69	3.11 ± 2.16	5.34 ± 4.34

Table 5.1: Static-Static test: average results evaluated for three distances from the center of the EPM (120 mm, 240 mm and 350 mm).

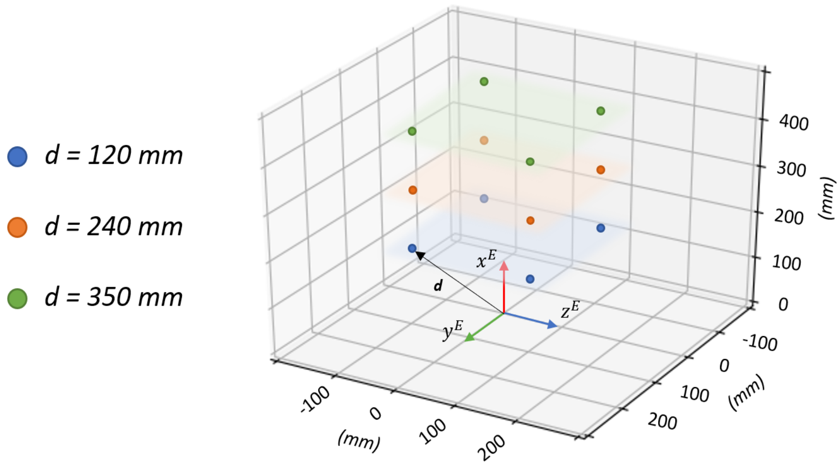


Figure 5.2: Flow chart of the new magnetic localisation algorithm.

Moving to the *Static-Dynamic test*, the simulated capsule is spawned at 15 cm from the magnet center and moved along the trajectory shown in Figure 5.3. As previously mentioned, it involves variations of all the state variables. According to the first set of tests, the absolute estimation error is evaluated after every iteration of the algorithm and used at the end for the computation of the average absolute error and standard variation (Table 5.2).

$e_{d_x} \pm \sigma_{d_x} [mm]$	$e_{d_y} \pm \sigma_{d_y} [mm]$	$e_{d_z} \pm \sigma_{d_z} [mm]$	$e_{d_\Delta} \pm \sigma_{d_\Delta} [mm]$
66.37 ± 47.41	39.70 ± 15.24	22.38 ± 9.40	20.96 ± 9.22

Table 5.2: Dynamic-Static test: average estimation errors and standard variation.

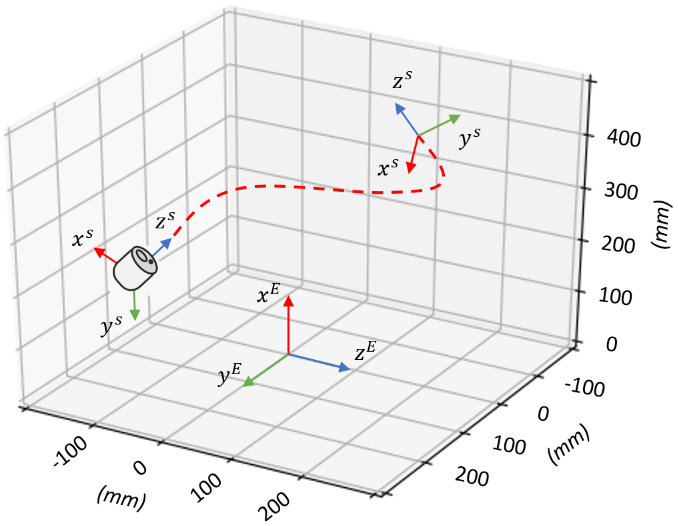


Figure 5.3: Flow chart of the new magnetic localisation algorithm.

By analysing the results, it is clear how the algorithm allows for a more precise pose estimation in static-static conditions while bigger estimation errors occur when the capsule moves around the EPM. The reason is probably linked to the role of the estimated pose that is used as *initial guess* by the employed Python solver (i.e scipy.optimize.root()). In fact, when the capsule endoscope is fixed, each iteration of the algorithm brings the initial guess closer to the solution. This does not happen during the dynamic-static test where the pose of the capsule is changing. Concerning this last test, it has been observed how the performance of the algorithm significantly worsens with the increase of the distance $\mathbf{d}$ between the capsule endoscope and the EPM; However, this behaviour was not observed when increasing the distance in the static-static test.

