Aalto University
School of Science
Master's Programme in Human-Computer Interaction and Design

Thomas Langerak

Model Predictive Contour Control for Electromagnetic Induced Haptic Feedback

Master's Thesis

Espoo, July, 1, 2018

Supervisor: Antti Oulasvirta, Professor

Thesis advisor(s): Otmar Hilliges, Professor

AALTO UNIVERSITY
School of Science
Master's Programme in Human-Computer Interaction and Design

ABSTRACT OF THE
MASTER´S THESIS

Author:
Thomas Langerak

Title of the thesis:
Model Predictive Contour Control for Electromagnetic Induced Haptic Feedback

Number of pages:	Date:
109	01/07/2018

SCI3020
Human-Computer Interaction and Design

Supervisor:
Antti Oulasvirta, Professor

Thesis advisors:
Otmar Hilliges, Proffessor

The loss of handwriting after a stroke is one of the most impactful injuries for both patient and relatives. For that reason we introduce MagnetPen, a system that, by giving haptic guidance control feedback to the learner, increases the learning rate in tasks such as handwriting. The core of the system is an electromagnet on a 2D axis system, which is capable of, both, attracting and repelling a pen. In this thesis we present a process control strategy, Model Predictive Contour Control, that is responsible for the positioning and strength of the electromagnet with respect to the current pen position as well as the desired path. We find that, albeit being relatively slow, this strategy works in almost any case.

Keywords:	Publishing language:
HCI, Haptic, Control Theory	English

Preface

This thesis was conducted for the double-degree master in Human-Computer Interaction and Design at Aalto University and University of Twente. This thesis was done at the Advanced Interactive Technology group at ETH Zurich.

Many people helped to bring this thesis to a succesful end. Where some helped me overcome obstacles, others had, seemlingy infinite, patience while explaining me control theory. However, there are some people without whom this project would have definitely ended differently. First of all, I would like to thank Christoph Gebhard, Tobi Naegeli and Stefan Stevsic for their help in explaining, formalizing and implementing the Model Predictive Contour Control algorithm. Secondly, I would like to thank Juan Zarate for his tremendous help in defining a theoretical framework for the electromagnet. Next to this, I would like to thank my supervisors at ETH Zurich (Prof. Dr. Omtar Hilliges and Velko Vechev), Aalto University (Prof. Dr. Antti Oulasvirta) and the University of Twente (Dr. Mannes Poel and Dr. Ing. Gwenn Englebienne) for their valuable feedback. Finally, my sincere thanks to everybody at the Advanced Interactive Technologies group for making my stay a pleasant experience.

Wordcount: 19997, Pages: 113

Contents

Chapter 1
Introduction

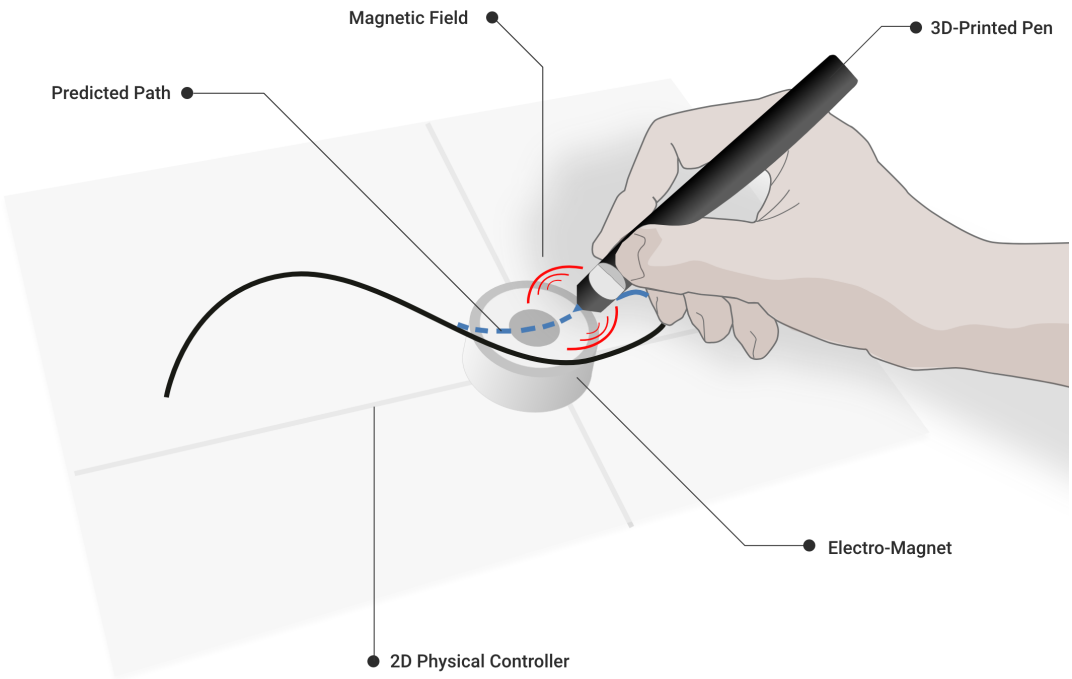


Figure 1.1: Sysem proposed in this thesis.

STROKES are a world-wide healthcare problem. In high-income countries strokes are among the most common causes for death. Most people, however, survive the initial injury. The biggest influence on them, and their relatives, is through long-term impairment. The most common type of this impairment is loss of motor control and affects about 80% of all stroke patients. Most affected patients report the loss of handwriting among the biggest disabilities [62]. A review on rehabilitation techniques and their effectiveness by Langhorne et al. [30] find that biofeedback, repetitive task training and robotic aids significantly improve the results in hand-related tasks. Especially when focusing on task-specific treatment. Also in arm-related tasks an improvement has been found, however to a smaller extent. The empirical evidence is based on standardize tests, such as the Fugyl-Meyer scale [11], however, this does not evaluate handwriting or drawing specifically.

Some attention has been paid on robotic aids for relearning writing after a stroke. Most notable are [65] and [40], both will be discussed in more detail in chapter 3.2. More emphasis has been placed on course motor tasks and are, generally, in the early stages of the rehabilitation process.

Examples of this are the works by Masiero et al. [35], Kahn et al. [23] and Lum et al. [33]. They, usually, involve human-sized exoskeletons that help with the arms motions; as depicted in Figure 1.2. What the latter works have in common is that they focus on tactile (or multi-modal) feedback rather than doing the movements for the user.

Figure 1.2: Example of an exoskeleton for rehabilitation [15].

However, stroke patients are not the only ones who need to (re-)learn specific motor tasks. Other target groups include children and aspiring artists, designers and architects. Calling for a more generalized approach.

This project, illustrated if Figure 1.1, in a broad sense, will be about building a system that will help with the (re-)learning of simple motor tasks. We will approach this problem from a very general perspective and not specifically targeting a specific user group. Instead, we will use a more technology-orientated approach by investigating the use of electromagnetism in guidance haptic feedback on a modified pen. In this thesis, which is part of the broader project, we will explore the control theory aspect of such as system. The intelligent control of haptic feedback has received limited attention, yet is vital for the success of haptic devices. we will gain inspiration specifically from the control of CNC machines. Although the focus of this thesis is on the control theory, the work is part of a larger project including the construction of the final hardware device. Since the theory has eventually to be implemented in the hardware one cannot consider the control theory isolated from the hardware. Therefore some reference to the hardware is made in this thesis.

Finally, we would like to stress that all code, with the exception of the solutions provided

by the ForcesPro[1] software has been made available as open-source on GitHub: `github.com/tlangerak/master-thesis`

[1] As I will elaborate on later; Forces Pro is commercial software that helps to solve some of our modeling problems.

Part I

Preceding Research

In this part we will discuss prior research. This includes background literature regarding motor skills and learning as well as theory on haptic perception. We will find that the theory supports our concept of electromagnetic induced haptic feedback for learning drawing tasks. As well as a lack of systems designed for this.

Chapter 2
Related Literature

IN order to have a clear understanding of the problem at hand, as well as to gain the necessary knowledge we will expend on a variety of related literature. Since the diverse nature of this project this chapter will also discuss a variety of topics. At first sight they might be unrelated, however, the common ground they share is that they all provide the required knowledge to bring this project to a successful end. we will start with discussing human motor control, muscle memory and learning of motor tasks. Secondly, we will discuss haptics, both from a perception as well as technology point of view. This chapter will be finalized with an analyses on how haptic can be used for improved motor control and learning.

2.1 Motor Control, Muscle Memory & Learning

2.1.1 Motor Control

Movements are made in a response to stimuli. Either an external signal (e.g. the change of colour of a traffic light) or an internal signal (e.g. remembering something). Some movements are automatic and happen before the signal can be processed by the brain (usually in case of pain) whereas other movements are deliberate and precise. In short, there are many different movements that involve different stimuli and information processing. The research domain on motor control can be roughly divided into two subsets. First of all, the physiological area and secondly, the psychological area. The former one is interested in the functional relations among physical elements. Where as in the latter processes and structures are discussed without necessary reference to their physical realization.

When taking a look at the physiological aspect of motor control we see that there are many levels to be taken into regard. From muscles, to synapses, to the motor cortex. Rosenbaum [51] has a good starting point in his book Human Motor Control to gain a understanding of the physiological aspects involved. Most of it is beyond the scope of this thesis. On a very general level, in the case of an external stimuli with a deliberate action, a signal gets send from the receptor to the motor cortex in the brain. This receptor could be for instance nerves sensitive to touch or the rods and cones in one's eye. The brain processes these stimuli and formulates an appropriate sequence of responses. The responses are send by the nervous system towards the appropriate muscles which move according the stimuli. This movement is controlled with a feedback loop, including sensors that measure muscle contraction.

One of the most well-researched topics from the psychological perspective is the serial-order problem. This problem is regarding how movements are sequenced. How does one go from input towards output. One of the most accepted theories is the hierarchy theory [51]. This theory

states that this process is hierarchical in nature, in which feedback is given to different 'levels' in the hierarchy. For example drawing, which, in its most basic division, can be divided into planning and execution; both influencing each other. However, both could be more easily further hierarchical defined. In our work will mainly focus in improving the latter by, once a plan is made, improving the execution of the plan on a muscle control level. When formulating this more from a physiological perspective; formulating the right muscle actuation based on certain stimuli.

2.1.2 Muscle Memory

"Muscle memory is a term commonly used in everyday discourse for the sort of embodied implicit memory that unconsciously helps us to perform various motor tasks we have somehow learned through habituation, either through explicit, intentional training or simply as the result of informal, unintentional, or even unconscious learning from repeated prior experience. In scientific terminology, such memory is often designated as procedural memory or motor memory because it enables us to perform various motor procedures or skills in an automatic or spontaneous fashion, without conscious deliberation of how the procedure should be followed and without any explicit calculation of how one identifies and achieves the various steps involved in the procedure and how one proceeds from step to step."

The definition above by Shusterman [56], though not concise, is an interesting one as it elaborates on certain key factors that should be discussed while discussing muscle memory (or motor memory). The first aspect is *unconsciously*; indicating without explicit attention to the motion or action. The for instance is useful while writing, one could focus on formulating sentences rather than how to write a certain letter. When framing this in the hierarchy theory outlined above, it means that the lower level are executed without direct attention. leaving more room for higher level hierarchies. Secondly, Shusterman elaborates on *learning*; a motor memory has to be learned (either intentional or unintentional). In which he emphasis that it learned through *repeated* experiences. This raises the question; how does a person learn a motor task?

2.1.3 Motor Learning

Schmidt et al. [55] defines motor learning as "a set of processes associated with practice or experience leading to relatively permanent changes in the capability for skilled movement". He continues to outline for characteristics of this definition: "(1) Learning is a process of acquiring the capability for producing actions. That is, learning is the set of underlying events, occurrences, or changes that happen when practice enables people to become more skilled at some task. (2) Learning occurs as a direct result of practice or experience. (3) Learning cannot be observed directly, as the processes leading to changes in behavior are internal and are usually not available for direct examination; rather, one must infer that learning processes occurred on the basis of the changes in behavior that can be observed. (4) Learning is assumed to produce relatively permanent changes in the capability for skilled behavior."

Within learning there are two types of feedback to the user involved. First of all, *inherent feedback*, feedback that is intrinsic to the task and secondly, *augmented feedback*; supplementary to the task. According to Schmidt et al. much research suggests that augmented feedback is the single most important variable for learning. Augmented feedback can take many forms; it can be a trainer giving auditory offline (after the task is complete) feedback, a technical interface giving tactile online (during the task) feedback and everything between. Furthermore, augmented feedback has three main roles in the learning process. First of all it acts as information, secondly it acts to form associations between movement parameters and resulting action, finally, it acts in a motivational role.

There are numerous theories that try to explain motor learning. However, none of them is capable of explaining all the evidence and observations in motor learning. The two major theories are the *closed-loop theory* [1] and *schema theory* [54]. The former holds that the learner acquires a reference of correctness through practice. The increase in performance is the increased capability of the learner to use this reference on a closed-loop control. The reference of correctness (or perceptual trace) is stored in memory. The inherent feedback that is produced when making a movement leave a trace that is compared to the internal reference of correctness. The user acts on the difference between the feedback and the trace to reduce the error between them.

Schema theory separates the movement and evaluation into *recall memory* and *recognition memory* respectively. The speed of the movement determines the involvement of both. Where in a fast, explosive movement the focus is on the recall memory and evaluation only happens after the movement; in a slow movement the movement is evaluated during the movement. At the heart of this theory is the idea that a generalized motor program is chosen for a movement to which parameters are added (such as relative timing) in order to specify the particular motion. After a movement there are four source of information available: (1) initial state of the body, (2) the parameters added to the program, (3) augmented feedback and (4) sensory/inherent feedback. These are used for learning in the two memories.

However, both theories make evident that augmented feedback could be beneficial. Whereas the schema theory mentions it explicit as a source of feedback, the closed-loop theory requires further analyses. However, it could be argued that augmented feedback will enlarge the overall feedback, creating a stronger reference of correctness and thereby increasing the learning speed. Secondly, when the haptic feedback increases the accuracy of the motor movement the reference of correctness is more correct. This could also increase the overall learning speed.

2.1.4 Measuring Learning

The third characteristic outlined by Schmidt et al. is especially interesting, since it outlines one of the fundamental questions in motor learning experiments is, ”how to measure and quantify learning?” Although we do not attempt to measure learning with users in this thesis, we will extent on this a bit since it will increase the overall understanding of what learning actually is in this context.

The most common way to illustrate the learning is by a *performance curve*. In which an error measure priorly defined is plotted on the vertical axes, the horizontal axes is the number of the trail. This shows chronological progress over time. An arbitrary example is shown in Figure 2.1.

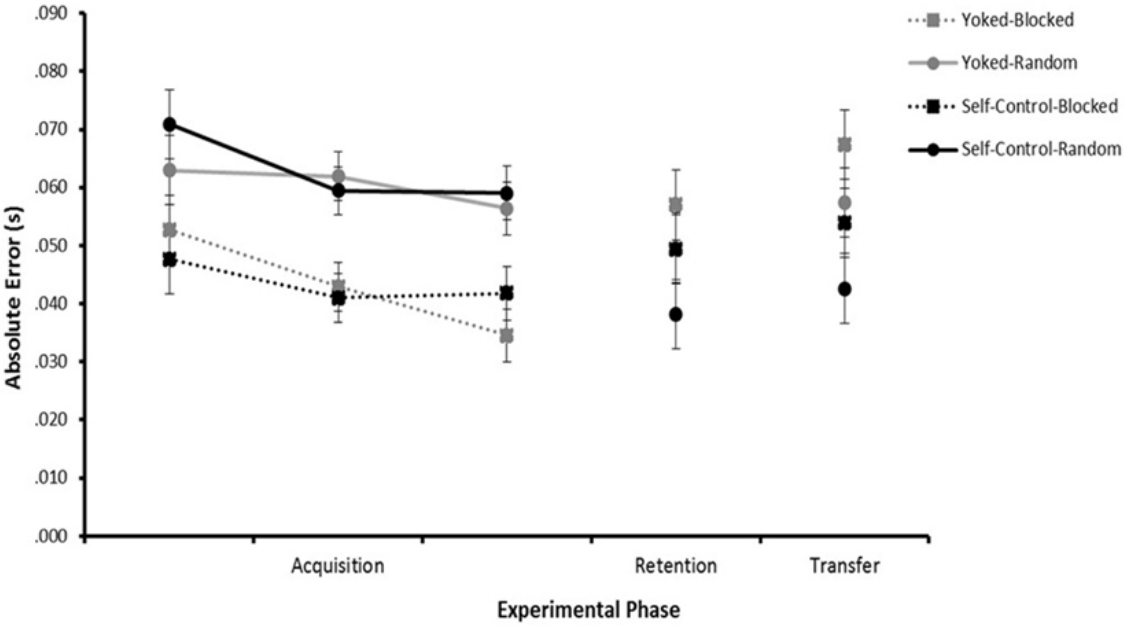


Figure 2.1: Example of different stages in learning in an experimental setting [2].

However, Schmidt et al. also elaborates on some problems involving this methodology. First of all he argues that a performance curve should not be confused with a learning curve. Since, performance curves also incorporate momentary changes; the early definition is solely about permanent changes. Secondly, this type of graph also leads to wrong conclusions due to between-subject variability. Since, usually, the performance is averaged it is likely to lose noise, but also key observations on the learning progress. Thirdly, the performance measure has a large influence on the shape of the curve and can lead to false conclusions.

Schmidt et al. continue to offers solutions to these problems. First of all, they suggest using a retention method. In which the experiment is divided in two stages. The first stage is acquisition and is similar to what is outlined above. However, a retention stage is added. After a prolonged break between the two stages, the same task is carried out again. The idea behind this, is that it able to measure more permanent changes. Secondly, Schmidt et al. argues for comparing two or more groups for the same task. So that relative rather than absolute conclusions can be drawn.

2.2 Haptics - Perception & Technology

The term *haptic* was first coined by the German Psychologist Dessoir in 1892. It was suggested as a name for academic research into the sense of touch. It was derived from the Greek: haptós

"palpable" and haptikós "suitable for touch." Perhaps more accurate is the definition by Gibson [16]: "the sensibility of the individual to the world adjacent to his body by use of his body." It emphasizes the perception through to body, as well as that it is of a physical exploratory nature.

Haptics relies on the forces experienced during touch [50]. Haptic technology is applying forces to the body of an individual to mimic a (virtual) object. This usually done by means of force feedback and the help of motors [49]. However, over the past years the term haptic technology is used more generalized and concerns itself with tactile feedback to the user as well as the available technological means has increased rapidly.

To illustrate the importance of haptic feedback we have to ask ourselves one simple question: "What would happen if we lose our sense of touch?" Robles-De-La-Torre [48] argues that the sense of touch is highly underrated. Furthermore, he argues that the loss of sense of touch cannot be adequately compensated by sight. It is especially striking to take a closer look at the case of Mr. Ian Waterman [7] who lost his sense of touch. Due to an autoimmune infection his sensory nerves were destroyed, however not the nerves for motor control. He could move his limbs, but not control them precisely. Yet this precise control, allows us to walk, to stand or to sit up. Through sheer force of will Mr. Waterman was able to regain some control back, replacing tactile feedback with sight. However, the consequences were devastating, he has to look where is arm is going when grabbing something in order to control the motion. He tends to grab things with too much force and fine motor tasks, such as handwriting, almost exceeds his cognitive capabilities. However, this is not only the case with Mr. Waterman. The effect of the loss of the sense of touch can be replicated by using local anesthesia [22]. It can even experienced at home, when trying to walk while with leg which feels like it is "sleeping." These experiments show generalizability of this phenomenon.

In modern day interactions with computers a lot of emphasis is placed on the virtual aspects and very little tactile feedback is provided. As we have seen with Mr. Waterman this severely limits performance in a variety of tasks. Although the analogy between Mr. Waterman and interacting with a virtual environment shows limitations, it still illustrates the importance of the perception of touch. A study by Finley and Nguyen [14] indicates the same. It evaluated the quality of robotic surgery with and without haptic feedback, finding a significant higher quality in the former (though more modern literature questions the results [60] and calls for more research).

Perhaps closer to home, touch screens also have limited haptic feedback. Buxton et al. [5] was probably the first to identity this inherent problem with touch screens: "Perhaps the most difficult problem is providing good feedback to the user when using touch tablets. For example, if a set of push-on/push-off buttons are being simulated, the traditional forms of feedback (illuminated buttons or different button heights) cannot be used. Also, buttons and other controls implemented on touch tablets lack the kinesthetic feel associated with real switches and knobs. As a result, users must be more attentive to visual and audio feedback, and interface designers must be freer in providing this feedback." This work, obviously, has to be evaluated in the time it was published: 1985. However, their analysis still holds largely true today.

Despite the origins of computer haptics being in enhancing human-computer interaction (as proposed by Buxton et al.), projects such as HHA by Mullins et al. [40] and dePENd (Yamaoka and Kakehi [65]) show that the field is surpassing this domain. In these projects the goal of the haptic feedback is not to improve the interaction with a computer system (often further specified to a GUI), but to improve the learning rate of motor skills. Indeed, haptics gain ground in physiotherapy sessions [32, 53, 47, 38] and is reported to decrease injuries, increase effectiveness of exercise and increase motivation [32].

In order to achieve haptic feedback several important elements are needed. The first element is a sensor, the second an actuator and the third element is a model; how should the actuator react based on the sensor readings. The sensor largely depends on what measurements are needed. Potentiometers, flex-sensors, computer vision, infrared sensors, to just name a few. The limitations in this domain are near limitless. The actuators for haptic feedback are also numerous, however, more easily divided into categories. Common are the vibrations motors, to provide rough feedback. Also DC motors are used, mostly to increase resistance in a certain movement. Solenoids are used to generate impact. Finally magnets are used to guide and control devices. Ofcourse this list is limited and extremely generalized.

2.3 Learning Motor Tasks Through Haptic Feedback

We have seen that augmented feedback is widely accepted to effectively enhance motor learning, even to the stage that it can become muscle memory. We have also illustrated that haptic feedback is seen as viable source for augmented feedback and its importance. However, we have not drawn the connection between haptic feedback and motor learning clearly beyond this point. Something we will extended on now. When taking a look at a modern-day review, such as the work by Sigrist et al. [57] it can be concluded that there are relatively many concepts surrounding haptic feedback for motor learning however relatively little empirical proof for its effectiveness.

Compared to other types of feedback (auditory or visual) haptic feedback has the unique property of being bidirectional. This means that the haptic sense is the only sense that allows humans to act with the world around them and at the same time perceive these interactions. Hence, it seems logical to use haptic feedback for motor learning [17].

Sigrist et al. [57] divide the haptic feedback research into several sub-catergories. First of all they elaborate on *position-control-based* haptic guidance, such as dePENd [65] we will discuss in prior work. Precisely, it is a method that controls movement and timing fully. That the theoretical foundation for this approach is flawed for successful motor learning was shown by Scheidt et al. [52]. They illustrated that the learning time was prolonged by about a factor 15 when subjects were prevented from making errors. Illustrating that actively correcting a mistake increases the learning rate. The findings by Sigrist et al. are that position control performs relatively poor compared to visual and auditory feedback. However, it improves the overall system when incorporating it in multi-model interaction. Furthermore, they highlight the instructional nature of

this type of feedback. It allows to illustrate a movement to the user they have never seen before.

The second category Sigrist et al. identify is *vibrotactile* haptic feedback. These systems have mainly been developed to reduce the workload of the visual and auditory system. They mainly call for more research in this area, since locations where vibrotactile actuators should be placed are unclear. Furthermore these types of actuators are unable to offer precise and localized feedback.

The final category is *haptic guidance*. This is the category that closely resembles my project. Compared to position-control-based haptic guidance it does not control the path and/or timing fully and only to a limited extend. Thereby it allows for the active correction of errors by the subject. Next to the general advantages of augmented feedback it shown that haptic guidance specifically has the following advantages: (1) strengthening of muscles and connective tissue, provoking motor plasticity and preventing stiffening; (2) somatosensory stimulation inducing brain plasticity and (3) reinforcement of the movement pattern by movement repetitions [34]. Despite showing promises, there is also some controversy in a therapy setting. Some research seems to suggest that traditional therapy results in better performance by the patient [19]. Whereas others get better results with robot-assisted therapy [18]. It is argued that therapists are better at feedback adaption than haptic devices and that this is vital for relearning movements. In general there is a lack of study on healthy subjects to be able to draw valid conclusions on its effectiveness.

Haptic feedback shows promise in the learning of simple motor tasks (such as drawing) compared to more complex tasks (such as rowing). However, there is a need for a more systematic evaluation of haptic feedback compared to visual or auditory feedback for more valid conclusions. Also more research into multimodal feedback is recommended and shows promise.

Chapter 3
Prior Work

HAPTIC feedback in combination with touchscreen input is a widely researched topic. Especially pen-like haptic interfaces have been suggested and researched in many different ways. Each suggestion focusing on different modalities, although tactile has gained significantly more attention, and priority. Three interesting subareas will be discussed. First of all, general haptic interfaces; the focus shall be on stylus like interfaces. Secondly, a subset of this the interactive sketching interfaces. Finally, another subset of haptic interfaces, the interfaces that use magnetic forces in order to provide haptic feedback. Table 3.1 summarizes the prior work investigated. It is important to notice that this chapter will discuss the work related to the encompassing vision of the project. Whereas the work and theoretical background regarding this thesis specifically will be discussed later.

Tool	**Purpose**	**Pen/Surface**	**Actuator**	**Ref.**
TouchEngine	Provide tactility to GUI buttons.	Surface	Vibration Motors	[45]
Haptic Pen	Enhance interaction with GUIs.	Pen	Solenoid	[31]
ImpAct	Give perception of depth.	Pen	DC Motor	[64]
UbiPen II	Braille, button and texture rendering.	Pen	Impact Generator	[26]
Ubi-Pen Series	Precise Control of GUI elements.	Pen	Impact Generator	[27]
PBHVE	3D shape design.	Surface	DC Motor	[20]
dePENd*	Improve sketching.	Surface	Neodymium Magnet	[65]
Digital Rubbing	Frottage from a digital image.	Pen & Surface	Solenoid	[25]
HHA	Relearn handwriting after a stroke.	Surface	Phantom Omni	[40]
Actuated Workbench	Move physical objects on a surface.	Surface	Electromagnets	[44]
Proactive Desk II	Haptic feedback for on desk objects.	Surface	2D Linear Induction Motor	[66]
Fingerflux	Near-surface haptic feedback.	Surface	Electromagnets	[63]

Table 3.1: Overview of prior work. The tables shows: the name of the system, its purpose, whether the main actuating mechanism is in the pen or in the surface that is touch and, if applicable, what type of magnet the system utilizes. *This work closely resembles our own, however there are some key differences.