5.2 Future Work

The work presented here highlights the complexity of the magnetic localisation of the Magnetic Flexible Endoscope and demonstrates how unexpected technical problems strongly affect the development of new techniques for pose estimation; So, these limitations can not be neglected during the design stage. In particular, this work has spotted the constraint introduced by the Generalized Complete Elliptic Integral for the application of a Kalman Filter which needs process and measurement models in an explicit form. Despite of this limitation, this work shows promising results for the development of a new localisation algorithm whose performance can be comparable to the one achieved by the previous techniques. It is worth mentioning that further studies are needed to properly optimize the algorithm and to make it fit with the real platform. Among the future works, the development of a Kalman Filter which is able to work with implicit measurement model surely would bring benefits to the new estimation technique: the system of nonlinear equations that is currently solved outside the filter with a computational expensive Python function (i.e. scipy.optimize.root()) could be directly used as implicit measurement model and allows for a faster pose estimation. Satisfactory results could be achieved by employing two localisation techniques running on two separate threads with different publication frequencies: the Particle Filter based one could be employed to provide with a lower rate more accurate results that are used to adjust the estimation of the computational less expensive new algorithm.

Moving to the developed simulator of the MFE, it could be used in future to allow interested students and researches to start becoming familiar with the working principles of the real robotic platform. It is worth noting that the MFE Simulator will never reached the same degree of complexity of the physical system due to parameter uncertainties and unmodeled dynamics. However, some improvements could be made to make it more robust and closer to the MFE. First and foremost, it could be re-written in C++ to improve the performances in term of operating speed. This first improvement would give the opportunity to reach higher operating frequencies and ensure the needed one of 100 Hz. It is worth mentioning that the current ROS simulator operates in the range of *80Hz - 90Hz*. In addiction, it would be interesting to implement the Goertzel Algorithm coherently with the real platform: it has been verified that the *reconstructed* coil magnetic field does not respect the Generalized Complete Elliptic Integral anymore. So, the Goertzel Algorithm must be considered during the design of the new magnetic localisation. Then, the algorithm could be tested on the robotic platform to validate the theoretical results obtained with the simulator and to highlight possible drawbacks. Finally, a comparison between the magnetic fields data sensed by the Hall-Effect sensors and the simulated ones could be useful to confirm the reliability of the MFE simulator.

Bibliography

[1] Ahoarau. iiwa_description kernel description, 2017.

[2] Isao Aoki, Akio Uchiyama, Kenichi Arai, Kazushi Ishiyama, and Shin Yabukami. Detecting system of position and posture of capsule medical device, July 6 2010. US Patent 7,751,866.

[3] Donald W Chakeres, Alayar Kangarlu, Harisios Boudoulas, and Donn C Young. Effect of static magnetic field exposure of up to 8 tesla on sequential human vital sign measurements. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 18(3):346–352, 2003.

[4] Gastone Ciuti, Pietro Valdastri, Arianna Menciassi, and Paolo Dario. Robotic magnetic steering and locomotion of capsule endoscope for diagnostic and surgical endoluminal procedures. *Robotica*, 28(2):199, 2010.

[5] Norman Derby and Stanislaw Olbert. Cylindrical magnets and ideal solenoids. *American Journal of Physics*, 78(3):229–235, 2010.

[6] Christian Di Natali, Marco Beccani, Nabil Simaan, and Pietro Valdastri. Jacobian-based iterative method for magnetic localization in robotic capsule endoscopy. *IEEE Transactions on Robotics*, 32(2):327–338, 2016.

[7] Christian Di Natali, Marco Beccani, and Pietro Valdastri. Real-time pose detection for magnetic medical devices. *IEEE Transactions on Magnetics*, 49(7):3524–3527, 2013.

[8] Christian Di Natali, Marco Beccani, Pietro Valdastri, and Keith L Obstein. Real-time pose and magnetic force detection for wireless magnetic capsule, November 26 2019. US Patent 10,485,409.

[9] Flybrianfly. Bolderflight/MPU9250 kernel description, 2017.

[10] Gerald Goertzel. An algorithm for the evaluation of finite trigonometric series. *The American Mathematical Monthly*, 65(1):34–35, 1958.

[11] Chao Hu, Mao Li, Shuang Song, Rui Zhang, Max Q-H Meng, et al. A cubic 3-axis magnetic sensor array for wirelessly tracking magnet position and orientation. *IEEE Sensors Journal*, 10(5):903–913, 2010.

[12] Invensense. Ps-mpu-9250a-01: Mpu-9250 product specification revision 1.1, June 2016.

[13] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.

[14] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[15] Min-Gyu Kim, Yeh-Sun Hong, and Eun-Joo Lim. Position and orientation detection of capsule endoscopes in spiral motion. *International Journal of Precision Engineering and Manufacturing*, 11(1):31–37, 2010.

[16] Zheng Li and Philip Wai-Yan Chiu. Robotic endoscopy. *Visceral medicine*, 34(1):45–51, 2018.

[17] Sebastian Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25:113–118, 2010.

[18] Robert Mahony, Tarek Hamel, and J-M Pflimlin. Complementary filter design on the special orthogonal group so (3). In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1477–1484. IEEE, 2005.

[19] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control*, 53(5):1203–1218, 2008.

[20] Federico Masiero. Magnetic Field Modeling kernel description, 2020.

[21] Takashi Nagaoka and Akihiko Uchiyama. Development of a small wireless position sensor for medical capsule devices. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 2137–2140. IEEE, 2004.

[22] Mercedes Navarro, Andrea Nicolas, Angel Ferrandez, and Angel Lanas. Colorectal cancer population screening programs worldwide in 2016: An update. *World journal of gastroenterology*, 23(20):3632, 2017.

[23] Marco Pirotta. Singularity-free localization of a magnetic medical capsule for colonoscopy, 2016.

[24] Vincent Schlageter, Predrag Drljaca, Radivoje S Popovic, and Pavel KuČERA. A magnetic tracking system based on highly sensitive integrated hall sensors. *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, 45(4):967–973, 2002.

[25] Rebecca L Siegel, Kimberly D Miller, Ann Goding Sauer, Stacey A Fedewa, Lynn F Butterly, Joseph C Anderson, Andrea Cercek, Robert A Smith, and Ahmedin Jemal. Colorectal cancer statistics, 2020. *CA: a cancer journal for clinicians*, 2020.

[26] Piotr R Slawinski, Keith L Obstein, and Pietro Valdastri. Capsule endoscopy of the future: What's on the horizon? *World Journal of Gastroenterology: WJG*, 21(37):10528, 2015.

[27] Piotr R Slawinski, Addisu Z Taddese, Kyle B Musto, Keith L Obstein, and Pietro Valdastri. Autonomous retroflexion of a magnetic flexible endoscope. *IEEE Robotics and Automation Letters*, 2(3):1352–1359, 2017.

[28] Addisu Z Taddese, Piotr R Slawinski, Marco Pirotta, Elena De Momi, Keith L Obstein, and Pietro Valdastri. Enhanced real-time pose estimation for closed-loop robotic manipulation of magnetically actuated capsule endoscopes. *The International journal of robotics research*, 37(8):890–911, 2018.

[29] T.Jaber. MPU6050 6 axis module kernel description, 2019.

[30] Trainman419. teleop_twist_keyboard kernel description, 2020.

[31] Aurélien Valade, Pascal Acco, Pierre Grabolosa, and Jean-Yves Fourniols. A study about kalman filters applied to embedded sensors. *Sensors*, 17(12):2810, 2017.

[32] Sören Vogel, Hamza Alkhatib, and Ingo Neumann. Iterated extended kalman filter with implicit measurement equation and nonlinear constraints for information-based georeferencing. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1209–1216. IEEE, 2018.

[33] W Weitschies, J Wedemeyer, R Stehr, and L Trahms. Magnetic markers as a non-invasive tool to monitor gastrointestinal transit. *IEEE transactions on biomedical engineering*, 41(2):192–195, 1994.

Appendix A

Arduino Code for Arduino Uno Board

```
void setup() {
  Serial.begin(115200);
  while(!Serial) {}
  status = IMU.begin();
  if (status < 0) {
    Serial.println("IMU initialization unsuccessful");
    Serial.println("Check IMU wiring or try cycling power");
    Serial.print("Status: ");
    Serial.println(status);
    while(1) {}
  }
  digitalWrite(A4,LOW);
  digitalWrite(A5,LOW);
}
void loop() {
  IMU.readSensor();
  Serial.print(IMU.getAccelY_mss(),4);
  Serial.print(" ");
  Serial.print(IMU.getAccelX_mss(),4);
  Serial.print(" ");
  Serial.print(-IMU.getAccelZ_mss(),4);
  Serial.print(" ");
  Serial.print(IMU.getGyroY_rads(),4);
  Serial.print(" ");
  Serial.print(IMU.getGyroX_rads(),4);
  Serial.print(" ");
  Serial.print(-IMU.getGyroZ_rads(),4);
  Serial.print(" ");
  Serial.print(IMU.getMagX_uT(),4);
  Serial.print(" ");
  Serial.print(IMU.getMagY_uT(),);
  Serial.print(" ");
  Serial.println(IMU.getMagZ_uT(),4);
  delay(10);
}
```