3.1 Haptic Interfaces for Stylus-type Interaction

Poupyrev et al. [45] propose to implement piezoelectric actuators in the corners of a monitor. The actuators bend the glass outward on a electrical signal and the user can feel the vibration through a pen. The authors added tactile feedback to GUI interaction (such as buttons and sliders), data perception and active pen inputs. Especially the later is interesting, since they focused on drawing and sketching operations. In this Poupyrev et al. found that the users appreciated tactile feedback, although it was not regarded as natural or paper-like. All users also reported that the feedback strength should change in proportion to the pen pressure.

The Haptic Pen (Figure 3.1), proposed by Lee et al. [31], has the actuator in the shaft of the

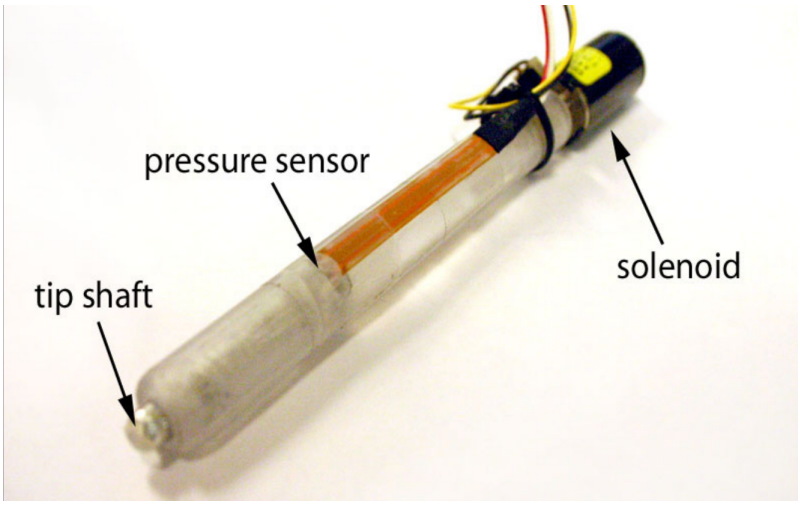


Figure 3.1: The Haptic Pen by Lee et al. [31].

pen rather than in the monitor. They argue for this decision due to their focus on collaborative displays. By adding the actuators to the pen they are able to give individual tactile feedback to multiple users. The actuator, a solenoid mounted coaxial at the place where a eraser normally would be placed, is used to generate force profiles. This enables Lee et al. to generate, for example, ”clicks” with different strengths. In the research the authors focused on interaction with buttons and do not incorporate free-form interactions, such as sketching.

A different approach is taken by Withana et al. [64], they try to give the illusion of depth on a screen. They do this with a stylus that is capable of changing in length. The stylus plus its virtual counter-part will be combined be a constant. This is illustrated in Fig. 3.3. They achieve this by combining a linear encoder, DC motor and Accelerometer-Magnetometer into the stylus. Although their original goal was to let the user perceive depth, Withana et al. argue that it can also be used to create 3D drawings. However, this is not empirically validated and just a suggestion.

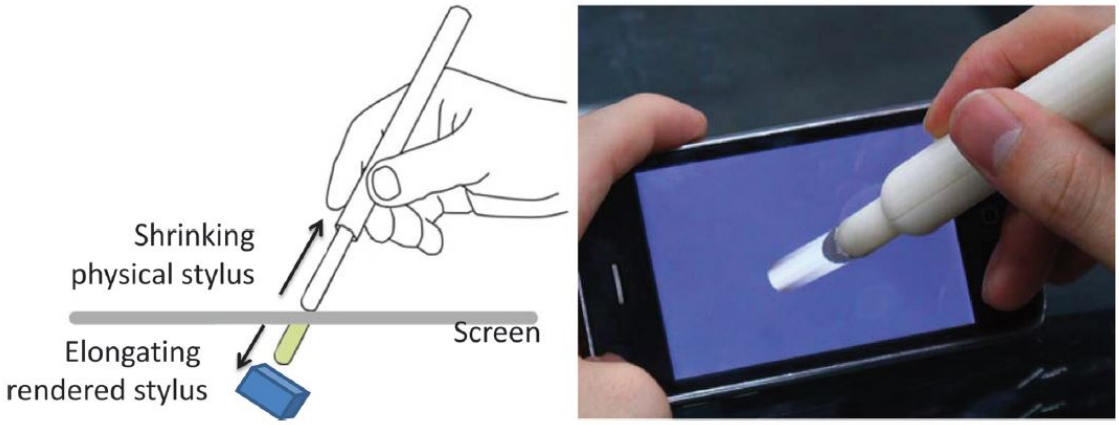


Figure 3.2: Illustration of the length change of the stylus in the ImpAct project [64].

Haptic interfaces also increase the accessibility of GUIs. This is nicely illustrated by the research conducted by Kyung et al. [26]. In their UbiPen II project they implemented a 3x3 pin array into a stylus like object. The pins has a linear travel of 1.0mm, with enough force to deform skin and be noticeable. Due to this Kyung et al. are capable of translating braille, buttons and texture into tactile feedback. However, it does not enable for tactile input. They empirically validated their proposal and found in all three areas (braille, buttons and texture) an improvement in time on task spend as well as an increased accuracy.

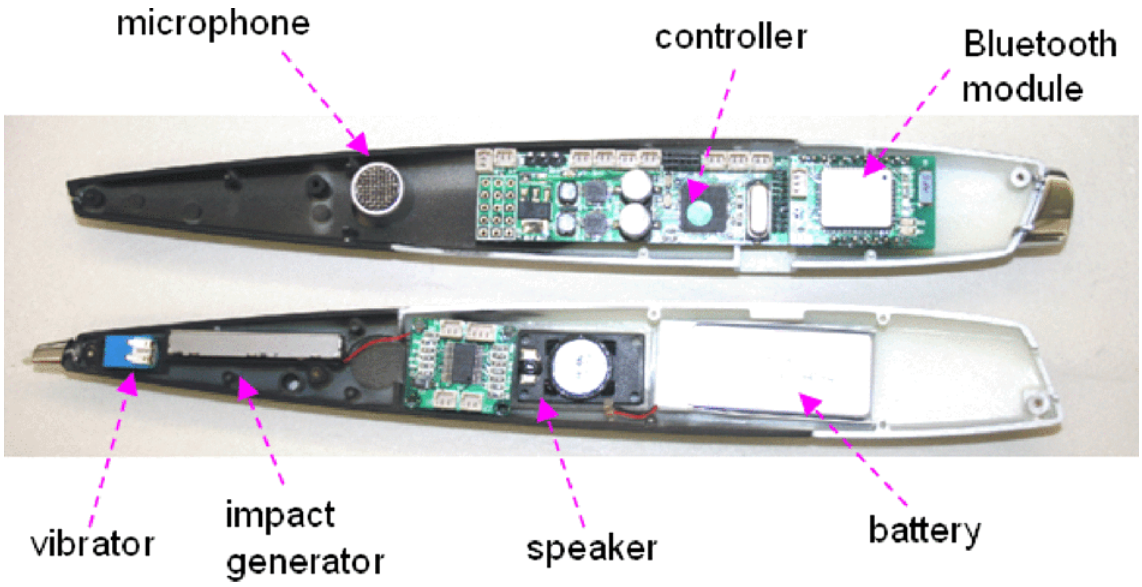


Figure 3.3: Ubi-Pen by Kyung et al. [27].

Kyung et al. continued this project and developed a new iteration in the Ubi-Pen Series [27] (Fig. 3.3). Besides improving the hardware (e.g. making the pen wireless), they also focused on more every-day tasks. Such as text-selection and navigating a menu. Kyung et al. validated their proposal with a survey (n=162) and qualitatively on conferences (n $\approx$ 1000). Although only 30 to 40% of the people expressed that they feel that they could manipulate objects faster with haptic feedback, more than 75% of all people felt that haptic feedback increased comfort and precision.

3.2 Interactive Writing, Sketching and Design Interfaces

Iwata [20] has developed a 6 degree-of-freedom force reflective master manipulator with a pen-shaped grip consisting out of two 3 degree-of-freedom manipulators; one on each end of the pen. This setup is shown in Figure 3.4. All joints in the setup are equipped with a DC motor and potentiometer. The purpose of the latter is the measure the position of the joint, the former is to apply tactile feedback to the user. The goal of this research was to develop a light-weight device for 3D digital shape manipulation and to evaluate its technical feasibility. However, it has not

been validated in a user study and is quite a cumbersome device that seem to have impact on the available movements and positions of the pen.

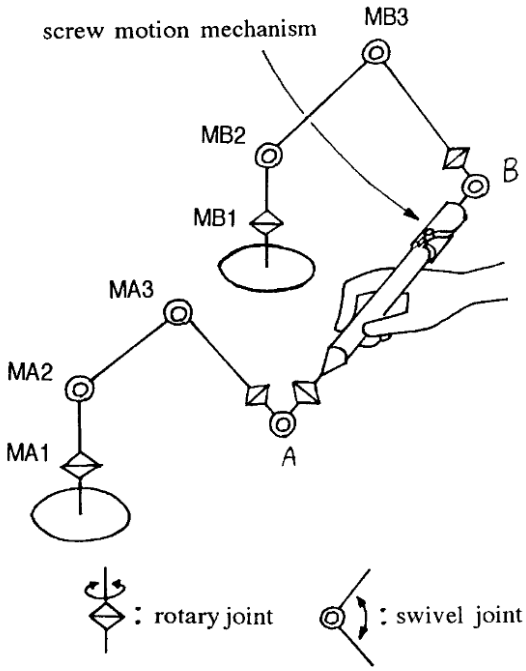


Figure 3.4: Mechanical overview of Iwata [20] their proposal.

dePENd (Fig. 3.5) by Yamaoka and Kakehi [65] is perhaps the closest prior research to our system. They proposed a a permanent Neodymium magnet on a two-axis setup, in order to control the pen of a user. They make use of the ferromagnetic feature of the metal tip of a regular ballpoint pen to attract it to the magnet. They do not draw conclusion on the accuracy of the system from a user perspective and instead state that it is less accurate than a regular plotter. The key difference between their proposal and our work lays within the magnet and its implications. By using a neodymium magnet, they are unable to alter the strength during the interaction. Due to this they "force", rather than guide, the user to follow a predefined path. Similarly, Yamaoka and Kakehi use a infrared sensor to determine the position of the pen. Because of this they are unable to determine the pressure applied by the user and react to this.

A project that provides a lot of lessons to be learned for our proposal is Digital Rubbing by Kim et al. [25]. They tried to allow for a rubbing draw technique from a digital image rather than a textured surface (Fig. 3.7). When the pen is over a surface that is part over the drawing a solenoid presses a pencil to the paper. Their location measuring method, a tablet with touch input, is similar to ours and brings to light some technical difficulties. Most important, that this method provides a significant delay between measurement and actuation. Hence, a prediction method was necessary. They used a basic method, since rubbing is a relatively predictable movement.

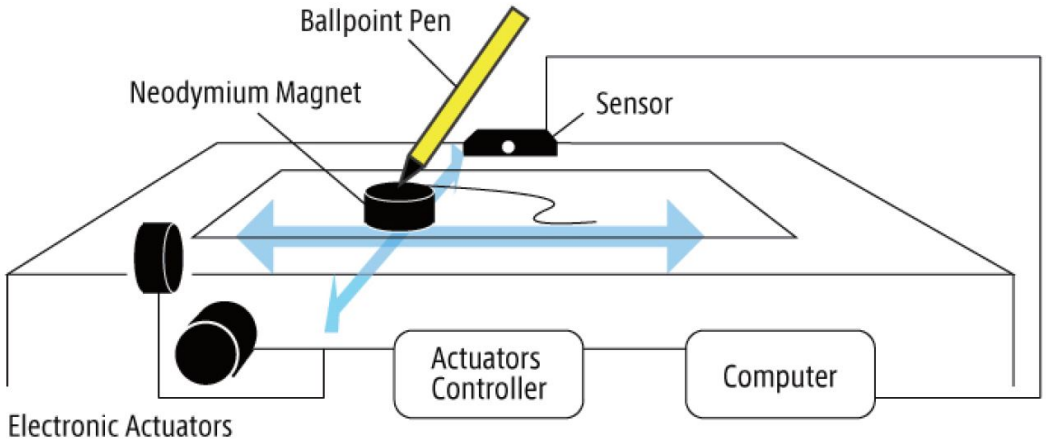


Figure 3.5: Schematic overview of the dePENd system [65].

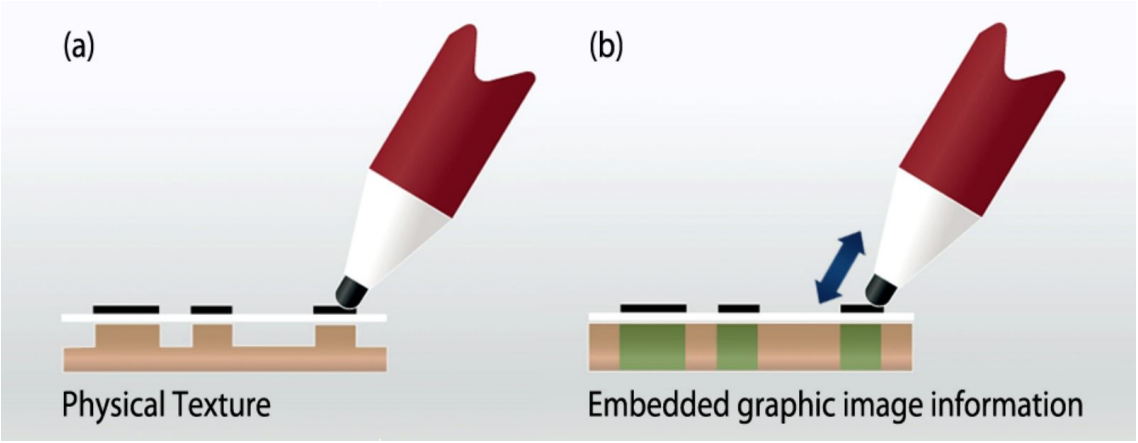


Figure 3.6: (a) Traditional Frottage (b) Digital Rubbing [25].

The Haptic Handwriting Aid (HHA) by Mullins et al. [40] is a project that has a similar end-goal compared to the proposed project. Namely, to (re)learn handwriting by means of haptic feedback. They use commercially available equipped (the Phantom Omni) which provides 3-DoF. However, despite haptic feedback, the user writes in air and thereby loses the feeling of writing on paper. Similar, the user uses a stylus not resembling a pen or pencil. Next to this they have not empirically validated the results.

3.3 Interfaces using Magnetic Forces

One of the more well-known tabletop interfaces using magnetic forces is the Actuated Workbench by Pangaro et al. [44]. They use an array of electromagnets to move a puck across a table surface. The Actuated Workbench was created as a response to interactive tabletops, in which objects

are tracked and response to users' physical input with a graphical output. However, if a user misplaces an object the tabletop is not able to correct this. Actuated Workbench is capable of doing this. An array has a significant weakness over our proposed printer-like setup, namely that creating smooth movements are often impossible to achieve. Interesting, however, is their modeling of the forces on the puck. This will be discussed in more detail in Chapter 10.

Figure 3.7: Actuated workbench, showing the electromagnet array, by Pangaro et al. [44].

A very similar prototype has been build by Yoshida et al. [66]. They used a 2-degrees-of-freedom linear induction motor to provide haptic feedback on multiple object in a desk-like scenario. The actuator, in practice, was a 2D array of coils. The coils are powered with an alternating current, by the strength as well as relative phase determine the direction in which the object moves. It is very closely related to the principle of a magnetic train, in which different magnets are turned on after each other to let the object move forward. Since the magnetic field was relative small, Yoshida et al. were able to move multiple objects in a single setup. Their project was not validated with users and was meant as a technical proof-of-concept.

Another project the bears resembles to the work of Pangaro et al. is FingerFlux by Weiss et al. [63]. They work with an almost identical setup as Pangaro et al., however their aim is to provide near-surface haptic feedback to users. They achieve this by attaching a permanent magnet to the users' finger. With an electromagnet they are able to attract or repel the magnet and therefore the finger. They claim that the biggest advantage of their work is *a priori* feedback, due to the near surface nature of their setup they can provide feedback before the user actually touches the interface. They found that their setup drastically reduces drifting of the finger. However, as of yet, FingerFlux is unable to track the finger making the practical implications still a challenge.

Chapter 4
Conclusion

WE have discussed a variety of prior work, ranging from stylus-related haptics, drawing assisting tools and tabletop haptics; both in two and three dimensional space. These are definitely not all projects done in this area of research, however, the ones discussed are most notable and relevant. Other interesting projects include, but not limited to, TeslaTouch Bau et al. [4], RealPen Cho et al. [6] and MudPad Jansen et al. [21].

Valuable lessons can be drawn for this project. When taking a look at [44, 66, 63] it is evident that an array of electromagnets has the preference in haptics over a motorized setup, as proposed in [65]. Generally, it is argued to limit the amount of moving elements in order to increase robustness. However, using an array like setup has the disadvantage of decreasing accuracy of the magnetic field. The accuracy is of utmost importance in drawing and writing tasks alike. Therefore, this project will make use of the setup similar to dePENd [65].

Another valuable consideration is the delay between input and actuation, as encountered by Kim et al. [25]. Where they had a use-case of rubbing, our proposal is about drawing and writing. Both of these are less predictable movements. From Kim et al. we learn that we should have prediction algorithms, however they should be more advanced than they have used. We discussed the possibilities in more detail in Chapter 8.

dePENd [65] is very similar to the project proposed by us. With a rather similar goal. However, as discussed, there are some significant differences in terms of vision and execution. Especially the forcefully controlling of the input device is something we do not agree with. Due to them using an static magnet they have lesser control over the force. Theoretically they could control the force on the pen by controlling the distance between the magnet and then pen.. However, due to the quadratic nature of magnet strength versus distance this requires extreme speeds and accuracy. A millimeter too far and the force on the pen will be too little, a millimeter too close and the pen will snap to the magnet. Although in our project we will try to keep the distant a constant, being able to vary the force will allow us to control the strength more accurately.

4.0.1 Opportunities

Combining all projects, it is safe to say that our proposal is unique in its kind. Our approach of training muscle memory has not been done before. Nor has it been empirically validated that haptics can improve accuracy as well as learning-rate for fine-grain motor tasks. This provides us with a clear research space with ample opportunities.

Similarly, a moving electromagnet is unexampled. When taking specifically a look at actuation that happens outside of the pen, the actuators are most often DC motors providing resistance, or an array of electromagnets. The problem with the former is that it does not come close to the experience of writing, since there is always something elaborate attached to the stylus. The latter

falls short in terms of accuracy and precision. As a result, we are confident that our research will provide to be a valuable contribution to the research area of applied haptics.

Part II

Overview

This part will discuss our research question and the goals derived from the question. It will build on preceding research to form these. Next to this it will outline the process, this will help to understand the remainder of the thesis and the context of the subsequent chapters.

Chapter 5
Research Aims

IN the related work chapter, 3, we have seen that using haptic guidance is a promising direction for learning simple motor tasks. In prior work it was illustrated that there is only a single project that makes use of magnetic forces for learning simple motor tasks, however this project uses position control haptics and not guidance control.

The aim of this project is to design, implement and empirically evaluate a haptic guidance feedback device in order to rehabilitate and train a specific fine-grain, yet simple, hand motor tasks; drawing geometrical shapes. Other tasks, such as handwriting, may seem similar to drawing; however, they are intrinsically more complex. For instance, the letter *a* has different stroke-speeds at different curves of the letter. Hence, this project will solely focus on drawing geometrical shapes. As illustrated, this is relevant for several targets groups and is under-researched; with a lack in empirical evaluation. This project is mainly meant as a proof-of-principle.

This device would deliver the haptic guidance by means of an electromagnet mounted on a x- and y-axis so that it can move on a horizontal plane. The electromagnet attracts permanent magnets embedded in a stylus. Next to this we will use a pressure sensitive touch pad to determine the position of the pen.

However, due to time-constraints, *this thesis will investigate the necessary algorithms plus models for the control of the strength and position of the electromagnet with respect to the pen position as well as the contour.*

Combining all, my thesis will provide a contribution to the field in two distinct manners; conceptual and algorithmic. First of all, it provides a conceptual contribution because haptic guidance with the help of electromagnets has not been done before. The second contribution, algorithmic, happens on two fronts. The modeling of a single electromagnet in such as setup has not been done before and secondly, Model Predictive Contour Control has not been used for a haptic guidance device.

5.1 Research Goals

1. *Model an electromagnet with a high accuracy.*
   Modeling electromagnets is an ill-posed problem. The most state-of-the-art models make use of the finite element method; this is not fast enough for our prototype [37]. Instead we need to find a closed-loop method that approximates the actual strength with a reasonable accuracy. We need to be able to predict how an input would influence the perceived strength for some of the control strategies we investigate.

2. *Investigate Different Control Strategies*
   We have seen that current solutions use a PID control strategy. However, this seems far from

ideal for our scenario. Hence, we will investigate PIDs, and more advanced strategies, in a very simple scenario. We do this in order to make an informed choice regarding the final strategy used.

3. *Propose Suitable Hardware*
   The hardware has a large influence on the formalization of the selected control strategy. Hence, attention needs to be paid to its design before we can formalize a method. Despite that the hardware will not be fully built for this thesis, we should propose a design.

4. *Formalize, Implement and Evaluate the most Relevant Control Strategy.*
   The major contribution of this thesis lies in this goal. First of all we will mathematically define the chosen control strategy. Based on this definition we will implement the strategy. Furthermore, we will outline the process of the evaluation as well as conduct the evaluation.

Chapter 6
Outline

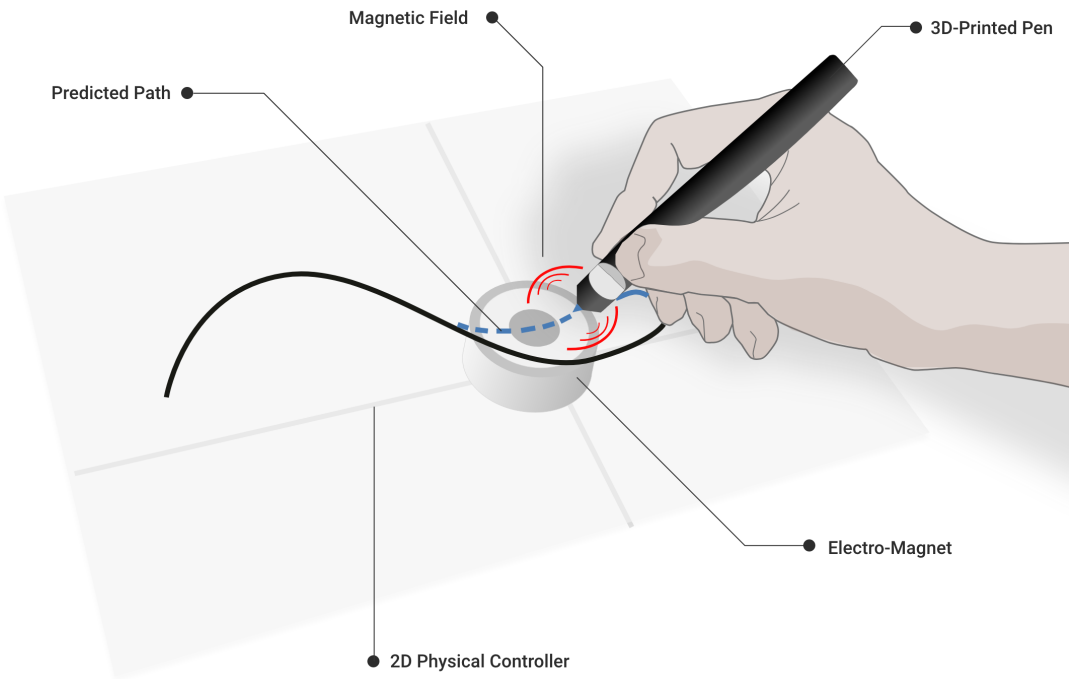


Figure 6.1: Overview of the implemented system.

The proposed system consists out of several different elements as shown in Figure 6.1. These have been investigated at different scales and resulted in a highly non-linear progress in which different elements were executed in parallel. For this reason it is valuable to explain the different parts of this thesis and how they will fit together. In Part III we will extent on two separate, yet vital, topics. First of all we formulate an analytical model for the electromagnet. This is necessary in most forms of process control. Secondly we will investigate different control strategies such as PID, MPC and MPCC. From this investigation we will conclude that the MPCC strategies will work best for our problem.

We continue in Part IV with our hardware design. In this chapter we will discuss the electronics as well as the physical system, such as the custom designed pen. In the same part we will also fully define the MPCC. For this we use the model of the electromagnet, the lessons learned from the MPCC investigation and the proposed hardware.

In Part V we will evaluate the MPCC as well as propose a user test to test the envisioned final system. We will do both a quantitative as well as a visual analysis of the results of the MPCC in a

simulated environment. We finalize this part with a discussion and conclusion.

Part III

Background

We will introduce an analytical model to approximate the strength of an electromagnet on the pen, taking into account distance and current. This is necessary for all control strategies to work. Next to this we will explore three different strategies. We will find that MPCCs are most suitable for the task at hand.

Chapter 7
Electromagnetism Framework

ELECTROMAGNETS are a specific type of magnet that creates a magnetic field based on electrical current. Electromagnets consist usually of a insulated wire wound into a coil. Often the coil is around a ferromagnetic core, which concentrates the magnetic flux and thereby increases the strength of the magnet [36]. Figure 7.1 shows a simple illustration of a coreless electromagnet.