Appendix B

C Code for the STM32F411RE Board

The code shown in this appendix implements the loop function that runs on the STM32 Nucleo board. B.1 perform the initialization of the IMU module: after the setting of the needed registers, the full-range scales for the accelerometer and gyroscope and the cutting frequency for an embedded digital low pass filter are selected.

B.1 Main

```c

int main(void)
{
    MPU_ConfigTypeDef myMpuConfig;
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_I2C1_Init();
    MX_USART2_UART_Init();
    MPU9250_WakeUp();
    MPU6050_Init(&hi2c1);
    printf("MPU9250 Initialization completed.\n");

    myMpuConfig.Accel_Full_Scale = AFS_SEL_2g;
   myMpuConfig.ClockSource = Internal_8MHz;
   myMpuConfig.CONFIG_DLPF = DLPF_260A_256G_Hz;
   myMpuConfig.Gyro_Full_Scale = FS_SEL_500;
   myMpuConfig.Sleep_Mode_Bit = 0;
   MPU6050_Config(&myMpuConfig);
   printf("\nMPU9250 Configuration complete.\n");
   HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_SET);
   printf("\nPress the User Push Button to start the calibration\n");
}
```

B.2 Endless loop

```
while (1)
    {
       if (HAL_GPIO_ReadPin(GPIOC, GPIO_PIN_13) == GPIO_PIN_RESET && flag ==
           false)
       {
          printf("\nCalibrating. Do NOT move the device.\n");
          HAL_Delay(1000);
          for (count = 0; count < 600; ++count)
          {
                 MPU6050_Get_Accel_Scale(&myAccelScaled);
               MPU6050_Get_Gyro_Scale(&myGyroScaled, GyroCalib);
                G_sum_X += myGyroScaled.x;
               G_sum_Y += myGyroScaled.y;
               G_sum_Z += myGyroScaled.z;
               HAL_Delay(10);
          }
          GyroCalib.x = G_sum_X/count;
          GyroCalib.y = G_sum_Y/count;
          GyroCalib.z = G_sum_Z/count;
          printf("Calibration is successfully completed.\n Data are available
              on the serial port COM3\n");
          flag = true;
       }
       if (flag == true)
       {
          char buffer[65] = {0};
          val += 1;
          MPU6050_Get_Accel_Scale(&myAccelScaled);
          MPU6050_Get_Gyro_Scale(&myGyroScaled, GyroCalib);
          sprintf(buffer, "\r\n%.4f %.4f %.4f %.4f %.4f %.4f ",
              myAccelScaled.x, myAccelScaled.y,myAccelScaled.z, myGyroScaled.x,
              myGyroScaled.y, myGyroScaled.z);
          uprintf(buffer);
          HAL_Delay(10);
       }
    }
```

Appendix C

Visual Basic Code for the computation of C

The Visual Basic code below has been developed by Derby *et al.* [5] and guarantees a closed form solution of the Generalized Complete Elliptic Integral.

```
REM macro for Calc spreadsheet
FUNCTION cel(kc, p, c, s)
    IF (kc = 0) THEN
        cel = "NaN"
        EXIT FUNCTION
    ENDIF
    errtol = .000001
    k = ABS(kc)
    pp = p
    cc = c
    ss = s
    em = 1.
    IF (p > 0) THEN
        pp = SQR(p)
        ss = s/pp
    ELSE
        f = kc*kc
        q = 1. - f
        g = 1. - pp
        f = f - pp
        q = q*(ss - c*pp)
        pp = SQR( f/g )
        cc = (c - ss)/g
        ss = - q/(g*g*pp) + cc*pp
    ENDIF
    f = cc
    cc = cc + ss/pp
    g = k/pp
    ss = 2*(ss + f*g)
    pp = g + pp
    g = em
    em = k + em
    kk = k
    WHILE ( abs( g - k ) > g*errtol )
        k = 2*SQR(kk)
         kk = k*em
        f = cc
        cc = cc + ss/pp
        g = kk/pp
        ss = 2*(ss + f*g)
        pp = g + pp
        g = em
        em = k + em
    WEND
    cel = (pi/2.)*(ss + cc*em)/( em*(em + pp) )
END FUNCTION
```

Appendix D

Python Code for the Magnetic Flexible Endoscope Simulator

The pieces of code shown in this appendix are used to implement the MFE Simulator in ROS. Although the entire code is not provided here, a line-by-line explanation of the Python file is provided at the end of Appendix D.

D.1 Main endless loop

```python

while not rospy.is_shutdown():
    if inputQueue.qsize() > 0.0:
        key = inputQueue.get()
    if flag:
        if MPU6050Queue.qsize() > 0.0:
            data = MPU6050Queue.get()
        q = madgwick_accel_gyro(data[0], data[1], data[2], data[3], data[4], data[5], q)
        [roll, pitch, yaw] = euler_from_quaternion([q[1], q[2], q[3], q[0]])
        real_pose.pose.orientation.x = q[1]
        real_pose.pose.orientation.y = q[2]
        real_pose.pose.orientation.z = q[3]
        real_pose.pose.orientation.w = q[0]
        real_pose.header.stamp = rospy.Time.now()
        [acc_x, acc_z, flag] = capsule.capsule_pose(key, real_pose, pub1)
        capsule.imu(q, acc_x, acc_z, pub2, pub13, data)

        try:
            (p_epm_caps, q_epm_caps) = magnet_capsule_lis.lookupTransform('capsule', 'magnet_center', rospy.Time(0))
            (p_caps_epm, q_caps_epm) = capsule_magnet_lis.lookupTransform('magnet_center', 'capsule', rospy.Time(0))
            q_epm_caps = q_epm_caps / np.linalg.norm(q_epm_caps)
```

```
        except (tf.LookupException, tf.ConnectivityException,
           tf.ExtrapolationException):
           continue

        epm(p_caps_epm, q_epm_caps, pub4, pub5, mag_epm)
        coil(p_caps_epm, q_epm_caps, pub6, pub7, pub8, pub9, pub10, pub11,
           mag_coil)
        rospy.Subscriber("mac/estimated_pose", PoseStamped, EstimateCallback)
        rate.sleep()
```

D.2 Madgwick Filter for Accelerometer and Gyroscope data

Implementation of the Madgwick Filter with linear accelerations and angular velocities as input data. The roll and pitch angles are reliable while the yaw angle is not.

```

def madgwick_accel_gyro(ax, ay, az, gx, gy, gz, q):
    f = 100.0
    dt = 1/f

    Kp = rospy.get_param("Madgwick_Kp")
    Ki = rospy.get_param("Madgwick_Ki")
    eInt = [0.0, 0.0, 0.0]
    q0 = q[0]
    q1 = q[1]
    q2 = q[2]
    q3 = q[3]

    a_norm = sqrt(ax * ax + ay * ay + az * az)
    ax = ax/a_norm
    ay = ay/a_norm
    az = az/a_norm

    vx = 2.0 * (q1 * q3 - q0 * q2)
    vy = 2.0 * (q0 * q1 + q2 * q3)
    vz = q0 * q0 - q1 * q1 - q2 * q2 + q3 * q3

    ex = ay * vz - az * vy
    ey = az * vx - ax * vz
    ez = ax * vy - ay * vx

    if Ki > 0.0:
        eInt[0] += ex
        eInt[1] += ey
        eInt[2] += ez
    else:
```

```
        eInt = np.zeros(3)

    gx = gx + Kp * ex + Ki * eInt[0]
    gy = gy + Kp * ey + Ki * eInt[1]
    gz = gz + Kp * ez + Ki * eInt[2]

    [pa, pb, pc] = [q1, q2, q3]
    q0 = q0 + (-pa * gx - pb * gy - pc * gz) * (0.5 * dt)
    q1 = pa + (q0 * gx + pb * gz - pc * gy) * (0.5 * dt)
    q2 = pb + (q0 * gy - pa * gz + pc * gx) * (0.5 * dt)
    q3 = pc + (q0 * gz + pa * gy - pb * gx) * (0.5 * dt)
    q = normalizeQuaternion(q0, q1, q2, q3)
    return q
```

D.3 Line by line explanation of the code of the MFE Simulator

In this appendix a line-by-line explanation of the code for the MFE Simulator is provided to the user. Note that the numbers of the lines are referred to the final version of the Python file mfe_simulator.py that is available in the STORM lab Github repository.