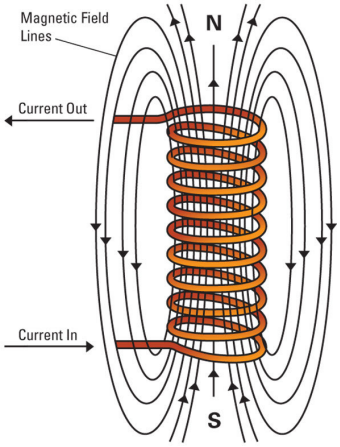


Figure 7.1: Illustration of an electromagnet [58].

The big advantage of an electromagnet over a permanent magnet is that the magnetic field can be controlled in terms of strength and direction. However, a downside is that electromagnets need a constant power supply. Another major downside is the Ohmic heating that occurs. The electromagnet is under constant current, due to the resistance in the wire heat is generated. Since the magnetic field is proportional to the number of windings N and the current I (Eq. 7.1), both can be used to minimize heat loss.

$$\text{Magnetic field} \propto N \cdot I \tag{7.1}$$

$$P = I^2 * R \tag{7.2}$$

The power dissipation is given by Equation 7.2. In which R is the resistance of the wire (which is proportional to N). From this equation combined with Equation 7.1. It can be determined that more windings (assuming the diameter does not change) and less current helps against heat loss, while keeping the magnetic force the same. However, the amount of windings is subject to spacial constraints.

Another problem with electromagnets (especially, those containing a ferromagnetic core) is the phenomenon of hysteresis. Magnetic hysteresis occurs when an magnetic field is applied to a ferromagnet (such as iron), the ferromagnet will remain magnetized to an extent. Even when there is no external magnetic field anymore.

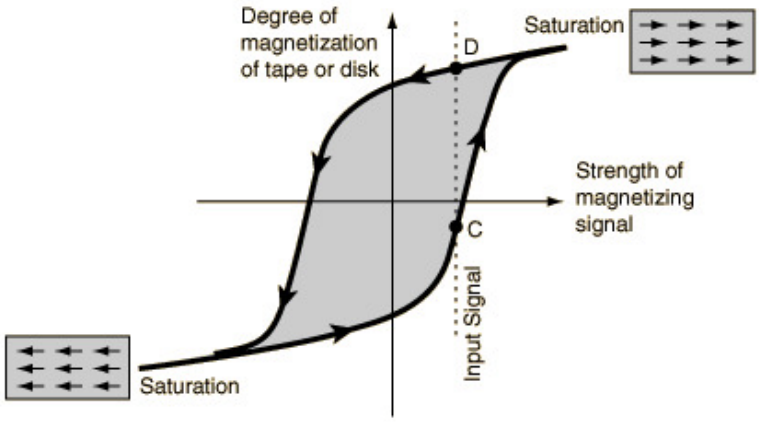


Figure 7.2: Magnetic Hysteresis. The magnetization and demagnetization of a ferromagnetic material [43].

Figure 7.2 shows a hysteresis graph. On the horizontal axis there is the strength of the magnetizing signal. In my case this is related to the current provided to the electromagnet. The vertical axis shows the magnetization of the core. If the core is not saturated the degree of magnetization is unknown (they gray area in the figure). This leads to the problem that the forces on the pen has a large uncertainty.

To over come this problem we will make use of pulse-width modulation (PWM) in which the electromagnet will receive enough current for saturation for a short amount of time and multiple times per second. The frequency of this will determine the perceived strength by the user. The perceived strength is linearly correlated with the frequency of PWM. This will also help to limit the power dissipation; since the current is not always on.

In order to be able to define the control algorithm we need to be able to predict the strength of the electromagnet on the pen based on the distance and the frequency of PWM. We will approximate this by formalizing the problem as a dipole-dipole magnetic interaction.

7.1 Formalization

Under the condition that two magnets are small and distant enough that their shape and size does not matter, both magnets can be modeled to be dipole and have a magnetic moment ($\mathbf{m_1}$ and $\mathbf{m_2}$, for the permanent and the electromagnet respectively). $\mathbf{m_1}$ is the magnetic moment of

the permanent magnet, which is based in the pen. This is calculated straightforward by:

$$m1 = M_0 * V \tag{7.3}$$

Where

M_0 = magnetic property of the material and V = Volume of the magnet

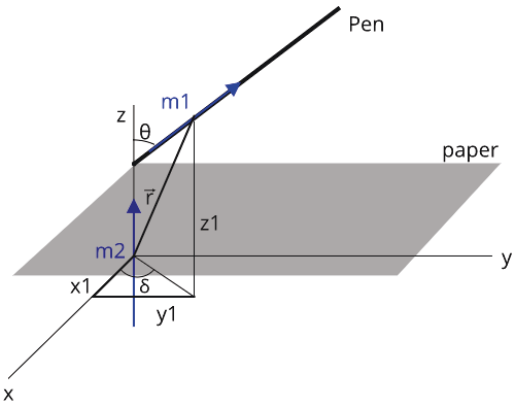


Figure 7.3: Variables in the electromagnet model.

However, since the pen is at an angle (Figure 7.3), the moment vector cannot be stated with full certainty and is depend on the angle at which the pen is hold. Next to this, there is no clear angles at which the pen is recommend to be hold compared to the paper. Our own measurements indicate that the pen is hold at an angle $\theta = 20^\circ$ from the z-axis towards the pen. The angle $\delta = 20^\circ$ is between the x-axis and the pen on the horizontal plane of the paper. However, this is likely to differ per person. Taking the angles into account results in the magnetic moment vector:

$$\vec{\mathbf{m}}_1 = \begin{bmatrix} m_1 \sin\theta \cos\delta \\ m_1 \sin\theta \sin\delta \\ m_1 \cos\theta \end{bmatrix} \tag{7.4}$$

Calculating $\mathbf{m_2}$, the magnetic moment of the electromagnet is relatively straightforward and can be formalized as follows, assuming the magnet does not saturate:

$$m_2 = N * I * \mu_r * \pi * R_m^2 \tag{7.5}$$

In which:

N = Number of windings

I = Electrical Current

μ_r = Relative permeability of the core

R_m = Mean radius of the coil

Since the electromagnet is in a fixed position, with its pole facing straight up, this results in the vector:

$$\vec{\mathbf{m}}_2 = \begin{bmatrix} 0 \\ 0 \\ m_2 \end{bmatrix} \tag{7.6}$$

The force of two dipole magnets on each other is defined as:

$$\mathbf{F}_{21}(\vec{\mathbf{r}}, \vec{\mathbf{m}}_1, \vec{\mathbf{m}}_2) = \frac{3\mu_0}{4\pi r^5} \left[\vec{\mathbf{m}}_2(\vec{\mathbf{m}}_1 \cdot \vec{\mathbf{r}}) + \vec{\mathbf{m}}_1(\vec{\mathbf{m}}_2 \cdot \vec{\mathbf{r}}) + \vec{\mathbf{r}}(\vec{\mathbf{m}}_1 \cdot \vec{\mathbf{m}}_2) - \frac{5(\vec{\mathbf{m}}_1 \cdot \vec{\mathbf{r}})(\vec{\mathbf{m}}_2 \cdot \vec{\mathbf{r}})}{r^2} \vec{\mathbf{r}} \right] \tag{7.7}$$

in which $\vec{\mathbf{r}}$ is the unit vector, point from magnet 1 to magnet 2. μ_0 is the relative permeability of air. r is the euclidean distance between those vectors:

$$\mathbf{r} = \begin{bmatrix} x_1 - x_2 \\ y_1 - y_2 \\ z_1 - z_2 \end{bmatrix} \tag{7.8}$$

$$r = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \tag{7.9}$$

Since we set m_2 as the origin we can simplify it to:

$$r = \sqrt{x_1^2 + y_1^2 + z_1^2} \tag{7.10}$$

Note that $\vec{\mathbf{r}}$ can be expressed as:

$$\vec{\mathbf{r}} = r\hat{\mathbf{r}} \Longrightarrow \hat{\mathbf{r}} = \begin{bmatrix} \frac{x_1}{r} \\ \frac{y_1}{r} \\ \frac{z_1}{r} \end{bmatrix} \tag{7.11}$$

However, it is known that z_1 is a constant. That is, the vertical distance between the electromagnet and then pen is always the same, we will introduce a dimensionless coordinate system, which is normalized by z_1. We will continue to work in this new space.

$$\begin{aligned} u &= x_1/z_1 \\ v &= y_1/z_1 \\ q &= r/z_1 \end{aligned} \tag{7.12}$$

A similar technique, as in Eq. 7.11, can be employed for the dipole moments $\vec{\mathbf{m}}_1$ and $\vec{\mathbf{m}}_2$:

$$\begin{aligned}\vec{\mathbf{m}}_1 &= m_1\hat{\mathbf{m}}_1 \Longrightarrow \hat{\mathbf{m}}_1 = \begin{bmatrix}\sin\theta\cos\delta\\ \sin\theta\sin\delta\\ \cos\theta\end{bmatrix}\\ \vec{\mathbf{m}}_2 &= m_2\hat{\mathbf{z}} \Longrightarrow \hat{\mathbf{z}} = \hat{\mathbf{m}}_{\mathbf{2}} = \begin{bmatrix}0\\0\\1\end{bmatrix}\end{aligned} \tag{7.13}$$

As well as for the coordinates:

$$\vec{\mathbf{r}} = qz_1\hat{\mathbf{r}} \Longrightarrow \hat{\mathbf{r}} = \begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix} \tag{7.14}$$

Let us now rewrite the original equation 7.7 in terms of the unit vectors $\hat{\mathbf{m}}_{\mathbf{1}}$, $\hat{\mathbf{m}}_{\mathbf{2}}$ and $\hat{\mathbf{r}}$:

$$\mathbf{F}_{21}(\vec{\mathbf{r}},\vec{\mathbf{m}}_1,\vec{\mathbf{m}}_2) = \frac{3\mu_0 m_1 m_2}{4\pi z_1^4}\frac{1}{q^4}\left[(\hat{\mathbf{m}}_1\cdot\hat{\mathbf{r}})\hat{\mathbf{m}}_2 + (\hat{\mathbf{m}}_2\cdot\hat{\mathbf{r}})\hat{\mathbf{m}}_1 + (\hat{\mathbf{m}}_1\cdot\hat{\mathbf{m}}_2)\hat{\mathbf{r}} - 5(\hat{\mathbf{m}}_1\cdot\hat{\mathbf{r}})(\hat{\mathbf{m}}_2\cdot\hat{\mathbf{r}})\hat{\mathbf{r}}\right] \tag{7.15}$$

We will define the f_e and $\vec{f}$ as follows:

$$\begin{aligned}f_e &= \frac{3\mu_0 m_1 m_2}{4\pi z_1^4}\\ \vec{f} &= \frac{1}{q^4}\left[(\hat{\mathbf{m}}_1\cdot\hat{\mathbf{r}})\hat{\mathbf{m}}_2 + (\hat{\mathbf{m}}_2\cdot\hat{\mathbf{r}})\hat{\mathbf{m}}_1 + (\hat{\mathbf{m}}_1\cdot\hat{\mathbf{m}}_2)\hat{\mathbf{r}} - 5(\hat{\mathbf{m}}_1\cdot\hat{\mathbf{r}})(\hat{\mathbf{m}}_2\cdot\hat{\mathbf{r}})\hat{\mathbf{r}}\right]\end{aligned} \tag{7.16}$$

So that:

$$\mathbf{F}_{21}(\vec{\mathbf{r}},\vec{\mathbf{m}}_1,\vec{\mathbf{m}}_2) = f_e\vec{f} \tag{7.17}$$

Notice how f_e is a constant depending on the characteristics (m_1, m_2 and z_1) of the system and $\vec{f}$ is a function of u, v, θ and δ:

$$\vec{f} = \begin{bmatrix}f_x(u,v,\theta,\delta)\\ f_y(u,v,\theta,\delta)\\ f_z(u,v,\theta,\delta)\end{bmatrix} \tag{7.18}$$

Resulting in:

$$\begin{aligned}\vec{f}(u,v,\theta,\delta) = \frac{1}{q^4}[&(\begin{bmatrix}\sin\theta\cos\delta\\ \sin\theta\sin\delta\\ \cos\theta\end{bmatrix}\cdot\begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix})\begin{bmatrix}0\\0\\1\end{bmatrix} + (\begin{bmatrix}0\\0\\1\end{bmatrix}\cdot\begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix})\begin{bmatrix}\sin\theta\cos\delta\\ \sin\theta\sin\delta\\ \cos\theta\end{bmatrix}\\ &+(\begin{bmatrix}\sin\theta\cos\delta\\ \sin\theta\sin\delta\\ \cos\theta\end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix})\begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix} - 5(\begin{bmatrix}\sin\theta\cos\delta\\ \sin\theta\sin\delta\\ \cos\theta\end{bmatrix}\cdot\begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix})(\begin{bmatrix}0\\0\\1\end{bmatrix}\cdot\begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix})\begin{bmatrix}u/q\\ v/q\\ 1/q\end{bmatrix}]\end{aligned} \tag{7.19}$$

Which in turn can be rewritten to:

$$\vec{f}(u,v,\theta,\delta) = \frac{1}{q^5}\begin{bmatrix} \sin\theta\cos\delta + u\cos\theta \\ \sin\theta\sin\delta + v\cos\theta \\ u\sin\theta\cos\delta + v\sin\theta\sin\delta + 3\cos\theta \end{bmatrix} - \frac{5(u\sin\theta\cos\delta + v\sin\theta\sin\delta + \cos\theta)}{q^7}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{7.20}$$

From this follows that:

$$\begin{aligned} f_x =& \frac{u(q^2-5)}{q^7}\left[\cos\theta + \left[\frac{(q^2-5u^2)\cos\delta - 5uv\sin\delta}{u(q^2-5)}\right]\sin\theta\right] \\ f_y =& \frac{v(q^2-5)}{q^7}\left[\cos\theta + \left[\frac{(q^2-5v^2)\sin\delta - 5uv\sin\delta}{v(q^2-5)}\right]\sin\theta\right] \\ f_z =& \frac{3q^2-5}{q^7}\left[\cos\theta + \left[\frac{(q^2-5)(u\cos\delta + v\sin\delta)}{3q^2-5}\right]\sin\theta\right] \end{aligned} \tag{7.21}$$

We continue with defining the in-plane force F_φ:

$$F_\varphi = f_e\sqrt{(f_x)^2 + (f_y)^2} \tag{7.22}$$

All that is left is to introduce a term α, which corresponds the the time-interval of the pulse-width modulation (PWM) so that $0 \leq \alpha \leq 1$. In which 0 is completely off and 1 is completely on. This linearly controls the force exerted by the magnet. Hence, the final formula to calculate the in-plane force with PWM becomes:

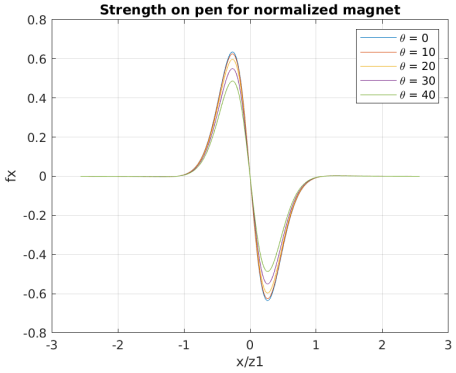
$$F_{EM\to P} = \alpha F_\varphi = \alpha f_e\sqrt{(f_x)^2 + (f_y)^2} \tag{7.23}$$

In order to investigate the influence of the angle θ we will assume that $v = 0$ and $\delta = 0$. As $v = 0 \Rightarrow q^2 = u^2 + 1$ and $f_y = 0$. As $\delta = 0 \Rightarrow \cos\delta = 1$ and $\sin\delta = 0$. Hence Equation 7.21 can be written as:

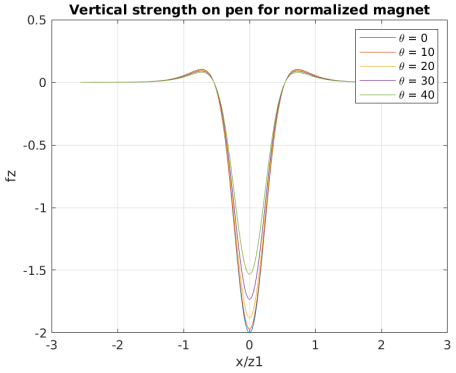
$$\begin{aligned} f_x =& \frac{u(u^2-4)}{(u^2+1)^{7/2}}\left[\cos\theta + \frac{(-4u^2+1)}{u(q^2-5)}\sin\theta\right] \\ f_z =& \frac{3u^2-2}{(u^2+1)^{7/2}}\left[\cos\theta + \frac{(u^2-4)u}{3u^2-2}\sin\theta\right] \end{aligned} \tag{7.24}$$

This allows us to plot the influence of different θ's on f_x and f_y over the normalized distance u. This can be seen in the figures 7.4a and 7.4b.

From this can be concluded that angle θ has a slight but significant influence on the force exerted on the pen. Next to this adding angle θ will, presumably, allow us to differentiate between left- and right-handed people. However, it is unfeasible to measure and compensate for θ in our current hardware setup. Hence, this will by a constant variable that can be measured per user.



(a) the force f_x of m_1 on m_2 compared for different values of u and θ.



(b) the force f_z of m_1 on m_2 compared for different values of u and θ. It is a negative, because the force of m_2 on m_1 is larger than vice versa for some values, resulting in a negative force from the perspective of m_1

Figure 7.4: Calculated forces on the pen given the distance and different angles (Electromagnet is fully on)

Low-fidelity in-house experiments have shown that the assumption that θ is constant is relatively save to make.

7.2 Evaluation

In order to validate our theoretical framework we will compare the result gotten from Equation 7.21 with the result from a Finite Element Analysis (FEA) done with the help of COMSOL.[1] For this analysis we will make the following assumptions: 1) $\theta = 0$, 2) $\delta = 0$ and 3) $r = z_1$. This assumptions will ensure that we only validate the f_z force. This will give us an indication on the quality of the framework laid-out above.

We do the analyses in different stages. Our model originated from a dipole-dipole interaction model as an approximation of an electromagnet-dipole model. Hence, we will first compare our results with a FEA of the former in a variety of cases. The selection of permanent magnets has been made based on an exploration in our setup on which magnets would be sufficient in strength. From this first analysis we will continue in exploring different cases for electromagnets (with and without core). The goal of this is twofold. First of all, the analysis the generalizability of our model. Secondly, to make an informed choice on our choice of electromagnet for our hardware. In all cases the magnet m_1 inside the pen consists out of two ring magnets. So that:

[1] `https://www.comsol.com/` - Multiphysics Modeling Sofware

Table 7.1: Properties of magnet m_1. Where OD is the outer diameter and ID is the inner diameter of the ring magnet. The 2 and 4 in the volume calculation are based on the fact that we use two permanent magnets in the pen. Hence the volume is twice the amount of a single magnet.

Two Ring Magnets	Values	Units
μ_0 =	1.26E-06	[H / m]
Br =	1.3	$[T = HA/m^2]$
OD =	0.01	[m]
ID =	0.004	[m]
h =	0.005	[m]
V = $\pi(OD^2 - ID^2)(2h)/4$ =	6.60E-07	$[m^3]$
m1 =	$Br * V/\mu_0$	
m1 =	0.683	$[Am^2]$

7.2.1 Permanent Magnet - Cube

For the first scenario we defined m_2 as two cube magnets. The properties of the magnet can be seen in Table 7.2. Figure 7.5 depicts the interaction between the two permanent magnets. In Table 7.3 the resulting f_e is shown.

Table 7.2: Properties of magnet m_2, where m_2 are two cube magnets with edges of 5mm. The 2 in the Volume calculation, is because we use two cube magnets to approximate the strength of the final electromagnet better.

Cube magnet 5 mm	Value	Unit
μ_0 =	1.26E-06	[H / m]
Br =	1.3	$[T = HA/m^2]$
L =	5.00E-03	[m]
V_{m_2} = (2L)*L*L =	2.50E-07	$[m^3]$
m2 =	Br * V / μ_0	
m2 =	0.259	$[Am^2]$

Table 7.3: f_e based on m_1 and m_2.

f_e	Value	Unit
μ_0 =	1.26E-06	[H / m]
m1 =	0.683	$[Am_2]$
m2 =	2.59E-01	$[Am_2]$
h_m1 =	0.010	[m]
h_m2 =	0.010	[m]
h_gap =	0.022	[m]
z1 =	0.032	[m]
f_e =	$3\mu_0 m_1 m_2/(4piz_1 1^4 4)$	
f_e =	0.051	[N]

From Equation 7.24 it can be determined that $f_z = 2f_e$ considering that $u = 0$. The sign does not matter, since it is the same force in the opposite direction. Hence, $f_z = 2 * 0.051 = 101[mN]$. The COMSOL analysis gives us $F1 = -106.82$ and $F2 = 86.89$. F1 and F2 were expected to have the same value but opposite sign, that this did not happen is due to numerical errors in the model.

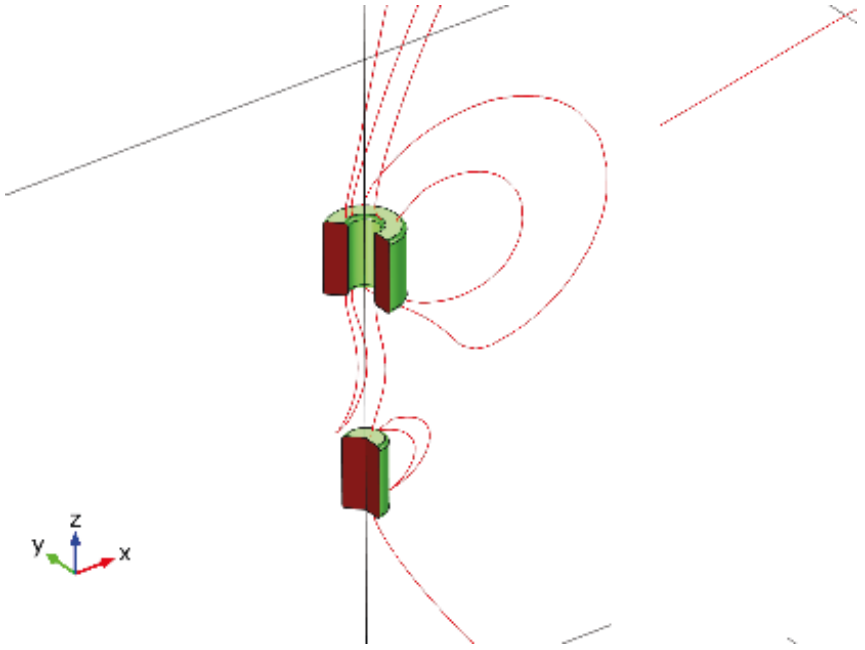


Figure 7.5: Schematic of cube magnet interacting with ring magnet.

This can be further improved. Despite this, our calculated f_z is in between F1 and F2 showing promise for further analysis and show that our framework is performing as expected.

7.2.2 Electromagnet - Coreless

The next step is the simulate an electromagnet without core and compare the results to the permanent magnet results. This will give us indication on the size and strength of our electromagnet.

Table 7.4: Properties of an electromagnet without core.

Coil with core	Value	Unit
μ_r =	1.00	
Rm =	0.0175	$[m]$
N =	900	
I =	0.46	$[A]$
h =	0.020	$[m]$
m2 =	$\mu_r N I \pi R m^2$	
m2 =	0.398	$[Am^2]$

We calculated an $f_e = 70.5[mN]$ from our analytical model. This is in sheer contrast with the predicted $F1 = -49.990$ and $F2 = 48.491$. Our model performs under par for a coreless electromagnet.

Table 7.5: f_e for a coreless electromagnet based on m_1 and m_2.

f_e	Value	Unit
μ_0 =	1.26E-06	[H / m]
m1 =	0.683	$[Am_2]$
m2 =	0.398	$[Am_2]$
h_m1 =	0.010	[m]
h_m2 =	0.020	[m]
h_gap =	0.024	[m]
z1 =	0.039	[m]
f_e =	$3\mu_0 m_1 m_2/(4piz_1 1^4 4)$	
f_e =	0.035	[N]

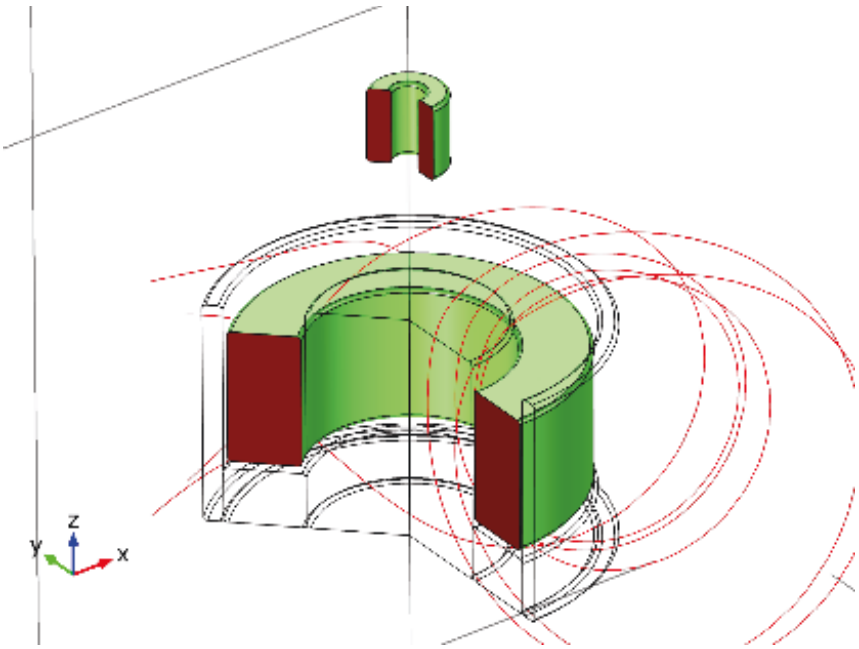


Figure 7.6: Schematic of electromagnet without core interacting with ring magnet.