Lines 1 - 21: All the needed libraries and ROS messages are included. AxialCylindricalEllipticMap can be found in the repository of the MFE. DrawEPMandCoil is defined in the file plot.py.

Lines 23 - 39: This message can be printed on the terminal when the simulator is running to help the user with the keyboard teleoperation of the capsule.

Lines 42 - 58: The Python class *CapsulePose* is defined. *dt* is coherent with the frequency of 100Hz. *moveBindings* includes the set of keyboard keys for the teleoperation. *speedBindings* groups the keyboard keys to change the maximum speed.

Lines 60 - 80: Initialization method and class attributes of the *CapsulePose*. The function*passed_key* is used to detect the pressed keyboard keys for the teleoperation. Function call at line 426.

Lines 82 - 153: *Lines 82-106*: according to the pressed keyboard keys, the target speed along x and z of the capsule is set either to 0.1 m/s or to 0.0 m/s. *Lines 107-133*: The velocity is increased of 0.0001 m/s at each cycle until reaching the target value. Linear accelerations along x and z are simply computed as $(v_{fin} - vin)/\Delta t$. *Lines 135-145*: Firstly, the increments of the position are computed with respect to the Capsule coordinate frame. Then, they are moved in the World RF by exploiting the capsule orientation. Finally, the new position is evaluated and published as PoseStamped ROS message. *Lines 146-153*: The new capsule position is moved in the EPM frame for future representation purposes. Linear accelerations are returned. When Ctrl+C ('/x03') is pressed, flag = False and the simulator is shut down.

Lines 155 - 178: The function *imu* is used to publish the values of acceleration and angular velocity sensed by the MPU9250 on ROS Topic */MAC/imu/raw*. Accelerations due to keyboard teleoperation are also considered in the computations. All the values are referred to the *imu_link* frame.

Lines 181 - 204: The EPM magnetic field sensed by the simulated capsule is evaluated in the EPM frame by means of the previously computed look-up table. Gaussian noise with mean value equal to 10 μT is added to simulate the electromagnetic disturbance. Then, the magnetic field is moved in the capsule frame knowing its orientation with respect to the World RF. Finally, the values are published on the ROS topics */MAC/epm/mfs1/raw* and */MAC/epm/mfs2/raw*. Note that the same orientation has been used for both the simulated Hall-Effect sensors; a simple arrangement of these values is needed if the configuration of the real capsule must be reproduced. All the data are referred to the *capsule* frame.

Lines 207 - 246: The Coil magnetic field sensed by the simulated capsule is evaluated in the EPM frame: to do that, the different orientation of the coil with respect to the external magnet must be considered. Gaussian noise with mean value equal to 10 μT is added to simulate the electromagnetic disturbance. As before, the magnetic field is moved in the capsule frame knowing its orientation with respect to the World RF. Finally, the values are published on the ROS topics */MAC/coil/mfs1/raw* and */MAC/coil/mfs2/raw*. All the data are referred to the *capsule* frame.

Lines 249 - 295: Implementation of the Madgwick Filter with acceleration and angular velocity data. The following steps are performed. *Lines 265-268*: normalization of the acceleration vector. *Lines 270-272*: estimation of the gravity direction. *Lines 274-276*: computation of the correction factor by means of the cross product between the estimated direction and the measured one. *Lines 278-283*: application of the integral feedback (only if enabled). *Lines 285-287*: Application of the proportional terms. *Lines 289-295*: Integration of the rate of change and quaternion normalization.

Lines 298 - 344: Implementation of the Madgwick Filter with acceleration, angular velocity and magnetic filed data. The following steps are performed. *Lines 313-316*: normalization of the acceleration vector. *Lines 327-332*: Gradient descent algorithm corrective step. *Lines 333-338*: application of the integral feedback. *Lines 339-344*: Integration of the rate of change and quaternion normalization.

Lines 347 - 357: Function for general quaternion normalization.

Lines 360 - 394: Functions used by the Madgwick Filter.

Lines 397 - 407: The function computes the inverse of the square-root of a number in a numerical efficient way.

Lines 410 - 438: This function is used to start a new thread that runs in parallel to the main process (Line 528). It is needed to read the pressed keyboard keys and add them in a queue. This method allows to teleoperate the capsule without affecting the operational frequency of 100Hz. When Flag == False, three plots are generated: the first one (*Lines 420-425*) shows the real Euler angles as a function of time. The second one (*Lines 4207-432*) shows the estimated angles as a function of time. The third one (*Lines 434-437*) shows the 3D trajectories, real and estimate, of the capsule with reference to the EPM.

Lines 441 - 457: The function MPU6050_val is used to read from the serial port the data sent by the chosen development board (Arduino or STM32). For this purpose, a separate thread is initialized. Once the data have been read, they are subdivided whenever a tab occurs; finally, they are saved in the array *data1* that is added to the queue of the thread. Note that magnetometer data are not used.

Lines 460 - 474: Function that is called whenever a new PoseStamped message is published on the ROS topic *mac/estimated_pose*. It is used to compute the estimated Euler Angles and append them for the final plots.

Lines 477 - 526: *Lines 480-482*: initialization of a separate thread that is used to read the pressed keyboard keys for the capsule teleoperation. *Lines 484-486*: initialization of a separate thread that is used to read the inertial data published on the serial port by the MPU9250. *Lines 488-501*: the list of the 11 published topics follows the initialization of the ROS node. *Lines 502-505*: initialization of the three transformation matrices between the capsule RF, EPM RF and World RF. Rate of the node set to 100Hz. *Lines 507-518*: buffers of undefined length are set to store data for the final plots. *Line 520*: starting quaternion for the Madgwick Filter. *Lines 524-525*: offline computation of the EPM and Coil maps. The input parameters are radius, high and magnetic remanence.

Lines 527 - 553: The main *while* begins. *Lines 528-529*: whenever a keyboard keys is pressed, the *key* is added to the queue and used as argument of the function *capsule.capsule_pose* (*Line 540*). *Lines 530-541*: when a set of inertial data is published by the MPU9250 on the serial port, the updated quaternion is evaluated and used to determine the orientation of the capsule with respect to the World frame. The attitude is also needed to compute the updated position through the function *capsule.capsule_pose*. *Lines 543-553*: the magnetic field maps previously computed need as input the relative position between the Capsule Frame and the EPM Frame. For this reason, two lookup-Transform *Lines 544-545* are defined to store the wanted data. With the function calls of *Lines 550-551*, the simulated magnetic field data are computed and published on the corresponding ROS Topics.

Lines 556 - 587: *Lines 558-560*: these parameters depends on the serial communication between the development PC and the board. The baund rate is fixed to 115200 bps while the serial port can change depending on the USB port used for the connection: possible examples are /dev/ttyACM0, /dev/ttyACM1 and /dev/ttyACM2.

Appendix E

Python Code for the new magnetic localization algorithm

The pieces of code shown in this appendix are used to implement the new magnetic localisation algorithm based on a Kalman Filter.

E.1 Measured state from magnetic fields data

The function implements the system of non-linear equations used by the Python command scipy.optimize.root(). Both the EPM and the Coil magnetic fields are modelled through the Generalized Complete Elliptic Integral. The additional needed parameters to solve the system are: initial condition x0 and a parameter vector