7.2.3 Electromagnet - Core

We have seen that our model performs quite poorly on a coreless electromagnet. However, we will show that by adding a core to our electromagnet the predictions of our theoretical framework will increase in accuracy.

The FEA results are -255.630 and 271.630 $[mN]$ for F1 and F2 respectively. This in correspondence of our calculated value of $f_z = 264.4$. By adding a core to the electromagnet we find a significant increase the accuracy of our model.

Table 7.6: Properties of an electromagnet with core.

Coil with core	Value	Unit
μ_r =	3.75	
Rm =	0.0175	$[m]$
N =	900	
I =	0.46	$[A]$
h =	0.020	$[m]$
m2 =	$\mu_r N I \pi R m^2$	
m2 =	1.494	$[Am^2]$

Table 7.7: f_e for an electromagnet with core based on m_1 and m_2.

f_e	Value	Unit
μ_0 =	1.26E-06	[H / m]
m1 =	0.683	$[Am_2]$
m2 =	1.494	$[Am_2]$
h_m1 =	0.010	[m]
h_m2 =	0.010	[m]
h_gap =	0.024	[m]
z1 =	0.039	[m]
f_e =	$3\mu_0 m_1 m_2 / (4piz_1 1^4 4)$	
f_e =	0.132	[N]

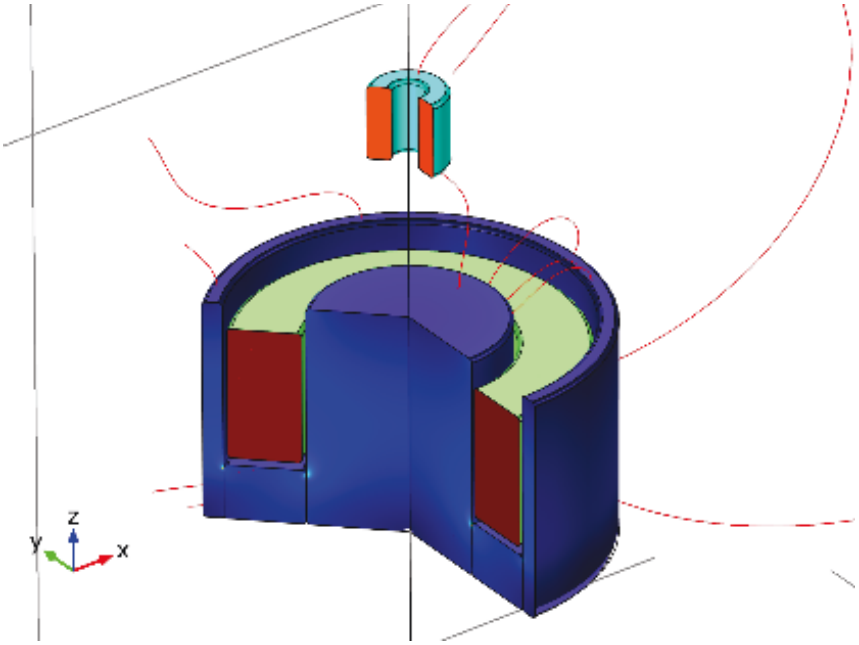


Figure 7.7: Schematic of electromagnet with core interacting with ring magnet.

7.3 Conclusion

In this chapter we proposed a analytical model in order to calculate the force the electromagnet has on the permanent magnets in the pen. We have shown that the angle at which the pen is

held impacts this force to such an extent that it should be taken into account. Finally we have validated our model with the help of a Finite Element Analysis. From which can be concluded that our analytical model is sufficiently accurate. The majority of the existing inaccuracy can be contributed to the assumptions we stated in the beginning: "Under the condition that two magnets are small and distant enough that their shape and size does not matter." In our setup z_1 is small enough for this assumption to not hold completely true, next to this our electromagnet is relatively large. Despite, this inaccuracy our theoretical framework is accurate enough to be incorporated into the control model.

Chapter 8
Control Strategies

DUE to the inherent latency between sensor and actuator some form of predictive control is needed. If the actuator waits for the latest sensor data it will always lag behind the user. For this reason, some prediction is needed where the user will be in the future and from that the most optimal position and strength for the electromagnet needs to be computed. In its very essence it bears some resembles to the software controlling Computer Numerical Control (CNC) machines, such as a laser cutter or 3D printer. However, there are some key differences. First of all, in our use-case the *fastest* route along a trajectory is less relevant. It is not our goal to be as fast as possible. Secondly, our scenario is interesting, since there are two systems, the setup and the user, of which the latter is only partially observable. This causes an interesting interplay between the pen position, electromagnet position and how they influence each other. Despite these differences it is interesting to investigate current trajectory tracking approaches and analyze their applicability to our scenario. There is a large variety of trajectory tracking approaches available. In this section we will discuss a subset of the most common approaches as well as evaluate simple implementations of them, in order to make a well-founded decision regarding the strategy.

8.1 Proportional-Integral-Derivative Controller

8.1.1 Introduction

The Proportional-Integral-Derivative (PID) control is perhaps one of the most common control strategy currently used in CNC machines; specifically a subset of PID controllers; Cascaded PI(D) Controllers. The PID controller continuously calculates the error value $e(t)$ as the difference between the desired set point and a measured variable. Based on proportional (P), integral (I) and derivative (D) terms (hence the name) it computes a correction . A blockdiagram of a simple PID controller is shown in Figure 8.1. The proportional term P can be seen as the current error, the integral term I as the accumulated error in the past and the derivative term D as the expected error in the future.

Formally expressed the output $u(t)$ of a PID controller is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{8.1}$$

where $e(t) = r(t) - y(t)$. The factors K are gained by loop tuning. Tuning is a hard and ill-posed problem, with the most advanced methods still being patented. For an overview of tuning methods please refer to the extensive review by Cominos and Munro [8].

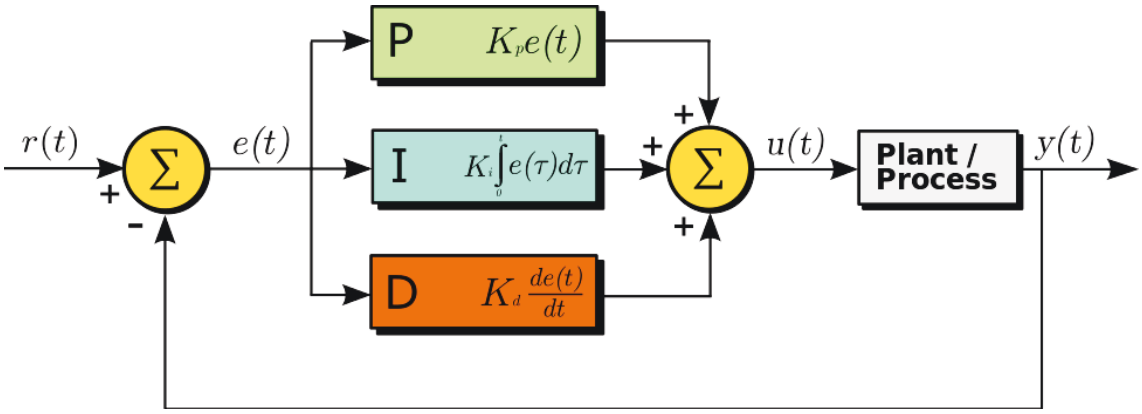


Figure 8.1: Blockdiagram of a PID controller in which $r(t)$ is the setpoint and $y(t)$ is the measured variable [59].

The biggest advantage of PID controllers is the general ease of implementation. However, compared to other control methods they perform relatively poorly. They are prone to overshooting and oscillating as well as are unable to model non-linear processes [3]. Next to this PID controllers cannot cope which changing variables, such as wear of mechanical systems [12]. Some of the problems can be overcome by alterations to the methods (such as cascading PID controllers), however compared to other methods the performance remains below par.

8.1.2 Implementation

The pseudocode in full can be seen in Alg. 1. It consists out of few steps, I will elaborate on some of them as well as the general principle. The input to the algorithm is the desired path. The parameters is a set of three weights for the PID controller, controlling the influence of each element. Then for each loop the actual position of the pen is measured. From the measurement the error with the desired position is calculated. The integral is the sum of all errors over all measurements and the derivative is the the current error minus the previous error. The suggested new position is the weighted sum of the three components.

Algorithm 1 Pseudocode for simple PID implementation

INPUT: desired position
PARAMETERS: weights
loop
actual position ← suggested position + noise
error ← desired position - actual position
integral += error
derivative = error - last error
last error ← error
suggested position ← w_p*error+w_i*integral+w_d*derivative
end loop

8.1.3 Simulation

For the simulation we had to define several elements in our implementation as well as make some assumptions. First of all, the weights were set to (0.9, 0.1, 0.0) for the P, I, D part respectively. Secondly, we assumed the actual electromagnet position was the suggested previous position along the x- and y-axis. The desired path was defined to be a sinusoid. It also should be noted that the parameters for the weights were optimized by hand, there are probably better weight distributions available.

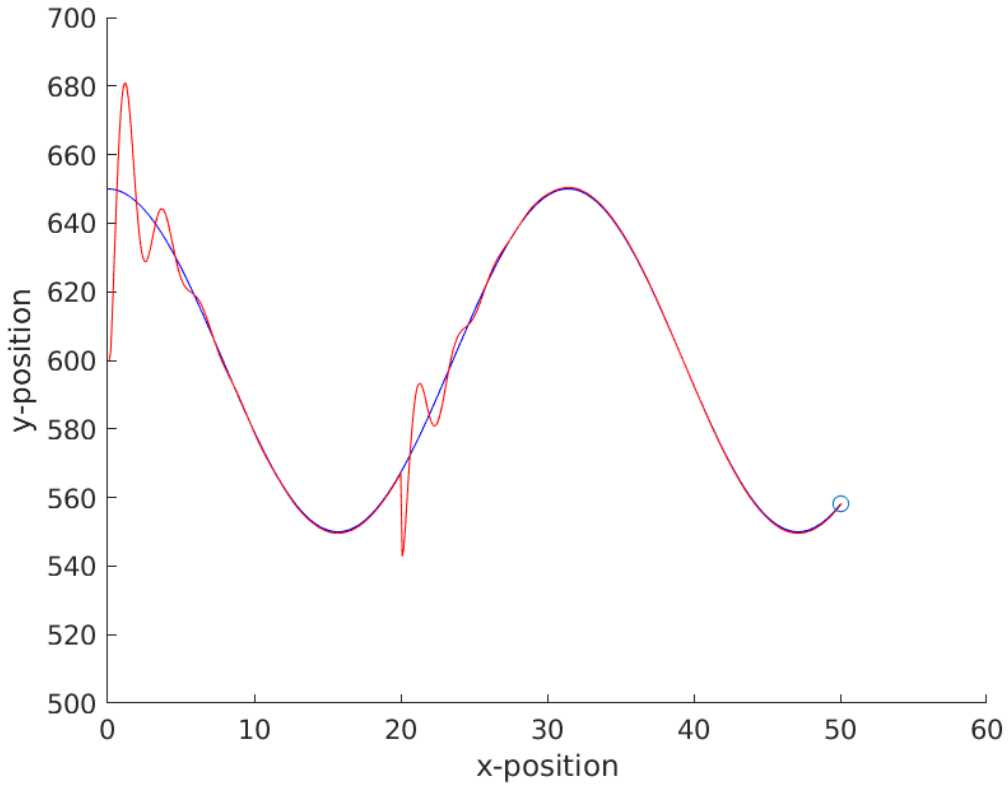


Figure 8.2: The result of the simulation of a PID Controller. Blue is desired path/position, red is path taken.

From the visualization in Figure 8.2 it can be concluded that the path of the electromotor would be extremely jerky in a PID setting. PIDs have inherently a nature of oscillating closer towards the desired path. In its essence they overshoot the desired path by a little less every time. However, smoothness is vital for our application area. Due to this it was decided to not continue developing PIDs any further and instead focus on MPC and MPCC solutions.

8.2 Model Predictive Control

8.2.1 Introduction

Model Predictive Control (MPC) is an advanced control strategy that is gaining large popularity over the last 25 years. Qin and Badgwell [46] calculated that the number of industrial machines using MPC doubled between 1994 and 1999. Morari and Lee [39] provide an, though slightly outdated, excellent analysis of the variety of MPC implementations as well as their (dis)advantages.

The principle behind MPC is to use a model of the system to predict the output of the system for i number of time-steps into the future. From this the control signal is optimized such that difference between the desired and the actual output signal is minimized over a sum of all time-steps by means of a cost function.

In CNC applications the cost function C for time k can be defined in many different ways, depending on the properties needed. Lam [28] defines one of the most basic MPC as:

$$C(k) = \sum_{i=1}^{H_p} ||x_{k+i} - x^d_{k+i}||^2_Q + \sum_{i=1}^{H_u} ||\Delta u_{k+i-1}||^2_R \tag{8.2}$$

where:

x_{k+i} is the predicted output of the system $k+i$, so that $x_{k+i} = f(x_{k+i-1})$

$f(x)$ is a model of the system.

x^d_{k+i} is the desired output of the system at time $k+i$

Δu_{k+i-1} is the predicted change in input

H_p and H_u are prediction and control horizons

Q, R are weight matrices.

$||e||^2_Q$ is used to denote the quadratic form $e^T Qe$

The number of time steps ahead used for prediction is known as the prediction horizon, H_p and the control horizon H_u specifies the number of optimized future control inputs. In practice it holds that $H_p = H_u$.

In its essence the cost function in Equation 8.2 tries to minimize two contradicting elements.The first element is the tracking error, the difference between the actual and the desired state. The second is the change in control input, when this change is lower the control signal will be smoothed. Q and R respectively control the relative importance of both.

The solution of the minimized $C(k)$ is a series of control inputs (u_{k+i}). The first element in this series, u_k is applied to the system. The optimization is executed again the next time-step. This is known as Fixed Receding Horizon. a second version of Model Predictive Control has a variable horizon rather than a fixed horizon. The difference between the two lays in the timestamps the model looks at for optimization. A fixed horizon looks at the n closest points (with n being a

positive integer) whereas a variable horizon divides the remaining path into n points. The latter allows to look longer ahead and predict more long-term. This will on one hand result in smoother motions, but also deteriorate the detail in the contour.

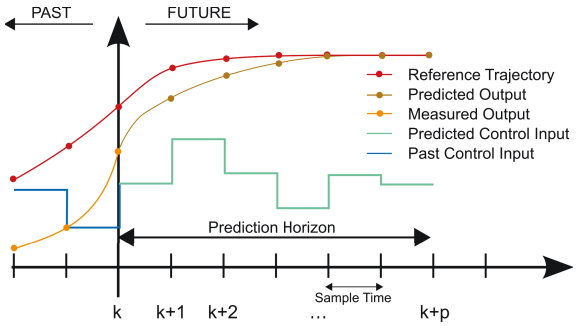


Figure 8.3: Example of in- and out-put as well as a reference trajectory of a MPC fixed receding horizon control system [61].

There are several advantages of using MPCs, for instance, that actuator constraints can be modeled explicitly [28]. Another advantage over PID controllers is that MPCs optimize over several time-steps into the future, rather than only one. This characteristic can help to overcome latency issues. However, there also disadvantages. It is relatively computational heavy, especially if the system model is nonlinear the optimization problem becomes a non-linear, non-convex problem; these are usually non-trivial to solve efficiently and sometimes local optima, rather than global, should be satisfactory. Next to to this MPCs use tracking error (the point at which the electromagnet had to be at a certain point in time) rather than contour error (the closest position on the contour), which is inherently less accurate.

8.2.2 Evaluation

The error function we will try to minimize is given as follows and is slightly adapted from 8.2 u^* is a list of actions to take to get the optimal result for the horizon length N:

$$u^* = \min_{u_i, x_i} \sum_{i=1}^{N} [x_{k+i} - x_{k+i}^{(d)}]^T Q [x_{k+i} - x_{k+i}^{(d)}] + \mathbf{u_{k+i}}^T R \mathbf{u_{k+i}} \tag{8.3}$$

subject to the following constrains:

$$\begin{aligned} x_0 &= \begin{bmatrix} 5 \\ 500 \end{bmatrix}, \\ x_{k+i} &= Ax_{k+i-1} + Bu, \\ \begin{bmatrix} -Inf \\ -Inf \end{bmatrix} &\leq x \leq \begin{bmatrix} Inf \\ Inf \end{bmatrix}, \\ \begin{bmatrix} -Inf \\ -Inf \end{bmatrix} &\leq u \leq \begin{bmatrix} Inf \\ Inf \end{bmatrix} \end{aligned} \tag{8.4}$$

With:

$$x = \begin{bmatrix} p_x \\ p_y \end{bmatrix}, u = \begin{bmatrix} \dot{p}_x \\ \dot{p}_y \end{bmatrix}, A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} w_r & 0 \\ 0 & w_r \end{bmatrix}, N = 20 \tag{8.5}$$

w_r was varied during the experiment to evaluate the different results. This is the smoothing factor. Hence, a higher w_r would mean less accurate path following, but smoother lines. The reference path, x_d, is defined as:

$$x_d = \begin{bmatrix} t \\ 50 * \cos(t/50) + 600 \end{bmatrix} \tag{8.6}$$

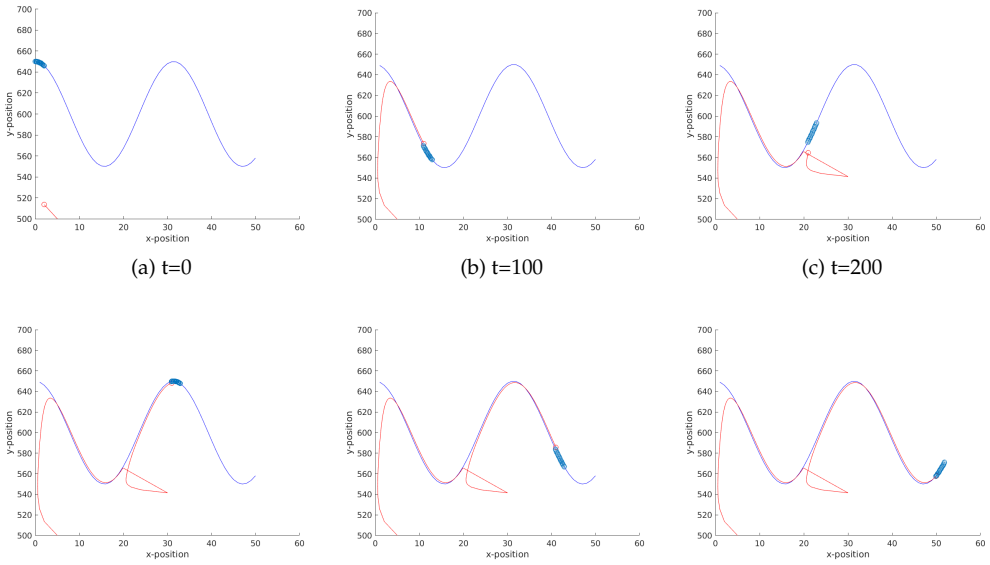
In which t is the timestamp.

Figure 8.4 shows a simulation of the model predictive contour strategy with an artificial error introduced as well as the motor actuations for both the x- as well as y-axis. From this can be concluded that this type of MPC responds relatively well to an introduced error (despite moving back for some time). On first sight MPC with a fixed horizon seems as suitable strategy. However, their reliance on time are a significant downside. Due to this, the user will need to move at a very specific pre-determined speed. Moving faster or slower will increase the error significantly. This is also the reason for needing to 'move back' after an error is introduced.

8.3 Model Predictive Contour Control

8.3.1 Introduction

As is shown above MPCs suffer from several flaws. However, the most prominent one is not being seen. MPCs, due to their nature, use timestamps in order to determine the next setpoints. This means that they are highly inflexible and will always move forward with a predetermined pace. This is a problem in several scenarios. The most prominent being when the user is lagging behind the system, the error will increase significantly.



(a) t=0 (b) t=100 (c) t=200

(d) t=300 (e) t=400 (f) t=500

Figure 8.4: Simple MPC, Blue line is desired path, red is past electromagnet positions, blue circles are desired points at that time frame.

8.3.2 Formalization

MPCCs solve this problem by also optimizing the reference point to those closest to the actual position of the pen. This is done by not only minimizing the cost function over u_i and x_i, but also θ and $\dot{\theta}$. These latter are the virtual set point and the virtual speed along the contour respectively.

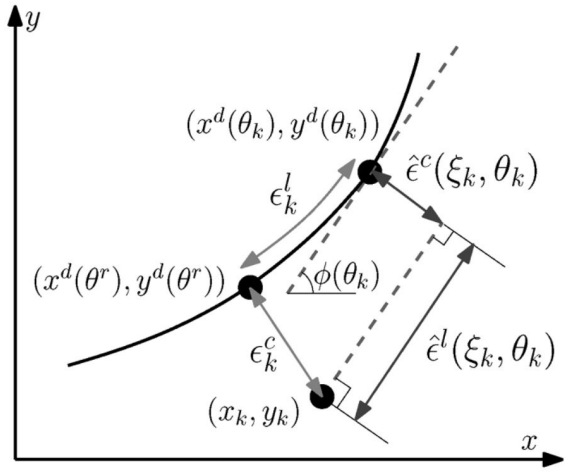


Figure 8.5: Schematic of the mpcc output as well as internal variables [29].

Solving for ϵ_k^c and ϵ_k^l (contour error and lag error, respectively) is a non-linear problem that would greatly increase the complexity of the algorithm. This is because finding $(x^d(\theta^r), y^d(\theta^r))$,

the real closest point on the desired spline, is an optimization problem on its own. For that reason, Lam et al. suggested to approach the desired coordinates in such a way that it linearly solvable.

They do this by assuming the desired path is parameterized by arc-length (which is a non-trivial problem on its own [13]). They also add the following dynamics to the system:

$$\theta_{k+1} = \theta_k + \dot{\theta}_k \tag{8.7}$$

In this $\dot{\theta}_k$ is a virtual input defined by the controller and θ_k denotes the value of the path parameter at time k. They propose to use θ_k as a close approximation of θ^r.

For this approximation they assume that θ_k approaches θ^r. In order to enforce the assumption Lam et al. introduce a lag-error, ϵ_k^l. This is the distance along the path between $(x^d(\theta^r), y^d(\theta^r))$ and $(x^d(\theta_k), y^d(\theta_k))$. However, since the true $(x^d(\theta^r), y^d(\theta^r))$ are unknown they will approximate it using:

$$\hat{e}^l(x_k, \theta_k) = -\cos\phi(\theta_k)(x_k - x^d(\theta_k)) - \sin\phi(\theta_k)(y_k - y^d(\theta_k)) \tag{8.8}$$

it can be observed from Figure 8.5 that when $\hat{e}^l(x_k, \theta_k) \rightarrow 0$ it is true that $(x^d(\theta^r), y^d(\theta^r)) \rightarrow (x^d(\theta_k), y^d(\theta_k))$. Hence this assumption holds true, in practice, when there is a high penalty weight in the cost function on $\hat{e}^l(x_k, \theta_k)$. When this holds true it is relatively straightforward to calculate the distance between (x_k, y_k) and $(x^d(\theta_k), (y^d(\theta_k))$, the contour error:

$$\hat{e}^c(x_k, \theta_k) = \sin\phi(\theta_k)(x_k - x^d(\theta_k)) - \cos\phi(\theta_k)(y_k - y^d\theta_k) \tag{8.9}$$

From this the cost function C_k can be formulated:

$$C_k = \sum_{i=1}^{N} \left(\begin{bmatrix} \hat{e}^l(x_{k+i}, \theta_{k+i}) \\ \hat{e}^c(x_{k+i}, \theta_{k+i}) \end{bmatrix}^T Q \begin{bmatrix} \hat{e}^l(x_{k+i}, \theta_{k+i}) \\ \hat{e}^c(x_{k+i}, \theta_{k+i}) \end{bmatrix} - q_\theta \dot{\theta}_{k+i} + \mathbf{u}_{k+i}^T R \mathbf{u}_{k+i} \right. \tag{8.10}$$

in which:

$\mathbf{u}$ is the input vector

$Q = \begin{bmatrix} q_c & 0 \\ o & q_l \end{bmatrix}$ is a matrix of weights.

$q_c, q_l, q_\theta > 0$ are the weights for the contour, lag and theta error

$q_l >> q_c$ the lag weight is bigger than the contour weight to ensure that $\hat{e}_l$ approaches e_l

R is a matrix of weights on the different inputs.

There is a reward for a positive $\dot{\theta}_k$ in order to ensure forward movement along the contour.

The definition proposed by Lam is on first sight more intuitive for those with less experience in vector calculation. However, this approach is challenging to generalize to dimensions that are bigger than 2. For this reason, Nägeli et al. [41] introduce a similar concept based on vector

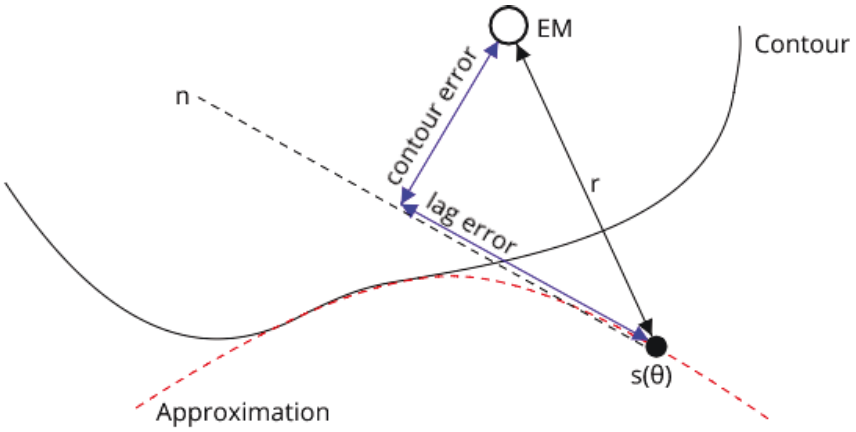


Figure 8.6: Simplified schematic of MPCC using vectors.

calculations as well as add an approximation stage to the calculations. The lag-error and contour-error are defined as follows.

Let the approximation of the contour path be $\mathbf{s}$. From this, $\mathbf{r}$ is defined as $\mathbf{r} = \mathbf{s}(\theta_k) - \mathbf{x}_k$ in which $\mathbf{x}$ is the actual position of the electromagnet. $\mathbf{n}$ is defined as the the normalized tangent vector so that $\mathbf{n} = \frac{\mathbf{s}'}{||\mathbf{s}'||}$, with $\mathbf{s}' = \frac{\partial \mathbf{s}(\theta)}{\partial \theta}$. Nägeli et al. use roughly the same cost function as Lam for the minimization., we adapted lag and contour errors, as in Equation 8.11 We will be using Nägeli et al. definition in the remainder of the thesis.

$$\begin{aligned} \hat{e}_l(x,\theta) &= ||\mathbf{r}^T\mathbf{n}|| \\ \hat{e}_c(x,\theta) &= ||\mathbf{r} - \left(\mathbf{r}^T\mathbf{n}\right)\mathbf{n}|| \end{aligned} \tag{8.11}$$

8.3.3 Evaluation

For this evaluation of MPCCs we define the cost function as:

$$J_k = w^{(l)}\hat{e}_k^{(l)} + w^{(c)}\hat{e}_k^{(c)} - w_{\dot{\theta}}\dot{\theta} \tag{8.12}$$

In which $w_l, w_c, w_{vel}, w_\theta$ are scalar weight parameters that are bigger than 0. e_l, e_c, e_{vel}. The cost function J_k is minimized in combination with a penalty on a large actuation $\mathbf{u}$ and terminal cost

at J_N on the final stage:

$$
\begin{aligned}
u^* = \min_{u_i, x_i, \theta, \dot{\theta}} & \sum_{k=0}^{N-1} \left(J_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k \right) + a_N J_N \\
\text{subject to: } & x_0 = \hat{x}(t) \text{ (Initial state)} \\
& \theta(0) = \hat{\theta}(0) \text{ (Initial path parameter)} \\
& \mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) \text{ (Internal Model Dynamics)} \\
& \theta_{k+1} = \theta_k + \dot{\theta} T_s \text{ (Virtual rail progress)} \\
& 0 \leq \theta_k \leq L \text{ (Path length)} \\
& \mathbf{x}_k \in \chi \text{ (State Constraints)} \\
& \mathbf{u}_k \in v \text{ (Input Constraints)}
\end{aligned}
\tag{8.13}
$$

In this evaluation the parameters are set to:

$$
A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix}
$$

$$
w_l = 2, w_c = 1, w_\theta = 2, w_{vel} = 1, w_u = 1, a_N = 1
$$

$$
x_d = [\theta, 50 * cos(\theta/50) + 600]
$$

$$
u_{ll} = [-Inf - Inf, -Inf], u_{ul} = [Inf, Inf, Inf], x_{ll} = [-Inf, -Inf, -Inf] x_{ul} = [Inf, Inf, Inf]
$$

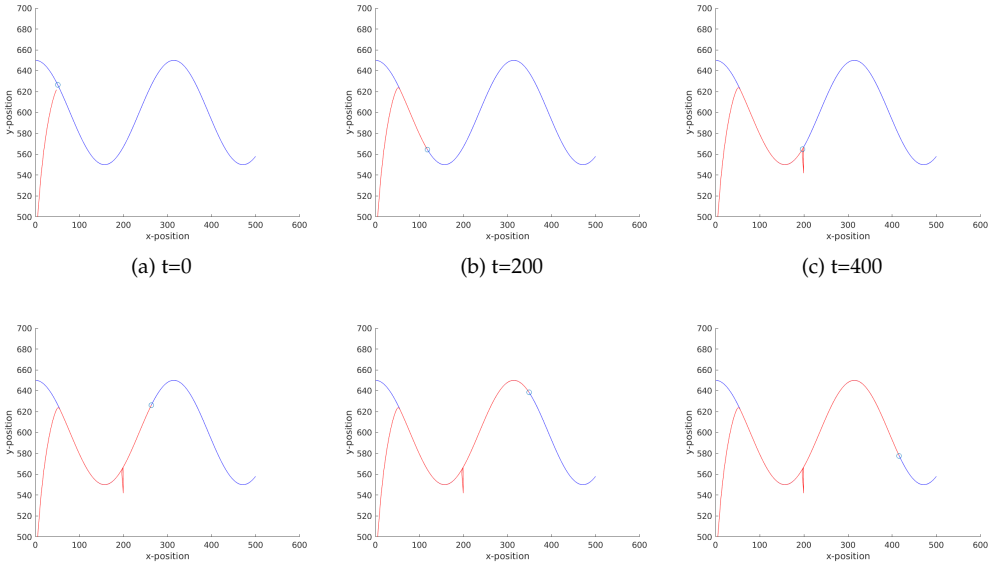
Next to this an artificial error of $[40, 0]^T$ was introduced at $t = 200$ in order to investigate the response of the MPCC on an unexpected event.

It is interesting to compare the simulation of the MPCC (Fig. 8.7) with the simulation of the MPC fixed receding horizon (Fig. 8.4). Despite the difference in scale, the time independence of the former is clearly visible. This is mainly seen by the fact that the actual path in the MPCC does not cover the seem distance as in the MPC, despite the same time. This also illustrates the importance of setting appropriate weights to the progress penalty. Despite this the MPCC performs well, also in the case of an introduced error.

8.4 Conclusion

	PID	MPC	MPCC
Predictive	×	✓	✓
Time-Independent	×	×	✓
Computational Quick	✓	×	×

Table 8.1: Advantages of different control strategies.



(a) t=0 (b) t=200 (c) t=400

(d) t=600 (e) t=800 (f) t=1000

Figure 8.7: Simple MPCC. Blue line is desired path, red line is past electromagnet points. Blue circle is desired point.

We have seen three different strategies and two different variants of the MPC strategy specifically. They are presented in ascending order of complexity. The PID controller is shown to converge on the desired path, however has an oscillating characteristic. This is disadvantageous for our setup, since we require smooth motions. Secondly, we analyzed model predictive controls. These strategies are characterized by utilizing an internal model to optimize the output over N timestamps in the future. The first variant we have taken a look at, fixed receding horizon, performs relatively well compared to its counterpart the variable receding horizon. However, both suffer from the flaw that the setpoints on the reference path are timestamped. This results in an increasing error if the pen is kept at the same position. Similarly, if the user is ' ahead' of the timestamps the error will increase. In other words, the speed of the movement is fixed. This is an undesirable characteristic. Model Predictive Contour Control overcomes this problem by making the location as well as speed along the contour an optimization parameter. This decouples time from location, allowing for more freedom for the user. One of the downsides is that the minimization problem increases in complexity (more free parameters), which in turn increases the computational cost. Despite this disadvantage, we have decided to extend the MPCC for our final model which incorporates the framework of the electromagnet.

Part IV

System

In this chapter we will extent on our final system design. First of all, we will discuss the proposed hardware and secondly, we will discuss the MPCC. For defining the MPCC we will use the contents of the previous part extensively.

Chapter 9
Hardware

DESPITE the hardware not being used in the evaluation of the proposed control strategy (in this thesis) we will elaborate on the design. This is for several reasons, first of all a significant amount of time was spend on it and secondly it will help in the further explanation of Model Predictive (Contour) Control.

The hardware exists out of several components that can be subdivided into several categories. First of all, the electrical components and secondly the non-electrical components. The latter is the axis system along which the electromagnet moves. The former can be again subdivided into sensors, actuators and controllers that all interplay together.

9.1 Physical Aspect

9.1.1 Axis

Having a smooth, reliable and low-resistance axis system is vital for having accurate and fluent motions of the electromagnet. For this reason, we re-purposed a MakerBot Replicator 2X. Using the MakerBot as the basis for our design has several advantages. First of all, it is build for 3D printing, which has the same requirements regarding accuracy and smoothness of the movement. Secondly, the stepper motors, that will be used to control the position of the electromagnet are already in place. We detached the printer head from the MakerBot, since we do not have need of that and replaced it with a 3D printed plateau on which the electromagnet will be placed. On the top of the MakerBot we added the surface on which the user will write. The plateau is made exactly to the height, so that there is only a millimeter between the top of the electromagnet and the surface. Since the strength rapidly declines over distance, having this space minimized was important. The inside of the setup can be seen in Figure 9.1

9.1.2 Pen

The second passive element in the setup is the pen. We have made our own stylus, so that we can incorporate permanent magnets in the base. Adding the electromagnets will allows us the make an easier and more accurate model to predict the strength of the electromagnet on the pen. By solely relying on the ferromagnetic nature of the pen-tip, we could have unpredictable interactions.

The design of the pen resembles, as closely as possible, a real pen in shape and size. However, the diameter is slightly bigger, in order to fit the two ring magnets in the bottom. The ring magnets are two Neodymium magnets with a 10mm diameter, with a 4 mm hole (to allow for the ink

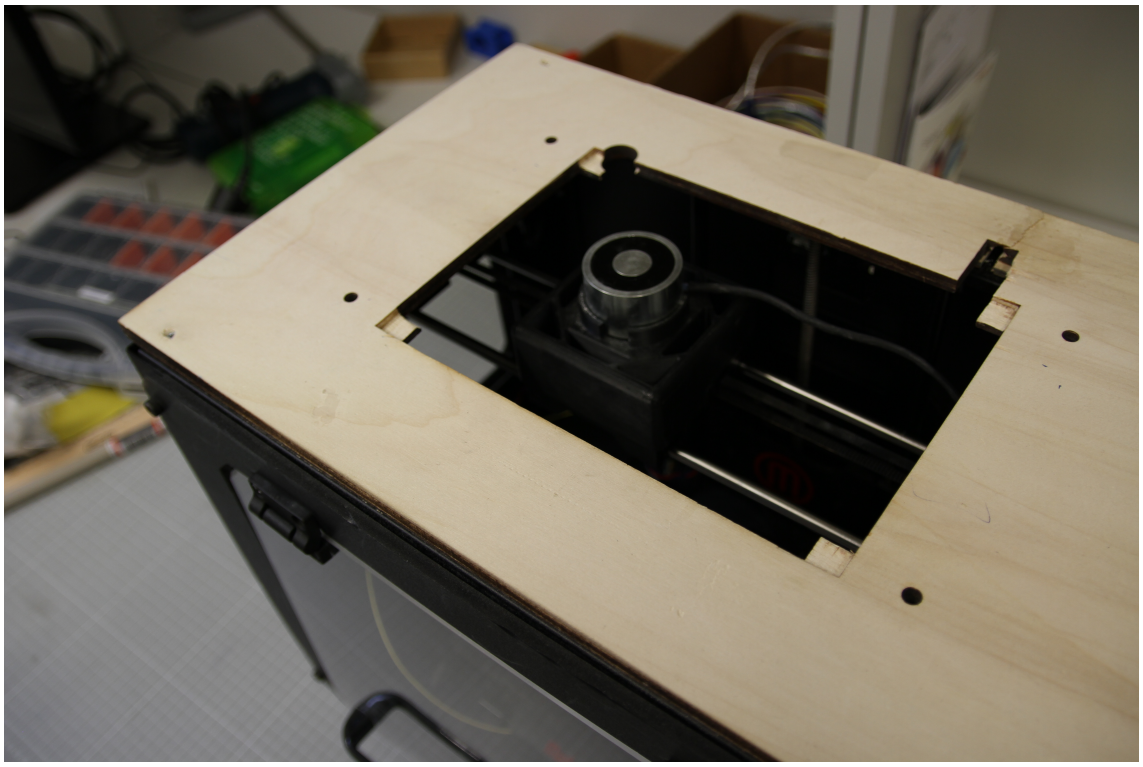


Figure 9.1: System without Sensel Morph.

cartridge) and a N42 magnetization. The pen itself has a diameter of 14mm and a length of 140mm . The pen was construct with a Stratasys F370 3D printer and constructed out of two halves (along the length of the pen), to make the exterior of the pen as round as possible. The printer files for the pen can also be found on GitHub.

Figure 9.2: Designed pen next to normal pen for dimensions.

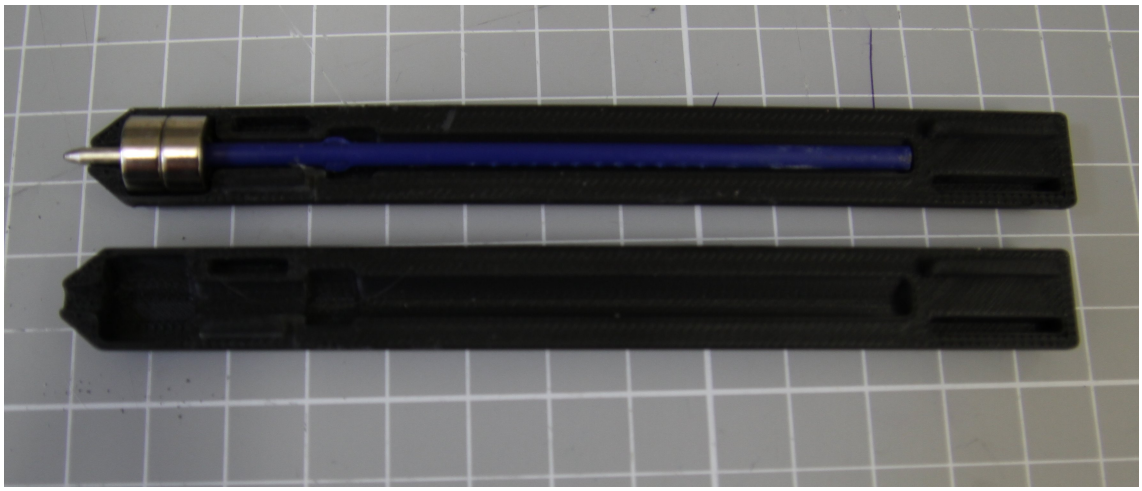


Figure 9.3: Inside of the designed pen.

9.2 Electronics

As stated the electronics are a combination of sensors, actuators and processing units that collaborate to measure the pen position, calculate the new electromagnet position plus strength and then move the electromagnet to the correct location. The overall circuit can be found in Figure 9.4.

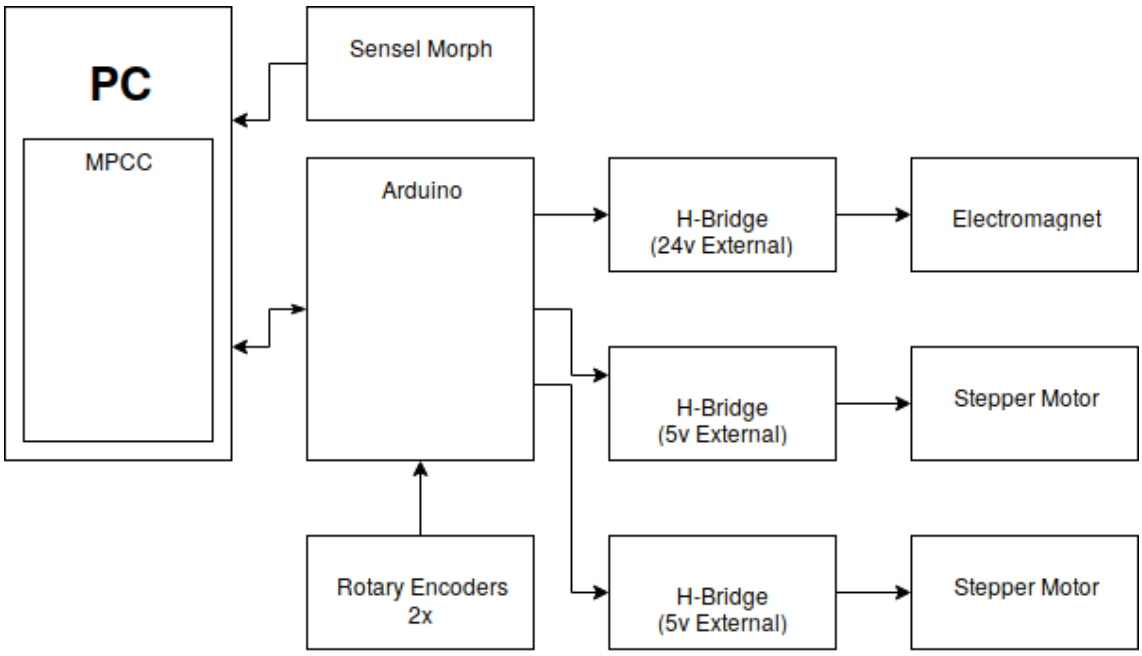


Figure 9.4: Simplified version of the designed circuit

9.2.1 Pen Location

In order localize the position of the pen accurately we are making use of a Sensel Morph Pressure Grid[1]. This is an extremely accurate multi-touch pad that also allows us to measure the downward force exerted. Despite this, we are not using the multi-touch nor the force feature at the moment. The Sensel Morph is connected to a PC via a USB connection.

Figure 9.5: System with Sensel Morph.

9.2.2 Processing

The solving of the MPCC happens on a PC (16GB, Intel Core i7-4770 CPU @ 3.40GHz x 4) running Ubuntu 17.10. It is important to notice that the operating system is vital. Since the code provided will only run on Ubuntu. It was also tested on the LTS version 16.04, however this did not work, 17.04 and 18.04 LTS are untested. The PC is connected to the Sensel Morph and an Arduino UNO via USB. The Arduino controls the stepper motors as well as the electromagnet, no significant calculations happen here.

[1] `https://sensel.com/`

9.2.3 Electromagnet Position, Calibration & Strength

The electromagnet position is controlled by two stepper motors (Moon's type: 17HD4063-03N), one for each axis on the plane. The stepper motors are connected to the Arduino via a H-Bridge to control the speed and direction. The H-bridge is also connected to a 5v external power supply, to ensure that the motors receive enough current. Since we use acceleration as an input, we added a rotary encoder to keep track of the exact position of the electromagnet (eg. if our input to the system is accelerate with 1 step per second^2 for 0.1 seconds, however in practice it is 0.095 seconds the error start to accumulate).

The electromagnet is attached to a PKA03'1 motor shield[2]. This allows us to easily use Pulse Width Modulation (PWM) in combination with an external power supply (24v) to control the strength of the electromagnet. Whereas a normal MOSFET or transistor would only allows to increase or decrease strength, the motor shield also allows to change the direction.

[2] https://www.velleman.eu/products/view/?id=412174

Chapter 10

Method

The final model proposed, in combination with the evaluation, consists out of two main parts. First of all the MPCC in general and, secondly, the user simulation to validate the findings. The MPCC, in turn, exists out of two elements. The system model and the cost function. The latter consists built up from different cost terms. This will also be the layout of this chapter. An overview of the method used for this thesis is depicted in Figure 10.1, this should be slightly adapted in a real-world context as is shown in Figure 10.2.

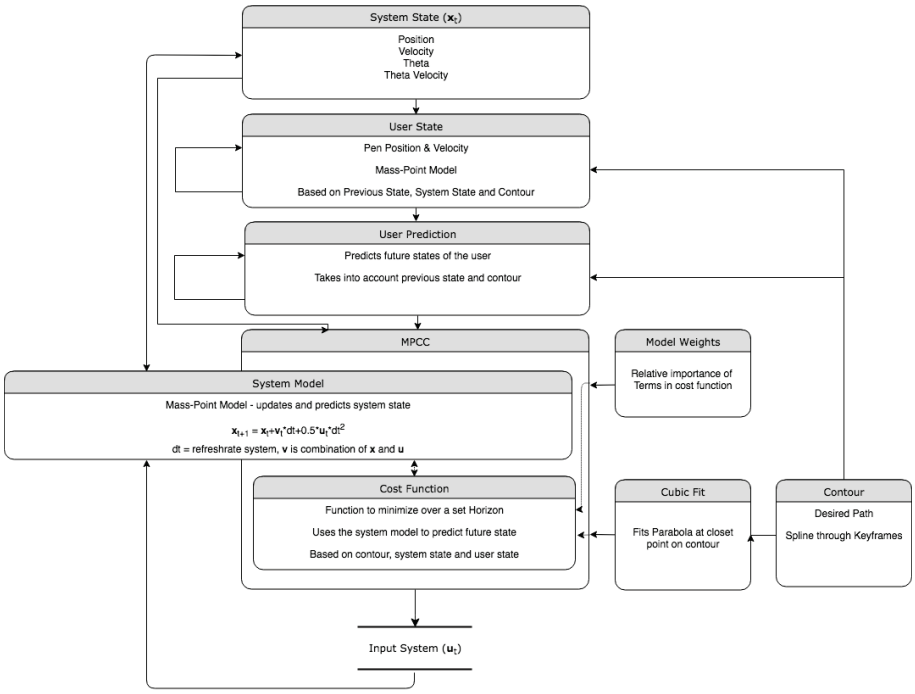


Figure 10.1: Flowchart of the proposed method in a simulation context.

10.1 MPCC

10.1.1 System Model

The internal system model is a simple mass point model. The system model represents how the hardware will, physically, react to certain inputs. It allows for the cost function to predict the future states of the system. It is this part that brings ”Model Predictive” to the name of this control

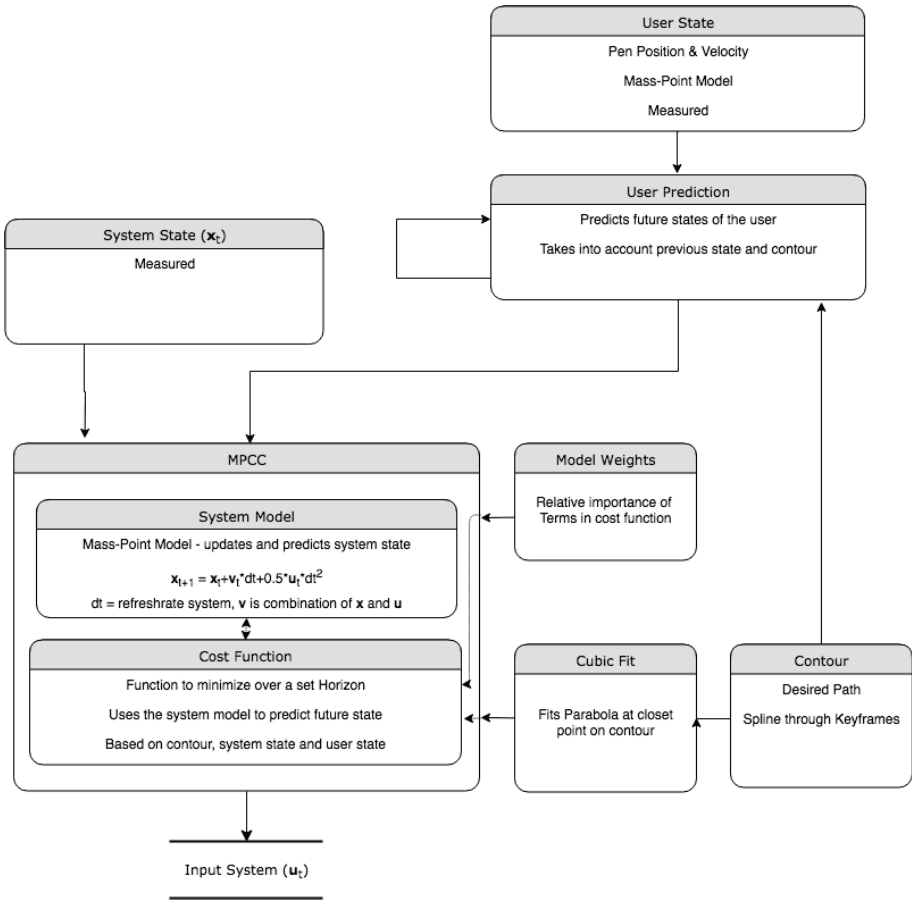


Figure 10.2: Flowchart of the proposed method in a real user context.

strategy. The state of the system, $\mathbf{x}$ is defined as:

$$\mathbf{x} = \left[x_{em}, y_{em}, \dot{x}_{em}, \dot{y}_{em}, \alpha, \dot{\alpha}, \theta, \dot{\theta}\right]^T$$

Similarly the input $\mathbf{u}$ is defined as:

$$\mathbf{u} = \left[\ddot{x}_{em}, \ddot{y}_{em}, \ddot{\alpha}, \ddot{\theta}\right]^T$$

Notice that the input to the system is acceleration, and velocity and position are regarded as states. By making the input acceleration, we reduced jerk in the system. Which would be more likely present if velocity was the input. This leads to a system model based on past state, velocity, acceleration and dt, which is the refresh rate of the system:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v} * dt + \frac{1}{2} * \mathbf{a} * dt^2 \tag{10.1}$$

When filling in the variables, this leads us to:

$$\begin{bmatrix} x_{em} \\ y_{em} \\ \dot{x}_{em} \\ \dot{y}_{em} \\ \alpha \\ \dot{\alpha} \\ \theta \\ \dot{\theta} \end{bmatrix}_{t+1} = \begin{bmatrix} x_{em} \\ y_{em} \\ \dot{x}_{em} \\ \dot{y}_{em} \\ \alpha \\ \dot{\alpha} \\ \theta \\ \dot{\theta} \end{bmatrix}_{t} + \begin{bmatrix} \dot{x}_{em} \\ \dot{y}_{em} \\ \ddot{x}_{em} \\ \ddot{y}_{em} \\ \dot{\alpha} \\ \ddot{\alpha} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix}_{t} * dt + \frac{1}{2} * \begin{bmatrix} \ddot{x}_{em} \\ \ddot{y}_{em} \\ 0 \\ 0 \\ \ddot{\alpha} \\ 0 \\ \ddot{\theta} \\ 0 \end{bmatrix}_{t} * dt^2 \tag{10.2}$$

10.1.2 Cost Function

The cost function, J_c, is the element of the MPCC that the solver tries to minimize. There are usually several cost terms that are given different weights, based on the importance of the term in the overall system. In our cost function there is the lag and contour error ($\mathbf{e}_{lc}$), as elaborated on in Chapter 8. In addition to this we add an error the control the strength of electromagnet on the pen (e_{em}), based on the distance from the pen as well as α. Since the strength of the electromagnet declines rapidly over distance, as seen in Chapter 7, we add an additional term that heavily punishes being out side a defined radius (e_d). The last term is, again standard, in which we add a cost to the input ($\mathbf{u}$). This is important as it allows for the solver to converge.