---

```python

def PositionFromEPMMeasurements(state, param):
    px = state[0]
    py = state[1]
    pz = state[2]
    yaw = state[3]

    # Generalized Complete Elliptic Integral: EPM
    br = rospy.get_param('epm_remenance')
    radius_ = rospy.get_param('epm_radius')
    height_ = rospy.get_param('epm_height')
    B_0_epm = br / pi
    rho = sqrt(square(px) + square(py))
    a = radius_
    b = height_ / 2
    zp = pz + b
    zm = pz - b
    alphap = a / sqrt(square(zp) + square(rho + a))
    alpham = a / sqrt(square(zm) + square(rho + a))
    betap = zp / sqrt(square(zp) + square(rho + a))
    betam = zm / sqrt(square(zm) + square(rho + a))
    gamma = (a - rho) / (a + rho)
    kp = sqrt((square(zp) + square(a - rho)) / (square(zp) + square(a + rho)))
```

```
    km = sqrt((square(zm) + square(a - rho)) / (square(zm) + square(a + rho)))
    Ca_val1 = alphap * C(kp, 1, 1, -1)
    Ca_val2 = alpham * C(km, 1, 1, -1)
    b_rho = B_0_epm * (Ca_val1 - Ca_val2)
    Cb_val1 = betap * C(kp, square(gamma), 1, gamma)
    Cb_val2 = betam * C(km, square(gamma), 1, gamma)
    b_axial = (B_0_epm * a / (a + rho)) * (Cb_val1 - Cb_val2)

    b_x_epm = b_rho * cos(atan2(py, px))
    b_y_epm = b_rho * sin(atan2(py, px))
    b_z_epm = b_axial

    # Generalized Complete Elliptic Integral: Coil
    br = rospy.get_param('coil_remenance')
    radius_ = rospy.get_param('coil_radius')
    height_ = rospy.get_param('coil_height')
    B_0_coil = br / pi
    rho = sqrt(square(py) + square(-pz))
    a = radius_
    b = height_ / 2
    zp = px + b
    zm = px - b
    alphap = a / sqrt(square(zp) + square(rho + a))
    alpham = a / sqrt(square(zm) + square(rho + a))
    betap = zp / sqrt(square(zp) + square(rho + a))
    betam = zm / sqrt(square(zm) + square(rho + a))
    gamma = (a - rho) / (a + rho)
    kp = sqrt((square(zp) + square(a - rho)) / (square(zp) + square(a + rho)))
    km = sqrt((square(zm) + square(a - rho)) / (square(zm) + square(a + rho)))
    Ca_val1 = alphap * C(kp, 1, 1, -1)
    Ca_val2 = alpham * C(km, 1, 1, -1)
    b_rho = B_0_coil * (Ca_val1 - Ca_val2)
    Cb_val1 = betap * C(kp, square(gamma), 1, gamma)
    Cb_val2 = betam * C(km, square(gamma), 1, gamma)

    b_x_coil = b_rho * cos(atan2(py, -pz))
    b_y_coil = b_rho * sin(atan2(py, -pz))
    b_z_coil = (B_0_coil * a / (a + rho)) * (Cb_val1 - Cb_val2)

    # Rotation from Capsule to World
    [roll, pitch, yaw_not_used] = euler_from_quaternion(param[2])
    q_caps_world = quaternion_from_euler(roll, pitch, yaw)

    # EPM: From World to epm RF
    caps2w_epm = quaternion_multiply(q_caps_world,
        quaternion_multiply([param[0].x, param[0].y, param[0].z, 0.0],
        quaternion_conjugate(q_caps_world)))
    B_epm_epm_RF = quaternion_multiply(param[3],
        quaternion_multiply(caps2w_epm, quaternion_conjugate(param[3])))
```

```
    # Coil: From World to epm RF
    caps2w_coil = quaternion_multiply(q_caps_world,
        quaternion_multiply([param[1].x, param[1].y, param[1].z, 0.0],
        quaternion_conjugate(q_caps_world)))
    B_coil_epm_RF = quaternion_multiply(param[3],
        quaternion_multiply(caps2w_coil, quaternion_conjugate(param[3])))

    f1 = b_x_epm - B_epm_epm_RF[0]
    f2 = b_z_epm - B_epm_epm_RF[2]
    f3 = b_x_coil - B_coil_epm_RF[0]
    f4 = b_z_coil - B_coil_epm_RF[2]
    return [f1, f2, f3, f4]
```

E.2 Pose estimation through Kalman Filter

Prediction and correction steps of the Kalman Filter based localisation algorithm.

```

# Prediction step
kf.predict(Q=Q)

# Choice of the initial guess
guess_state = guess_initial_state(epm_mfs1.magnetic_field,
    coil_mfs1.magnetic_field, q_caps_world, q_world_epm)

# Computation of the measurements vector
measurements = get_measurements(epm_mfs1.magnetic_field,
    coil_mfs1.magnetic_field, q_caps_world, q_world_epm, guess_state)

# Correction step
kf.update(R=R, measurements=measurements)
new_mean = kf.mean
(p_epm_world, q_epm_world) = epm_world_listener.lookupTransform('world',
    'magnet_center', rospy.Time(0))
old_state = [measurements[0], measurements[1], measurements[2], 0.0]
old_state = quaternion_multiply(q_epm_world, quaternion_multiply(old_state,
    quaternion_conjugate(q_epm_world)))
position = p_epm_world + old_state[0:3]
```