The contour ($\hat{e}_c$) and lag error ($\hat{e}_l$) are defined in Chapter 8. For the sake of completeness, we will repeat it here. Let us define the contour path as $\mathbf{s}$ and a virtual speed along that path as $\dot{\theta}$, so that $\mathbf{s}(\theta)$ is a specific location a long that path. Then we can define $\mathbf{r}$ as $\mathbf{r} = \mathbf{s}(\theta) - \mathbf{x}$, where $\mathbf{x}$ is the current position of the electromagnet. Similarly we can define $\mathbf{n}$ to be the normalized tangent. The tangent at θ is defined as $\mathbf{s}'(\theta) = \frac{\partial \mathbf{s}(\theta)}{\partial \theta}$, which leads to $\mathbf{n}(\theta) = \frac{\mathbf{s}'(\theta)}{||\mathbf{s}'(\theta)||}$. From these we can define the contour ($\hat{e}_c$) and lag error ($\hat{e}_l$), which is the used definition in the MPCC.

$$\begin{aligned} \hat{e}_l(x,\theta) &= ||\mathbf{r}^T\mathbf{n}|| \\ \hat{e}_c(x,\theta) &= ||\mathbf{r} - \left(\mathbf{r}^T\mathbf{n}\right)\mathbf{n}|| \\ \mathbf{e}_{lc} &= \begin{bmatrix} \hat{e}_l(x,\theta) \\ \hat{e}_c(x,\theta) \end{bmatrix} \end{aligned} \tag{10.3}$$

From this it follows that the weight matrix Q should be constructed as:

$$Q = \begin{bmatrix} w_l & 0 \\ 0 & w_c \end{bmatrix} \tag{10.4}$$

Electromagnetic Strength Error

In the proposed cost function we use theoretical model of the electromagnetic strength model as elaborated on in Chapter 7. It is defined as the square of the difference in the desired in-plane strength versus the actual in-plane strength. In order to recall, the force on the pen is the PWM (α) times the in-plane force (F_φ). So that:

$$F_{EM\to P} = \alpha F_\varphi = \alpha f_e \sqrt{(f_x)^2 + (f_y)^2} \tag{10.5}$$

where f_e is a constant depended on the magnets in the system. f_e is defined in Equation 7.16. f_x and f_y are defined as follows:

$$\begin{aligned} f_x =& \frac{u(q^2-5)}{q^7}\left[\cos\theta + \left[\frac{(q^2-5u^2)\cos\delta - 5uv\sin\delta}{u(q^2-5)}\right]\sin\theta\right] \\ f_y =& \frac{v(q^2-5)}{q^7}\left[\cos\theta + \left[\frac{(q^2-5v^2)\sin\delta - 5uv\sin\delta}{v(q^2-5)}\right]\sin\theta\right] \end{aligned} \tag{10.6}$$

with:

$$\begin{aligned} u =& x_1/z_1 \\ v =& y_1/z_1 \\ q =& r/z_1 \end{aligned} \tag{10.7}$$

θ and δ are the angle between the pen and the vertical (z) axis, and the angle between the pen and the, horizontal, x-axis. In which the x,y axis are a plane that is directly on the writing surface.

Based on this we can define this cost term as:

$$e_{em} = ||F_{EM\to P} - F_d|| \tag{10.8}$$

in which F_d is the desired strength.

The strength perceived by the pen from the electromagnet declines quadratically over distance. For this reason, an error term is added to penalize the electromagnet being outside to effective distance. This term is interesting, since the error term itself changes based on the distance. We can define a circle with radius r around the pen:

$$\frac{(x-x_p)^2 + (y-y_p)^2}{r^2} = 1 \tag{10.9}$$

in which the x, y-pairs are points on the radius of circles From this follows that the distance from the radius of the circle is:

$$d = \sqrt{\frac{(x_{em}-x_p)^2 + (y_{em}-y_p)^2}{r^2}} - 1 \tag{10.10}$$

If $d > 0$ the electromagnet is outside the radius, otherwise the electromagnet is on or inside the radius. It is here, that the distinction is made in different error terms.

$$e_d = \begin{cases} 1000 * d, & \text{if } d > 0 \\ 0, & \text{otherwise} \end{cases} \tag{10.11}$$

However, it also not ideal if the electromagnet is too close to the pen. Not only does the strength in the x, y-plane decrease and the vertical strength increase. But also, it allows for the pen the move over the electromagnet within the timeframe. Resulting in that the electromagnet is behind the pen. For this reason we define a very similar definition as above.

$$d_{inner} = \sqrt{\frac{(x_{em} - x_p)^2 + (y_{em} - y_p)^2}{r_{inner}^2}} - 1 \tag{10.12}$$

in which $r_{inner} < r$. However, this time error term is also slightly different:

$$e_{close} = \begin{cases} 1000 * d_{inner}, & \text{if } d < 0 \\ 0, & \text{otherwise} \end{cases} \tag{10.13}$$

We add a heavy penalty on being inside the inner circle. This results in a donut-shaped area for the electromagnet to be in.

The input errors are simply defined as $\mathbf{u}$. Adding a penalty on an high $\mathbf{u}^T\mathbf{u}$ has two advantages. First of all, it allows the model to converge. Secondly, by limiting the acceleration the overall change in the states of the model will be more smooth. By defining the weight matrix R as:

$$R = \begin{bmatrix} w_{\ddot{x}} & 0 & 0 & 0 \\ 0 & w_{\ddot{y}} & 0 & 0 \\ 0 & 0 & w_{\ddot{\alpha}} & 0 \\ 0 & 0 & 0 & w_{\ddot{\theta}} \end{bmatrix} \tag{10.14}$$

different weights to different inputs can be given. In the proposed system the weight on $\ddot{\alpha}$ and $\ddot{\theta}$ are relatively low compared to $w_{\ddot{x}}$ and $w_{\ddot{y}}$. In order for our electromagnet to adapt quickly in strength as well as that θ is on the closest position on $\mathbf{s}$ from the electromagnet.

This leads to the cost function, were the sum of all terms over the horizon length, N, is minimized:

$$J_c = \sum_{n=1}^{N} \mathbf{e}_{lc} Q \mathbf{e}_{lc}^T + w_{em} * e_{em}^2 + w_d * e_d^2 + w_{close} * e_{close} + w_\theta * ||\theta - \theta_f||^2 + \mathbf{u} R \mathbf{u}^T \tag{10.15}$$

in which

θ_f is the final theta of the spline.

Q is a 2x2 matrix, usually diagonal.

R is a 4x4 matrix, usually diagonal.

10.2 User Model

In order to be able to validate the MPCC a user model was necessary. This model approximates how a user will react to the electromagnet, as well as is in input for the MPCC that needs future pen positions on the horizon length. For this reason we created two different models.

10.2.1 User Model - Update

The model responsible for updating the pen position takes into account the current velocity (and direction) of the pen movement, the position of the electromagnet and the contour. It should be noted that currently it does not take into account the strength of the electromagnet. This results in the update formula:

$$\begin{aligned}
\mathbf{a}^{(\mathbf{t})} &= \frac{(\mathbf{p}_{pen}^{(t)} - \mathbf{p}_{em}^{(t)})}{||\mathbf{p}_{pen}^{(t)} - \mathbf{p}_{em}^{(t)}||} \\
\mathbf{b}^{(\mathbf{t})} &= \frac{\mathbf{v}^{(\mathbf{t-1})}{}_{pen}}{||\mathbf{v}^{(\mathbf{t-1})}{}_{pen}||} \\
\mathbf{c}^{(\mathbf{t})} &= \frac{\mathbf{s}^{(t)}(\theta) - \mathbf{p}_{pen}^{(t)}}{||\mathbf{s}^{(t)}(\theta) - \mathbf{p}_{pen}^{(t)}||} \\
\mathbf{v}^{(\mathbf{t})} &= \frac{(w_a * \mathbf{a}^{(\mathbf{t})} + w_b * \mathbf{b}^{(\mathbf{t})} + w_c * \mathbf{c}^{(\mathbf{t})} + n) * vel}{dt} \\
\mathbf{acc}^{(t)} &= \frac{\mathbf{v}^{(\mathbf{t})} - \mathbf{v}^{(\mathbf{t-1})}}{dt} \\
\mathbf{p}_{pen}^{(t+1)} &= \mathbf{p}_{pen}^{(t)} + \mathbf{v}^{(t)} * dt + \frac{1}{2}\mathbf{acc}^{(t)} * dt^2
\end{aligned} \tag{10.16}$$

$\mathbf{p}_{pen}^{(t)}$ and $\mathbf{p}_{em}^{(t)}$ are the position of the pen and the electromagnet and time t respectively. $\mathbf{v}^{(t)}$ is a velocity vector of the pen and $\mathbf{s}^{(t)}(\theta)$ is the closest point on the contour. w_a, w_b and w_c are weights for the different components (so that $w_a + w_b + w_c = 1$), n is Gaussian distributed noise and vel is a velocity constant. The different vectors are illustrated in Figure 10.3.

10.2.2 User Model - Prediction

The predictive model is similar to the updating model in Equation 10.16. The major differences are that the electromagnet is not taken into account, that there is no noise and that the weights, w_b and w_c, are slightly adjusted to counter the lag of w_a. vel should be exactly the same.

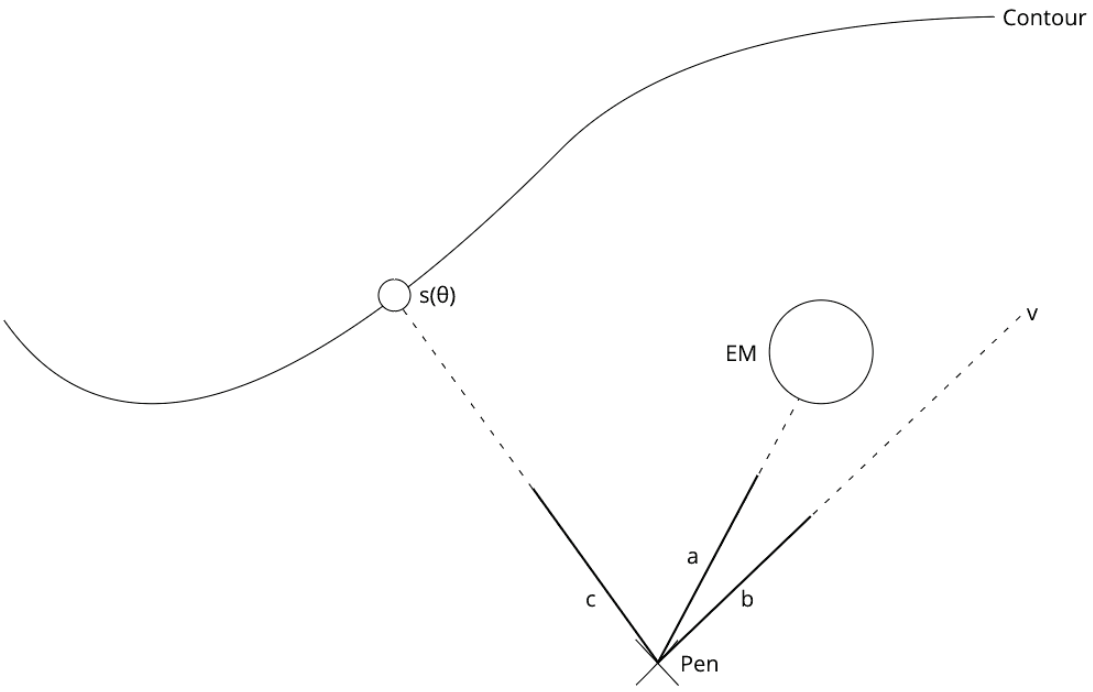


Figure 10.3: Overview of the components of the user model.

Part V

Evaluation

Finally, we will evaluate our proposal. First, we introduce how we will evaluate the different parts of the system. As well as, despite that this has not been conducted yet, propose a user study. We will show the results as well as discusses them. We will end by draw conclusions and suggest future work.

Chapter 11
Evaluation Procedure

THE evaluation of our method exists out of multiple phases. In this thesis the main focus is on simulating a user and numerically validate the model as well as visually inspect the characteristics of the model. The numerical validation gives an indication on the computation time as well as overall quality of the result. The latter will highlight cases in which the implemented MPCC is not guaranteed to work. Secondly, we will extent on a proposed user study. Due to time constraints, this user test was not conducted. However, it will be done over the coming weeks.

11.1 Simulation

11.1.1 Implementation Details

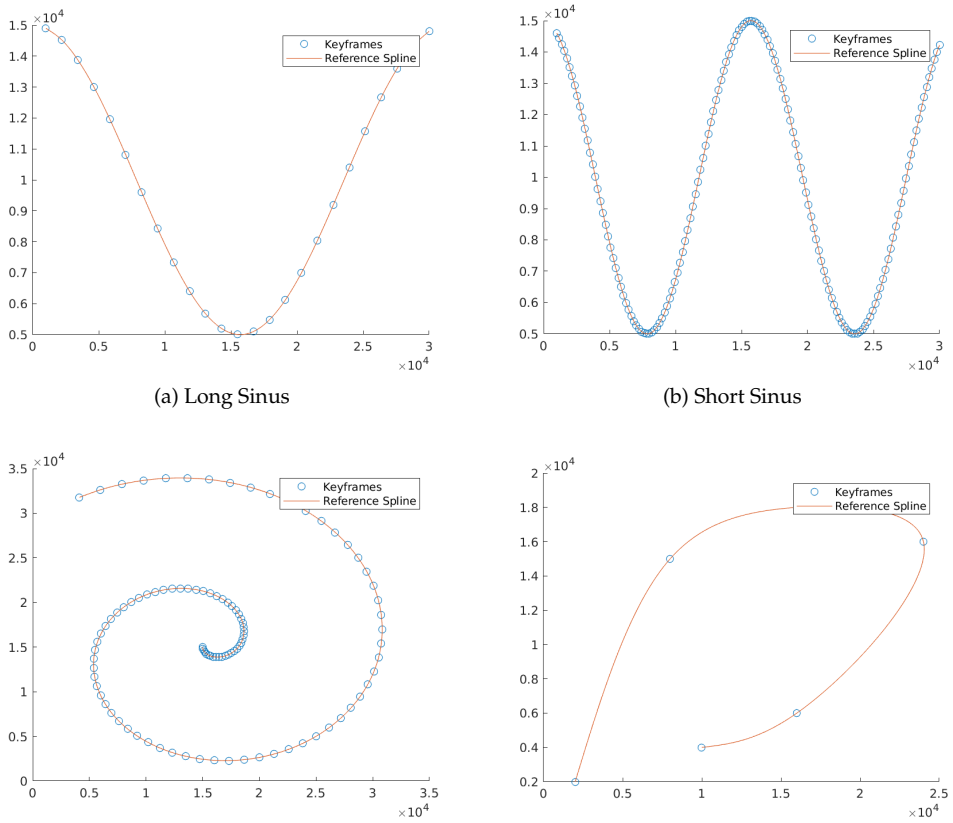
The MPCC as well as the simulation are implementd on a single desktop (16GB, Intel Core i7-4770 CPU @ 3.40GHz x 4, Ubuntu 17.10) using MATLAB. At the moment no specific attention has been paid to improve the computational efficiency of the MPCC. The MPCC problem is solved by the Forces Pro software [10]. Force Pro generates fast solver code exploiting charasteristics of non-linear programming. It allows for the user to define the problem mathematically, this results in a relatively intuitive interface. Furthermore, it is both MATLAB as well as Python (2.x and 3.x) compatible. This compatability is useful in the further stages of our project, since it allows us the define the problem in MATLAB yet use the solver in Python.

Parameters

The overall system can be initialized with a large variety of different parameters that will determine how the system will be have. These are usually set with trial-and-error until a suitable combination has been found. For the complete evaluation the same parameters have be chosen. They can be viewed in Appendix A.1.

11.1.2 Quantitative Analysis

In the quantitative analysis we determine the quality of our model. This is done in two different ways. First of all we will take a look at the computational time for the MPCC solver and the user prediction model. This is interesting because our systems needs to be capable of a high refreshrate. However, it is important to note that no extraordinary effort has been made currently to optimize this performance as well as that the prediction horizon of the MPCC has an tremendous influence (predicting 20 timesteps into the future versus 30 will be a major difference in terms of



(a) Long Sinus

(b) Short Sinus

(c) Spiral

(d) Free Form

Figure 11.1: The different contours used for the evaluation

quality as well as computational time), yet this was treated as a fixed variable during the complete experiment.

The computational times will be measured with the MATLAB tic-toc function. We will measure it for 4 different scenarios (Figure 11.1), 3 different starting positions per scenario and three different noise levels per starting position. This creates a total of 26 different problems. We collecting the times each time the solver of the user model is used. This happens for every timestep until the pen reaches the final stage or for a maximum of a 1000 steps.

11.1.3 Visual Analysis

With the general visual analysis we will take a close look at the resulting paths of the pen and the electromagnet. We will try to detect any anomalies. This could be in the paths, but also the interaction between the electromagnet and the pen. From this we will gain cases in which the proposed method does not work.

11.2 Proposed User Study

Despite the face that there has been no user test conducted for this thesis, we will present a plan here on how to do our future user test. This plan assumes a completely working system as currently envisioned.

In the introduction we proposed the use case for patients recovering from a stroke. However, these patients are challenging to approach, it introduces more variables and gaining a large number of participants is difficult. Because of these reasons, we have decided to conduct the user test with healthy subjects.

Next to this, we briefly discuss in Related Literature, Chapter 2, the challenges and pitfalls while measuring the learning of a simple motor tasks, such as measuring retention. We will take the advice, such as comparing two groups, Schmidt et al. [55] offer to solve these problems into account.

Combining everything thus far, the question that requires an answer most is: *How much faster do subjects learn a certain drawing task when using our system, compared to subjects not using our system?* The nature of this question already divides our group of participants into two groups, a test group and a control group. The only difference between the two groups is that for the latter the electromagnet is turned off.

The next element that needs to be defined is the error measure. This measure should be an objective and quantitative means to approximate the learning curve. We have chosen the Root Mean Squared Error (RMSE) for this measure. Since we can compute the exact curve, as well as know the pen location as each point in time we can calculate this. The measure is expected to reduce with practice, and to reduce faster in the group using our system.

We will measure the RMSE per user for a set number of repetitions for a set number of different contours. After the user has done this, we will ask them to fill in a survey regarding their experiences. However, this survey is mostly meant as a break. After the survey the subjects will draw each contour once more. By separating it into two parts, we can approximate the retention (despite that, arguably, the break is too short to truly measure the retention).

Chapter 12
Results & Discussion

IN this chapter we will show the results obtained from the simulation part of the evaluation. Next to this we will elaborate on the implication of the results and how we can use them to determine the applicability of our method.

12.1 Quantitative

One of the first things we will take a close look at is the computational time our method needs. For this we will plot the time it takes for the solver to reach a solution, over the horizon in different cases; different starting points, contours and noise levels. Next to this we will also investigate whether the distance from a contour has influence on the solving time.

Table 12.1: Overview of computational time for the different starting positions.

	Avg (s)	SD (s)
On path	0.1342	0.0388
Above path	0.1325	0.0398
Below path	0.1330	0.0391

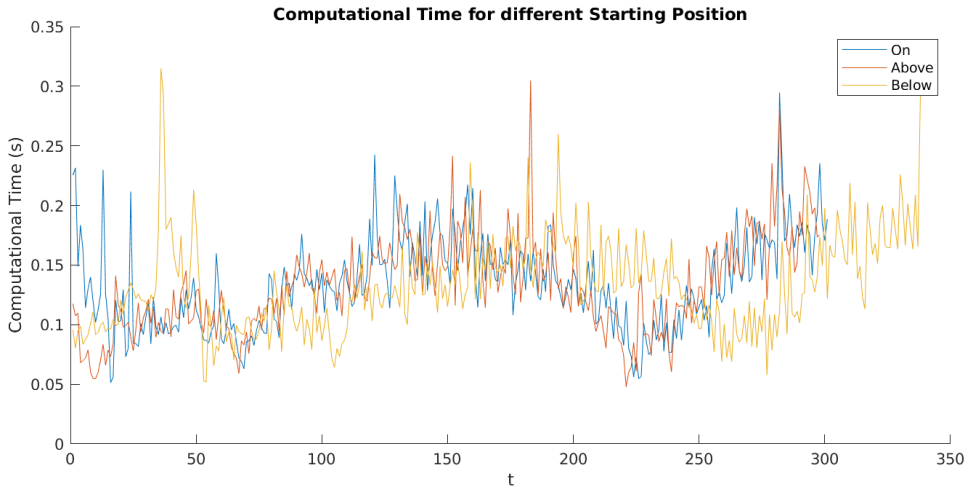


Figure 12.1: The computational time of the solver over iterations, for different starting positions on the same contour.

In Figure 12.1 we compare the computational time against different starting positions. For this the desired path was always a short sinus with no added noise to the user. The results are summarized in Table 12.1. We can see that there is no significant difference between the different

starting positions on average. Despite the standard deviation being fairly high (almost 33%) it does not differ between the different cases. However, what we see is an interesting sinusoidal behavior in the computational time. This coincides with the desired path also being a sinusoid. Yet, we lack any explanation why this behavior happens and this should be investigated in the future.

Table 12.2: Overview of computational time for the different contours.

	Avg (s)	SD (s)
Short Sinus	0.1342	0.0388
Long Sinus	0.1172	0.0358
Spiral	0.1621	0.0393
Free Form	0.1981	0.0566

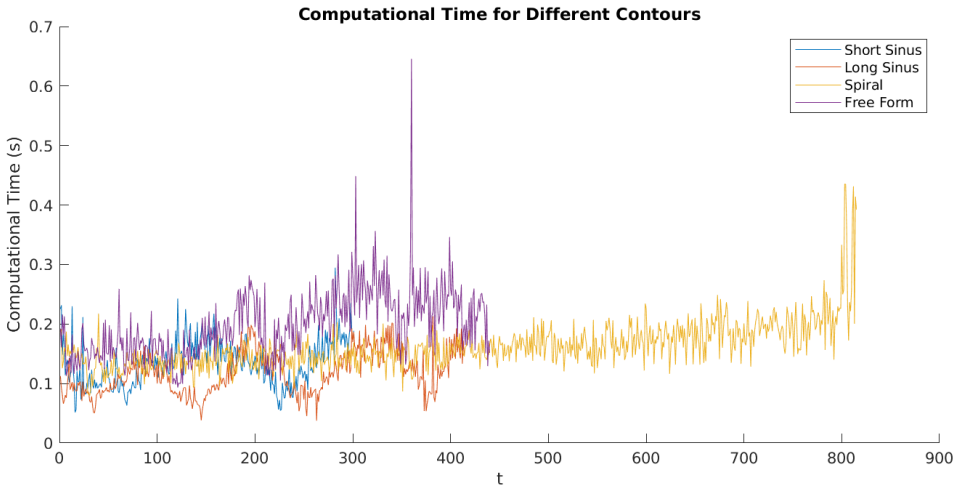


Figure 12.2: The computational time of the solver over iterations, for different contours.

Secondly, we took a look at different contours. For this experiment the pen was always on the path and there was no added noise. The results are summarized in Table 12.2. The computational time averages are in the same order of magnitude as before, however the spiral and free from desired path take slightly longer on average. Both also have an increased standard deviation which could indicate peaks that skew the average. When taking a look at Figure 12.2 we can definitely see peaks in the third and fourth case. Next to this we can again observe the sinusoidal behavior for the short and long sinus, in which the former seems to have a shorter wavelength.

Table 12.3: Overview of computational time for the different noise levels

	Avg (s)	SD (s)
0%	0.1342	0.0388
10%	0.1353	0.0385
30%	0.1350	0.0394

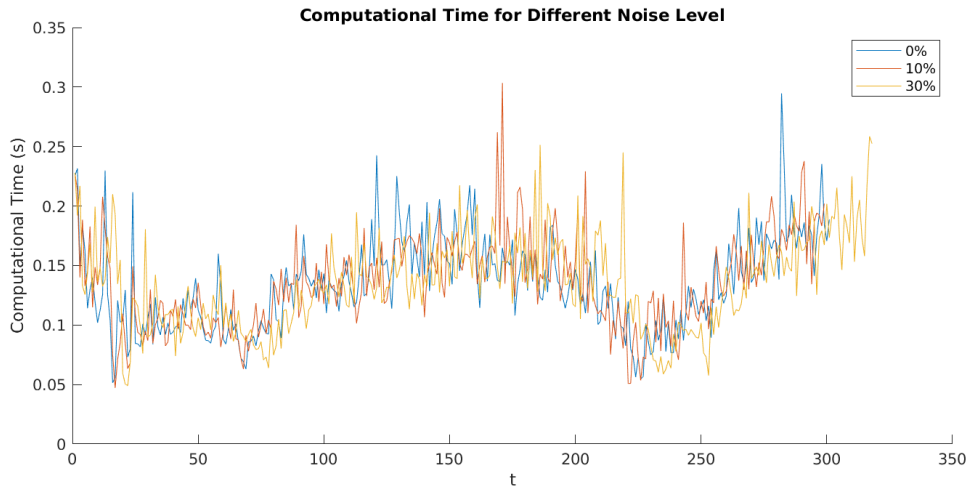


Figure 12.3: The computational time of the solver over iterations, for different contours.

Thirdly, we wondered whether an increased noise in the user would increase the average computational time. We kept the contour on a long sinus and let the pen start exactly on the contour. The results are shown in Table 12.3 and visualized in Figure 12.3. There appears to be no clear correlation between the noise (therefore inaccuracy) and the computational time.

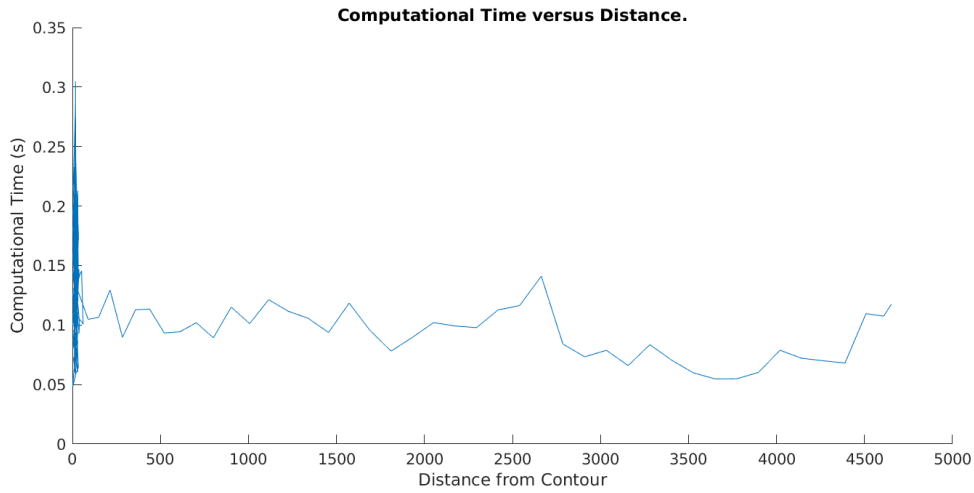


Figure 12.4: The computational time of the solver versus different distances from the contour.

Finally we took a look at the distance versus computational time which is visualized in Figure 12.4. This shows the computational time for the long sinus, with no noise where the pen starts above the desired path. Again we see no clear correlation between the distance and computational time. The large peak close to 0 distance seems to be one of the extreme cases corresponding to $t \approx 290$ in Figure 12.3.

It is fairly challenging to discuss the result of the distance from the pen to the contour without

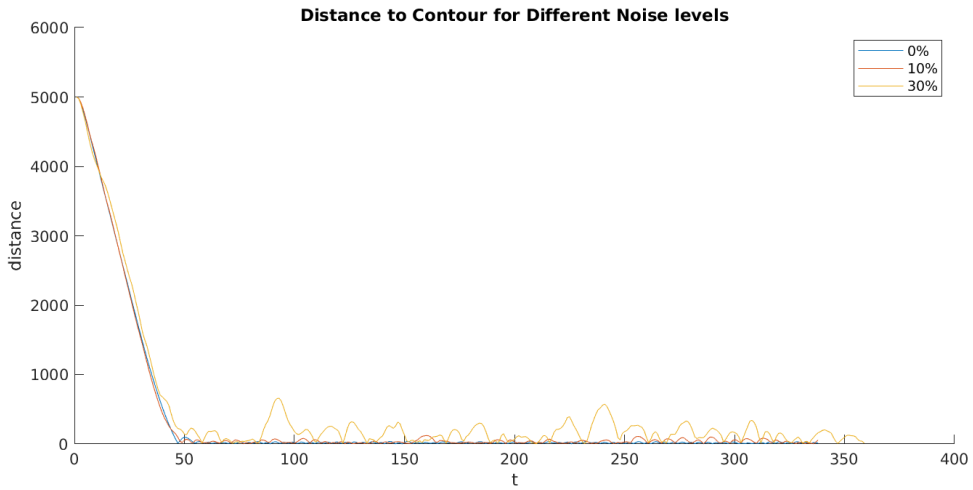


Figure 12.5: Pen distance from contour. In a the case of a simple sinus. The pen start 5000 units of the contour.

a proper user test being included. This metric was originally intended to be compared to real users. Without this data, getting meaningful insights is hard. Naturally, we expect the pen move towards the contour. Similarly, it is evident in Figure 12.5 that users with higher noise levels tend to be farther away from the contour. The error recovery (t=0 to roughly t=50) goes in a fairly constant speed which de-accelerates when the distance decreases. This illustrates that the movement is relatively smooth. Next to this we can observe in the figure, especially in the high noise cases, that the peaks are not sharp but a bit rounded. This further supports the claim that the speed and direction changes are smooth.

12.2 Visual Inspection

12.2.1 General Quality

We have tested different scenarios; in which we varied the desired path, starting point and noise level of the user. Overall, we see that the proposed model and implementation perform well with the exception of certain scenarios.

We can determine that the contour, largely, does not matter for the MPCC to work. This is illustrated in Figure 12.6 and Figure 12.8. This is due to that the fact that we do a cubic approximation of the contour at theta. Hence, the MPCC will optimize for a parabola rather than the exact contour which ensure it will always work. One of the downsides could be that we lose overall accuracy. However, this was not clearly observed.

Next to this we see, for instance in Figure 12.8 that the model is able to correct an error and bring the pen back smoothly towards the contour. In this scenario, we introduced an error that is big enough that the electromagnet has to move towards the pen. The electromagnet does so accordingly and brings the pen back to the contour. One could argue that this happens at too much of a right angle to be fluent and that a user would move more forward while making a correction. This is something that should be validated during the user test. Yet, it is a matter of picking the weights correctly and does not pose a direct threat to the method.

When taking the look at the perceived strength (Figure 12.6 and 12.8. We can see that the desired strength of 0.05 is relatively constant in both cases, with the exceptions of some peaks; which are unexplained. Next to this we see that the alpha value reacts to a change in distance. However, it reaches also above ”1” which is technically impossible in a real-life system.

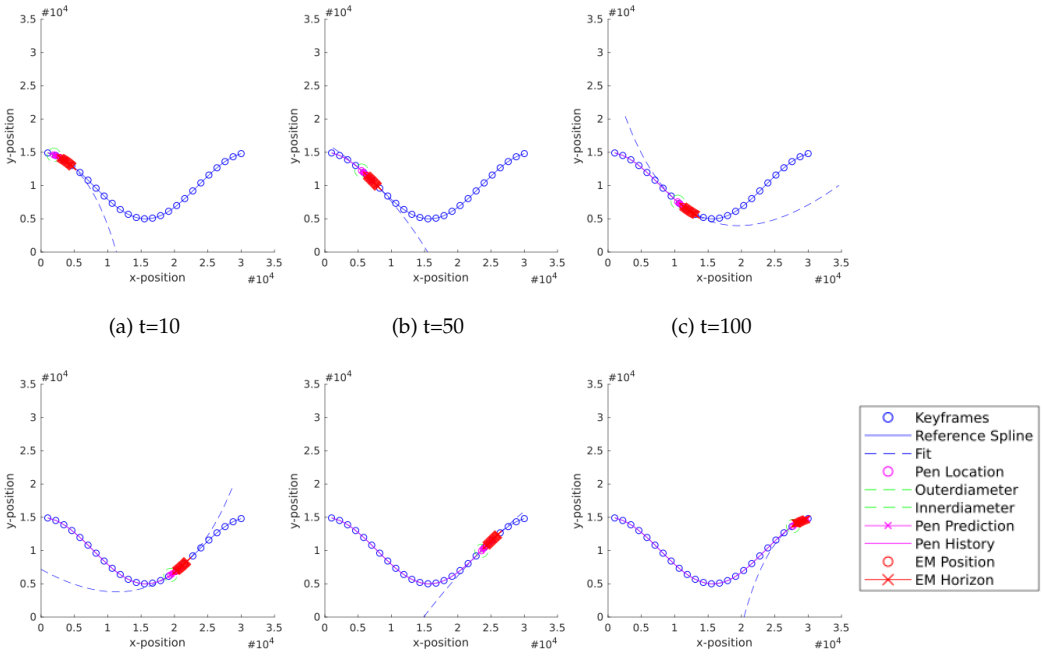
Similarly, using more complex shapes such as a sinus with a short wavelength (Figure 12.10), a spiral (Figure 12.12) or a completely free form contour (Figure 12.14) does not seem the pose any significant challenges. Again this is due to the face that we use a cubic approximation. Hence, the MPCC does not optimize for the desired path, but for the approximation of it. This leads us to believe that this method could work for a majority of desired paths.

In all of these cases the perceived strength also behaves as expected (Figures 12.10, 12.12 and 12.14). Although in these cases the Alpha value exceeds 1 by a significant amount. It is possible to make the maximum alpha value a hard constraint or increases the weight on higher values to counter this problem.

The next stage is to add noise of different levels to the user model. This is Gaussian noise on the speed vector in the user model. We will showcase one example in which the pen starts above the contour, the contour is a long sinus and there is 30% noise. The result can be seen in Figure 12.16. From this overview we can observe that even adding a large noise to the user does not significantly impact the final result with regard to position.

However, when taking a look at Figure 12.16 we observe that the large fluctuation in distance increases that vibration in perceived strength. This makes sense, since the perceived strength is largely based on the distance. Distance, in turn, is influenced by the noise level. Since the noise is

unpredictable the system can not react to this on time.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

Figure 12.6: Chronological overview of MPCC with no noise and pen starting on contour.

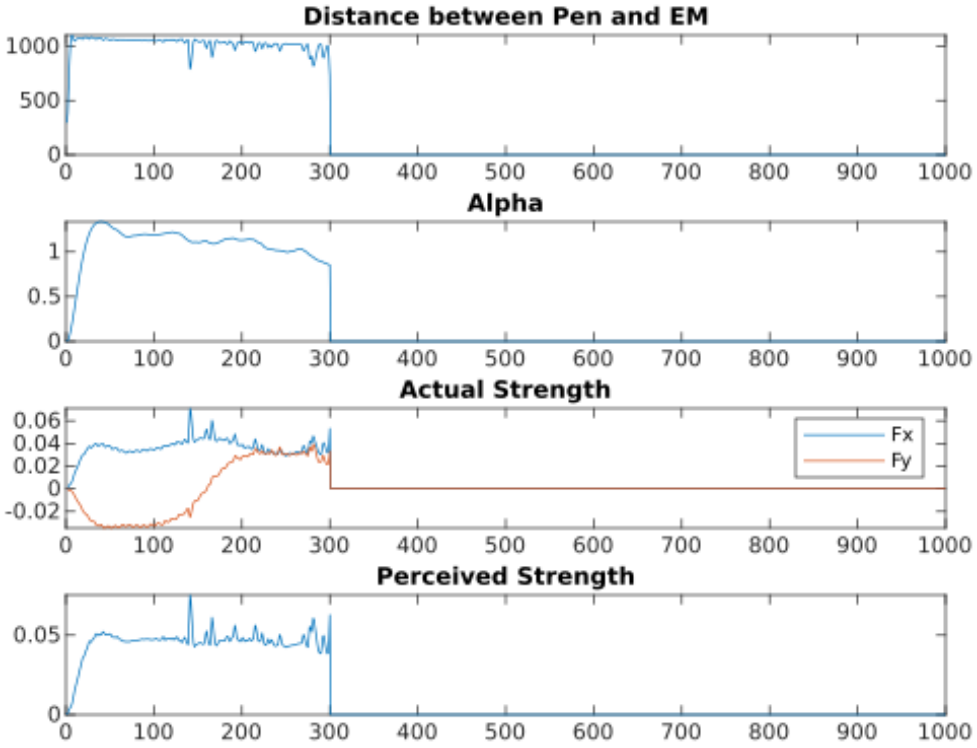
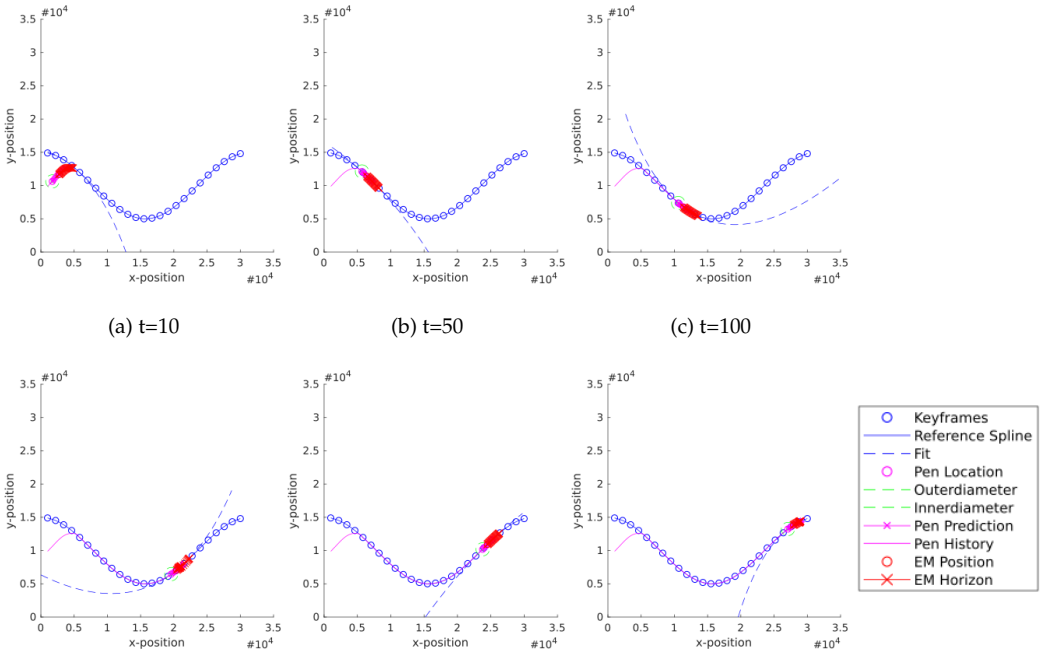


Figure 12.7: The distance, PWM input and strength for the first 300 timeframes.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

Figure 12.8: Chronological overview of MPCC with no noise and pen starting below contour.

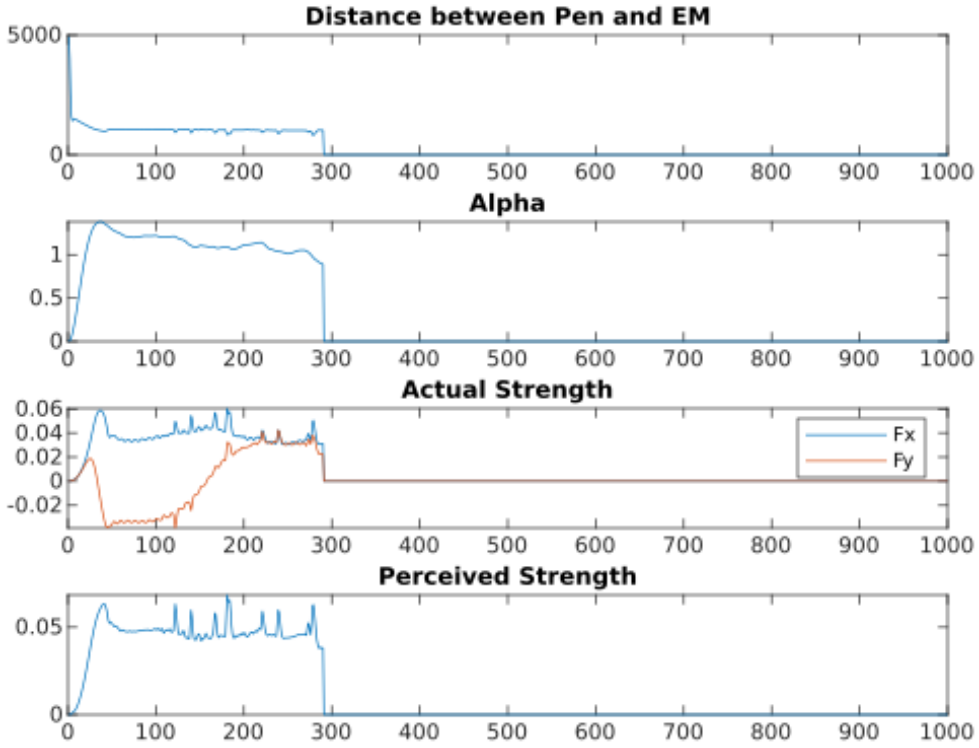
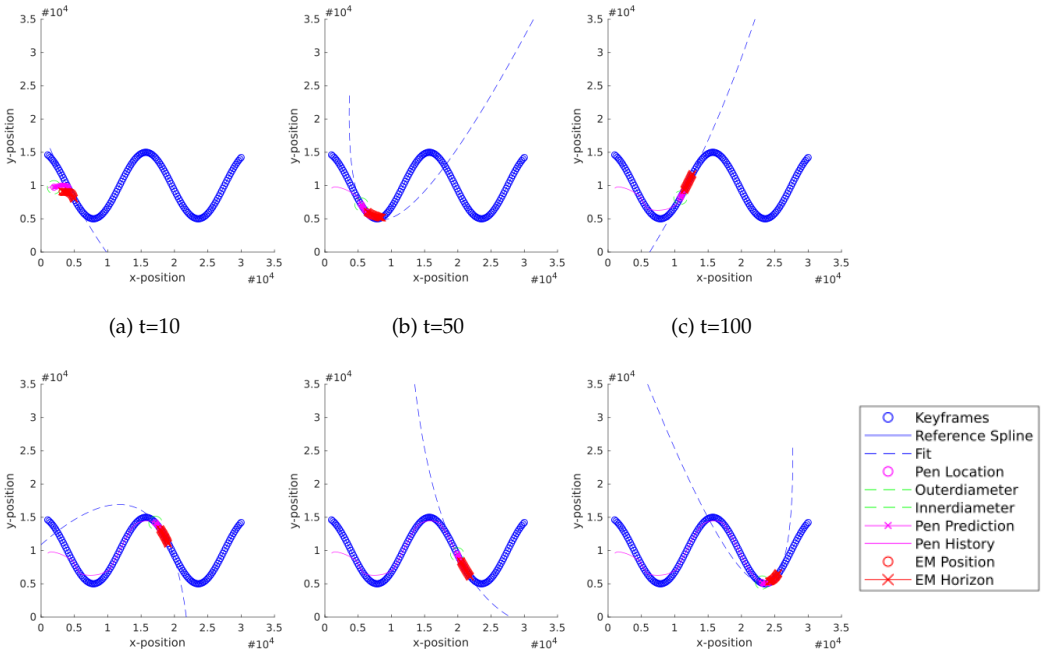


Figure 12.9: The distance, PWM input and strength for the first 300 timeframes.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

Figure 12.10: Chronological overview of MPCC with no noise and pen starting on a sinus contour.

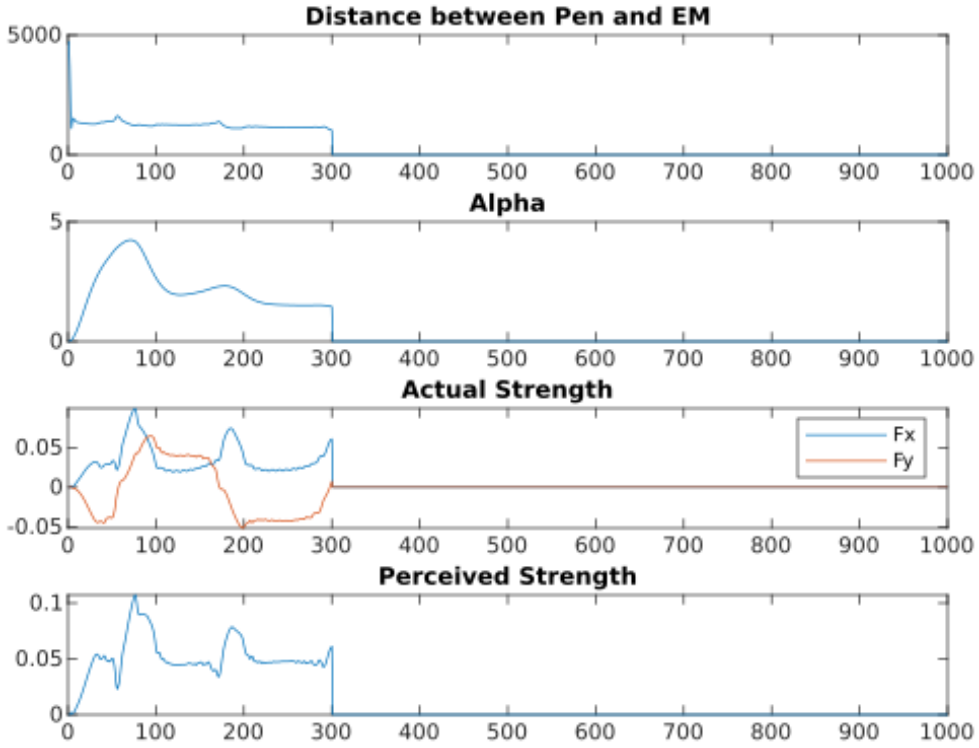
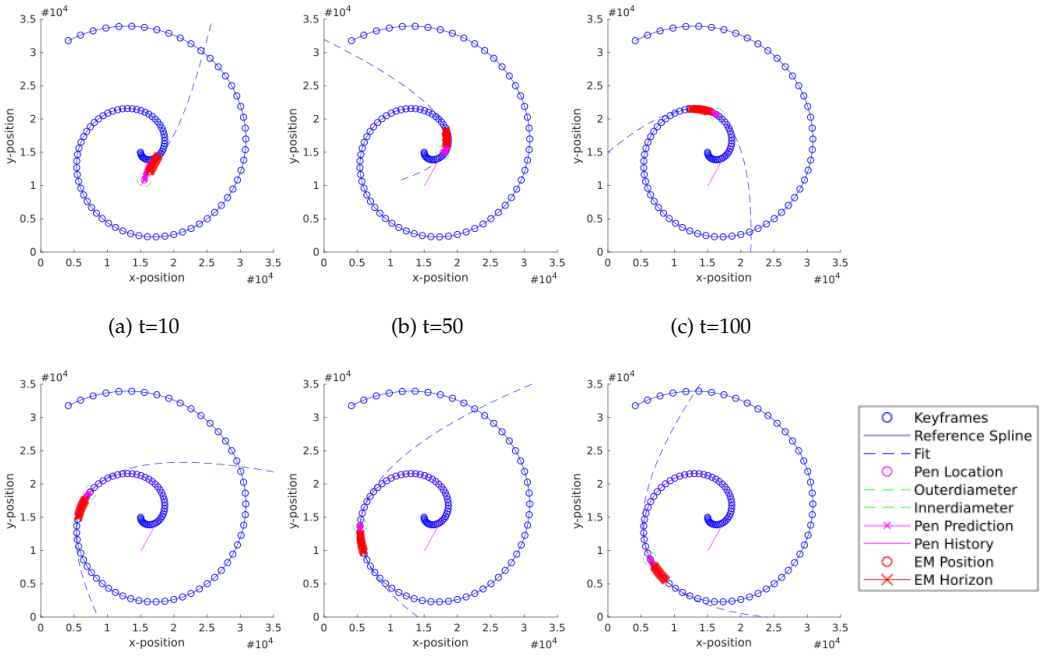


Figure 12.11: The distance, PWM input and strength for the first 300 timeframes.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

Figure 12.12: Chronological overview of MPCC with no noise and pen starting below a sinus contour.

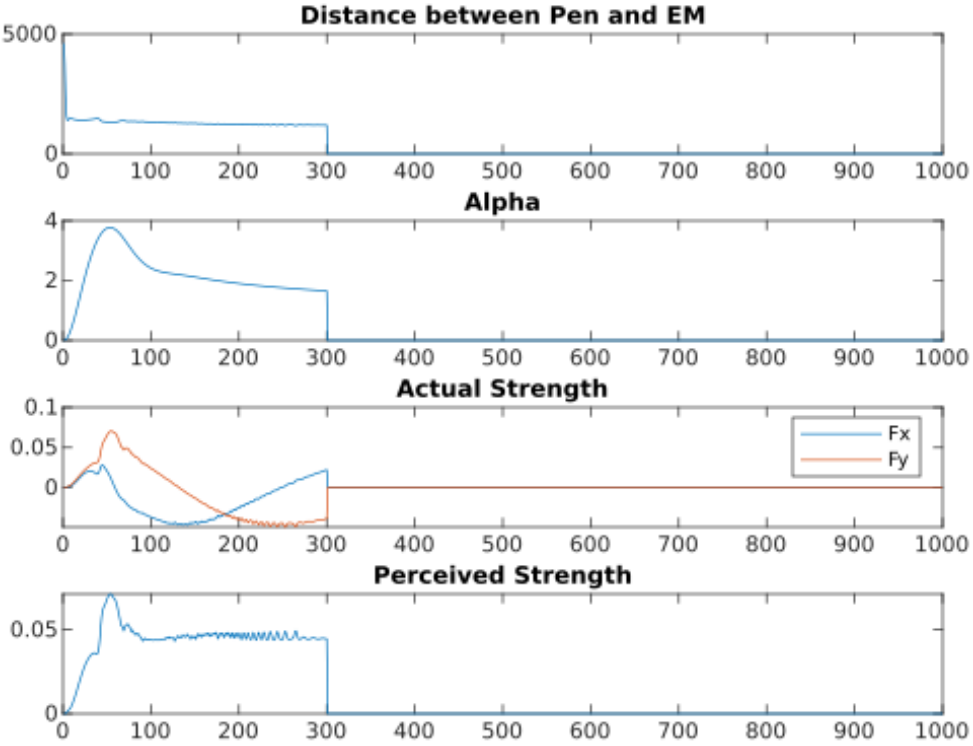
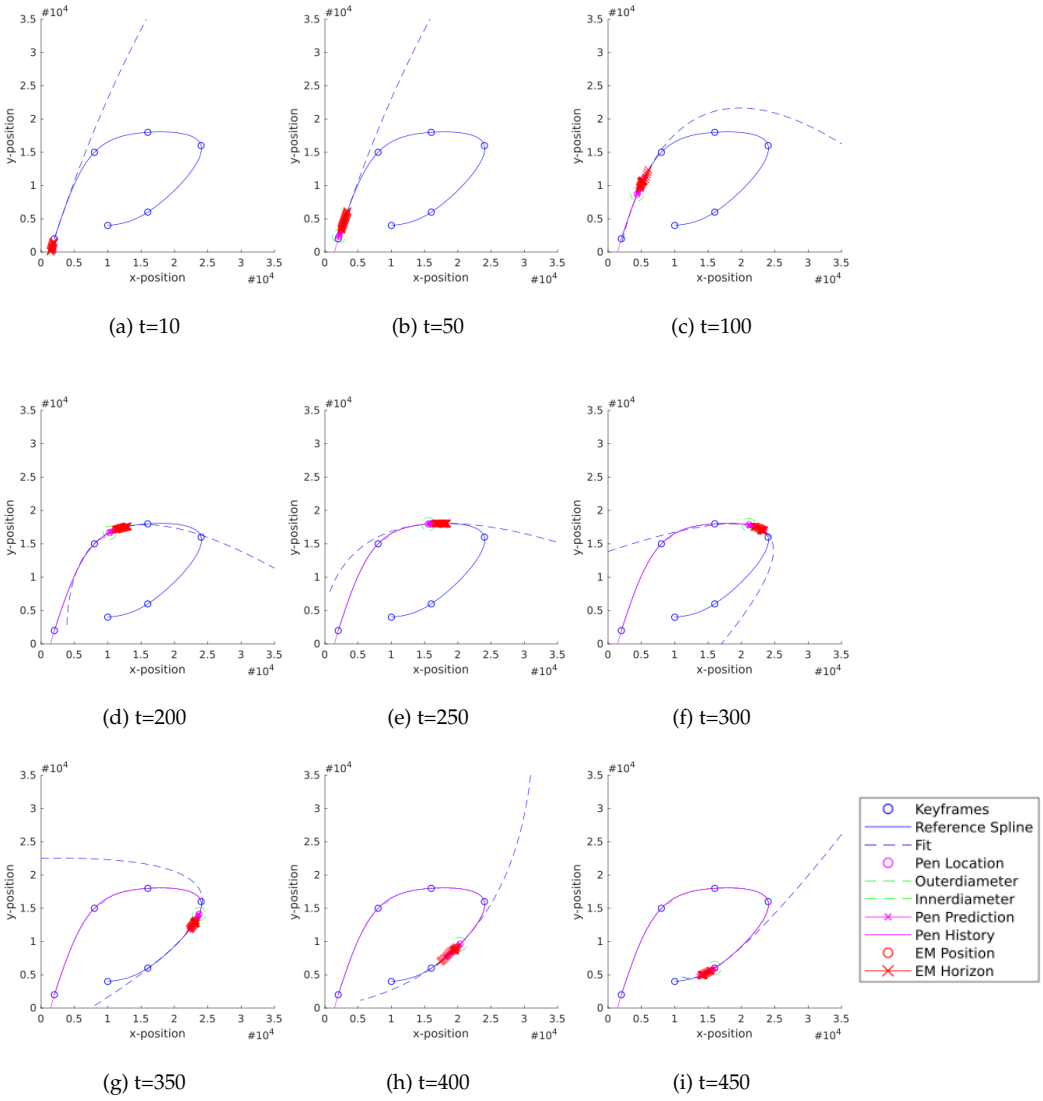


Figure 12.13: The distance, PWM input and strength for the first 300 timeframes.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

(g) t=350 (h) t=400 (i) t=450

Figure 12.14: Chronological overview of MPCC with no noise and pen starting below a spiral contour.

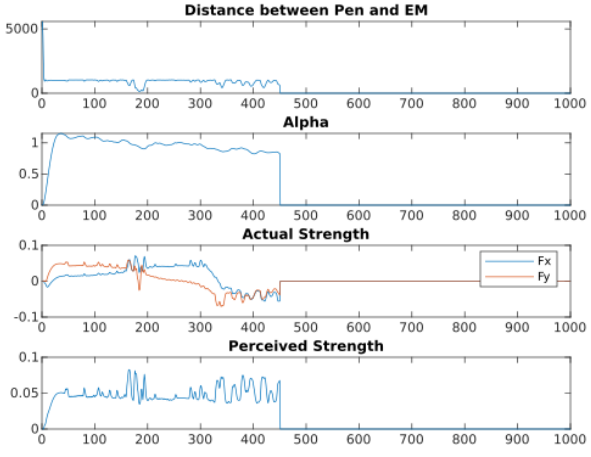
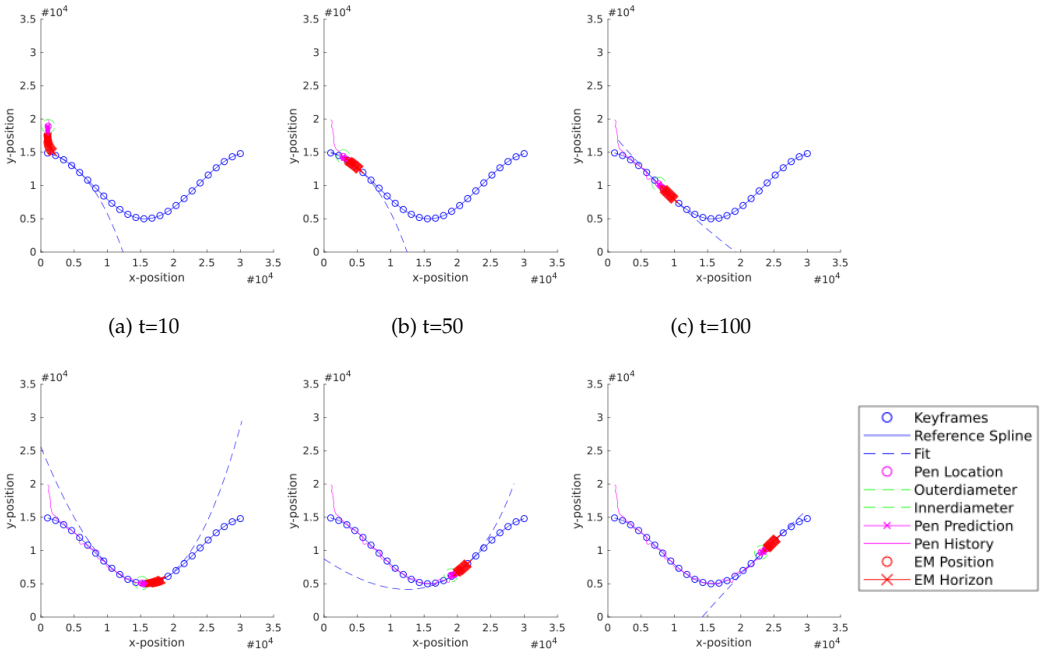


Figure 12.15: The distance, PWM input and strength for the first 450 timeframes.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

Figure 12.16: Chronological overview of MPCC with 30% noise and pen starting above a long sinus contour.

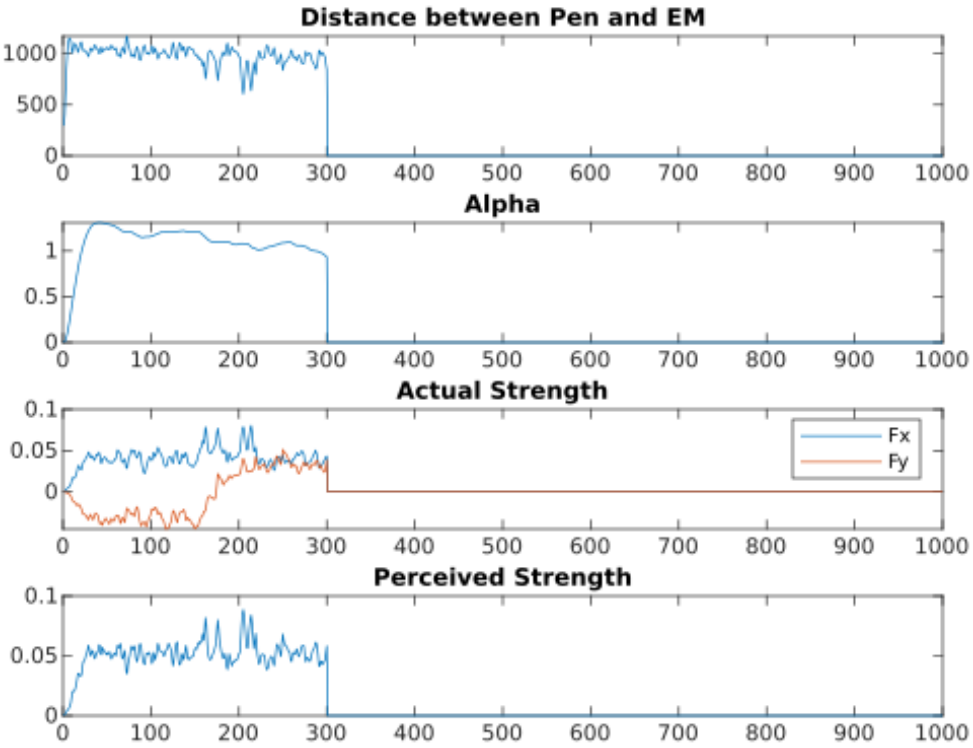


Figure 12.17: The distance, PWM input and strength for the first 300 timeframes.

12.2.2 Challenges

Irregular Electromagnet Prediction

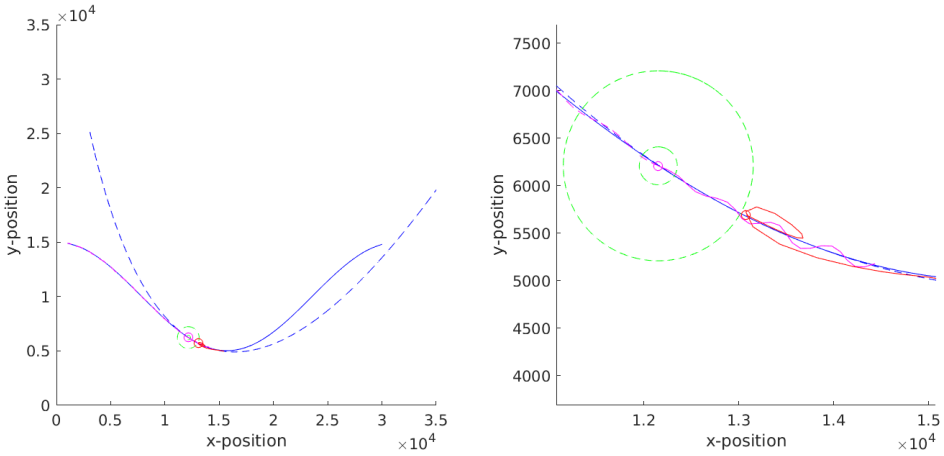


Figure 12.18: Close-up of a not optimal prediction for the electromagnet positions

A rare, but present, occurrence is that the result of the solver is not optimal. This can be seen in Figure 12.18, in which the the predicted positions of the electromagnet make little sense. One would expect a more straight line along the contour. This prediction is also immediately correct in the next time step.

It is hard to find a cause for the phenomenon with certainty. However, we hypothesize that it is due to the relative complexity of the cost function. This might make the optimizer prone to being content with a local optima. This can be solved in different manners. Theoretically, we could reduce the complexity of the cost function. However, in practice we need all the terms present. Secondly, we could try different weights. For instance, increasing the theta-progress weight could ensure that the seemingly random points are less likely to happen. However, this would reduce the relative importance of the other terms. Finally, we could make the threshold for an acceptable result more strict. However, this could increase the computational time heavily.

Difficult Scenarios

We have also seen that our proposed strategy does not work in every case. Most notably in a combination of tight corners and high noise levels (Figure 12.20 and 12.19). We think this particular behavior is due to the fact that noise plus tight corner cause the unique scenario in which the pen ends up in front of the electromagnet. This causes an interplay in which the pen starts to move backwards until it again in the acute angle in which the magnet ends up in front of the pen again, making the pen move forward through the corner again. This is likely to become a loop in which

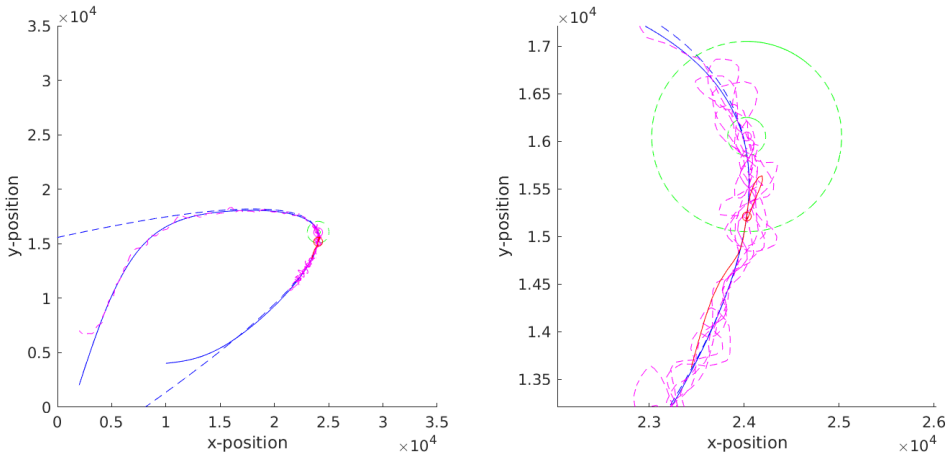
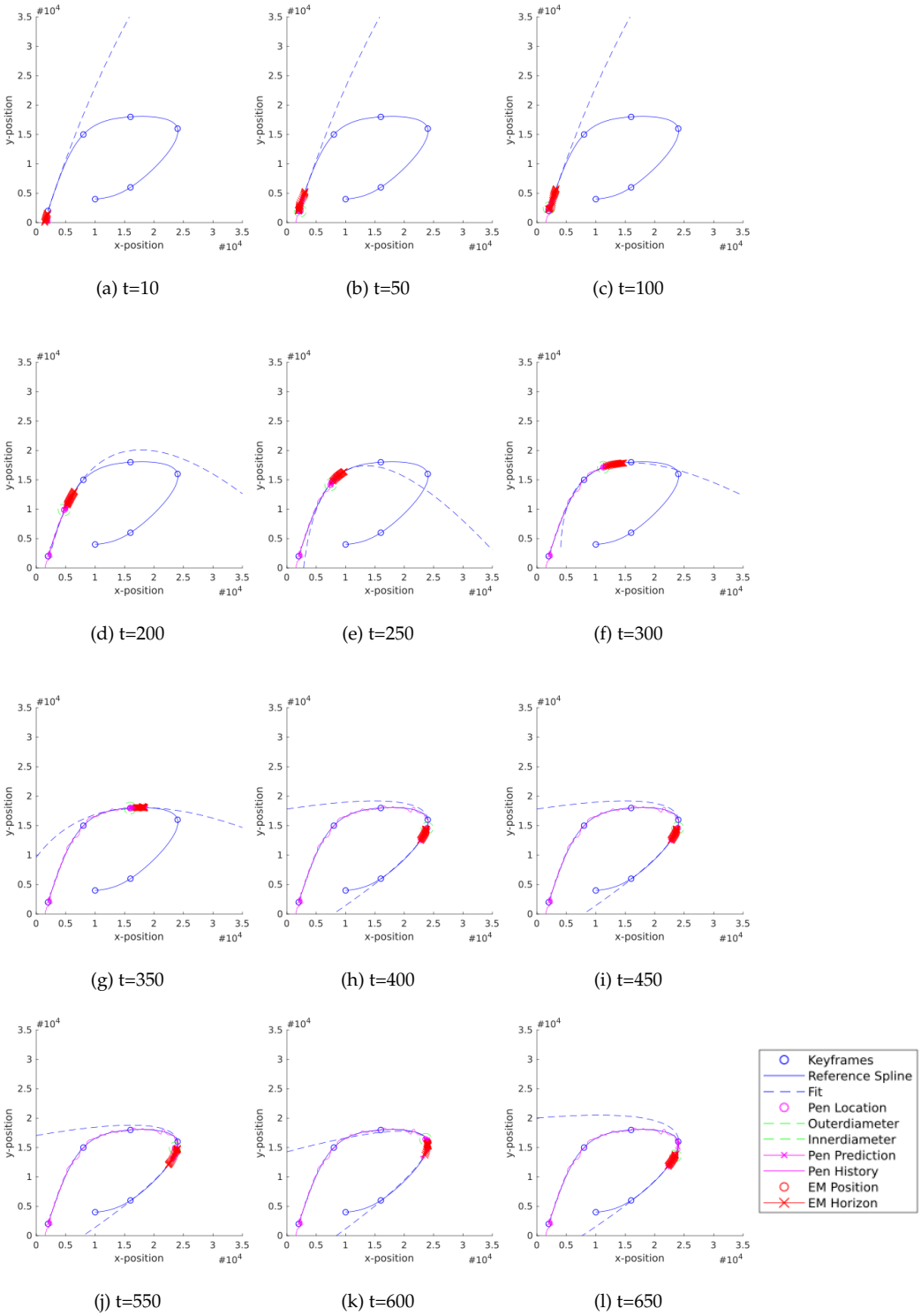


Figure 12.19: Close-up of challenging area for high noise tight corner.

the system gets stuck.

However, we also argue that this is an artifact of our simulation and user update model. We assume that a real life user will not be inclined to move back in such a fashion, yet we need to validate this assumption in our user test. Similarly, we could introduce an addition cost term into our MPCC that force the electromagnet to always be in front of the pen. This is challenging in two aspects though. First of all, we would have to define what ”in front” exactly means. Secondly, it would increase the complexity, of our already complex, cost function.



(a) t=10 (b) t=50 (c) t=100

(d) t=200 (e) t=250 (f) t=300

(g) t=350 (h) t=400 (i) t=450

(j) t=550 (k) t=600 (l) t=650

Figure 12.20: Chronological overview of MPCC with no noise and pen starting below a free form contour.

Chapter 13
Conclusion

IN this thesis a control strategy for electromagnetic induced haptic feedback was introduced, formalized and evaluated. The high-level goal was to develop a system that increases the learning-rate of drawing. This is especially relevant for stroke rehabilitation, pupils and aspiring artists. This control strategy, Model Predictive Contour Control, was evaluated in a quantitative manner, in which we took a look at computational time, as well as qualitative manner, this was a visual inspection of the result to find challenging cases. We showed that the MPCC performs well in a large variety of cases; including noise, different desired paths and unexpected events. However, the combination of high noise and a sharp corner in the desired path still posses problems. Next to this the computational time of the MPCC solver is approximately 20x too long.

Next to this we introduced a proposed hardware design that corresponds with the introduced MPCC. This includes a custom designed pen that has permanent magnets, an axis system (which is based on a 3D printer) and an electronic circuit. Within the larger scope of the full project an integration of control theory and hardware is aimed for, however that is beyond the scope of this thesis where we focus on de control theory.

Compared to already existing research we have a large focus on guidance control, rather than position control, this should help to user to make errors and therefore learn. Next to this we use a more intelligent predictive strategy than majority of the prior work, which are based on a PID control. This actually enables us to use guidance control, as well as counters oscillation that could happen in previous projects.

13.1 Research Goals

In the beginning of this thesis we formulated four different goals to be achieved in this part of the project. We will now look back at them and evaluate our achievements compared to the initial goals.

1. *Model an electromagnet with a high accuracy.*
   In Chapter 7 we introduce our analytical closed-format model for calculating the strength of an electromagnet on a permanent magnet given the Pulse Width Modulation percentage and distance in three dimensions. We finalize this chapter by comparing the results of our approximation to the state-of-the-art Finite Element Method. Despite encountering inconsistencies in the output of the FEM we can conclude that our proposal is reasonable under the condition that the electromagnet has a core.

2. *Investigate Different Control Strategies*
   Chapter 8 discusses PID, MPC and MPCC controllers from a theoretical perspective as well

as evaluates an implementation of each version. The combination of theory and implementation leads us to be able to formulate advantages and disadvantages for the different strategies. From this we were able to conclude the MPCC would be the most suitable controller type to use in this project.

3. *Propose Suitable Hardware*
   Although not largely extended on in this thesis, Chapter 9 introduces suitable hardware that is designed for specifically a MPCC controller. This design ensures that we can use acceleration as an input to the hardware, which should reduce the jerk in the system. This chapter is of high relevance to the follow up of this thesis, when the developed theory of this word has to be implemented in hardware.

4. *Formalize, Implement and Evaluate the most Relevant Control Strategy.*
   The main body of this thesis was concerned with achieving this goal. Chapter 10 discusses the formalization of the MPCC where as Chapter 11 discusses the implementation details as well as evaluation procedure. Finally, Chapter 12 shows the results of this evaluation.

From this we can conclude that we were able to achieve the goals that we set out from the beginning of this thesis. However, the contents of this thesis are not the complete project. This allows us to take a critical look at what still needs to be done in the remainder of the project.

13.2 Outlook

There are ample of elements that require further attention. One of the major issues at the moment is that the limitations of the hardware are not taken into account in the MPCC. Currently we limit the MPCC by having weights on the input, increasing these weights would ensure that the optimal inputs are always in the capabilities of the proposed hardware. However, would be never a certainty. Our implementation allows to put virtual hard-constraints on both input and states. This opportunity should be explored further. An added benefit of this would be that there is a chance that this would decrease the computational time, since it would limited the solutions space to a large extent.

Secondly, the current implementation of the user model has no clear theoretical foundation. It is currently based on a series of assumptions which seem logical. However, they were not validated. There are some alternative models that find there foundation in the theory of motor control. Kawato [24] provides an extremely elaborate overview of different types of models; including the jerk model for instance. These require future attention and could increase the overall quality of the method.

Finally, it is essential that the connection with the hardware is made since this allows us to test our proposal with real users. For this reason we already proposed a user-test. Executing this proposal would allow us that say something about the quality of the method, about the

assumptions have been made and finally, it would enable us to answer whether our proposal can increase the learning-rate of drawing.

13.3 Final Remarks

Despite the aforementioned future points, this thesis provides a unique and valuable approach to using haptic feedback for (re-)learning handwriting and drawing tasks. We hope that we can extent on this work in the near months to come and address the above mention points in order to publish at a major international conference. Overall, we are happy with till which point we got during the time-frame that we had and look-forward to continue working on this project.

Part VI

Appendix

Appendix A
Evaluation

A.1 Parameters

These parameters are used in the simulation.

Contour Fitting

Order = 2

Number of fitting points = 50

MPCC

N = 25

dt = 0.1

Maxit = 5000

Limits = [-Inf, Inf]

Number of stages = 1000

Weights

w_l=1e2

w_c=2e1

w_{em}=1

w_d=1e5

w_{close}=1e3

w_θ=1e2

$w_{\dot{\theta}}$=5e-1

$w_{\ddot{x}}$=1e-2

$w_{\ddot{y}}$=1e-2

$w_{\ddot{\alpha}}$=1e-3

$w_{\ddot{\theta}}$=1e-2

User - Update

w_a = 0.3

w_b = 0.6

w_c = 0.1

noise = 0, 10%, 30%

User - Predict

w_a = 0.5

w_c = 0.5

Bibliography

[1] Jack A Adams, Philip H Marshall, and Norman W Bray. Closed-loop theory and long-term retention. 1971.

[2] Asif Ali, Bradley Fawver, Jeffrey T Fairbrother, and Christopher M Janelle. Too much of a good thing: random practice scheduling and self-control of feedback lead to unique but not additive learning benefits. *Frontiers in psychology*, 3:503, 2012.

[3] Karl Johan Åström and Tore Hägglund. *PID controllers: theory, design, and tuning*, volume 2. Instrument society of America Research Triangle Park, NC, 1995.

[4] Olivier Bau, Ivan Poupyrev, Ali Israr, and Chris Harrison. Teslatouch: electrovibration for touch surfaces. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 283–292. ACM, 2010.

[5] William Buxton, Ralph Hill, and Peter Rowley. Issues and techniques in touch-sensitive tablet input. *ACM SIGGRAPH Computer Graphics*, 19(3):215–224, 1985.

[6] Youngjun Cho, Andrea Bianchi, Nicolai Marquardt, and Nadia Bianchi-Berthouze. Realpen: Providing realism in handwriting tasks on touch surfaces using auditory-tactile feedback. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 195–205. ACM, 2016.

[7] Jonathan Cole and Jacques Paillard. Living without touch and peripheral information about body position and movement: Studies with deafferented subjects. *The body and the self*, pages 245–266, 1995.

[8] P Cominos and N Munro. Pid controllers: recent tuning methods and design to specification. *IEE Proceedings-Control Theory and Applications*, 149(1):46–53, 2002.

[9] Max Dessoir. *Über den Hautsinn*. 1892.

[10] Alexander Domahidi and Juan Jerez. Forces pro: code generation for embedded optimization. `https://www.embotech.com/FORCES-Pro`, 2016.

[11] Pamela W Duncan, Martha Propst, and Steven G Nelson. Reliability of the fugl-meyer assessment of sensorimotor recovery following cerebrovascular accident. *Physical therapy*, 63 (10):1606–1610, 1983.

[12] Kaan Erkorkmaz and Yusuf Altintas. High speed cnc system design. part iii: high speed tracking and contouring control of feed drives. *International Journal of Machine Tools and Manufacture*, 41(11):1637–1658, 2001.

[13] Kaan Erkorkmaz and Yusuf Altintas. Quintic spline interpolation with minimal feed fluctuation. *Journal of Manufacturing Science and Engineering*, 127(2):339–349, 2005.

[14] David S Finley and Ninh T Nguyen. Surgical robotics. *Current surgery*, 62(2):262–272, 2005.

[15] Antonio Frisoli, Fabrizio Rocchi, Simone Marcheschi, Andrea Dettori, Fabio Salsedo, and Massimo Bergamasco. A new force-feedback arm exoskeleton for haptic interaction in virtual environments. In *Eurohaptics Conference, 2005 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. World Haptics 2005. First Joint*, pages 195–201. IEEE, 2005.

[16] James Jerome Gibson. The senses considered as perceptual systems. 1966.

[17] Kelly S Hale and Kay M Stanney. Deriving haptic design guidelines from human physiological, psychophysical, and neurological foundations. *IEEE Computer Graphics and Applications*, 24(2):33–39, 2004.

[18] Joseph Hidler, Diane Nichols, Marlena Pelliccio, Kathy Brady, Donielle D Campbell, Jennifer H Kahn, and T George Hornby. Multicenter randomized clinical trial evaluating the effectiveness of the lokomat in subacute stroke. *Neurorehabilitation and neural repair*, 23(1): 5–13, 2009.

[19] Britta Husemann, Friedemann Müller, Carmen Krewer, Silke Heller, and Eberhardt Koenig. Effects of locomotion training with assistance of a robot-driven gait orthosis in hemiparetic patients after stroke: a randomized controlled pilot study. *Stroke*, 38(2):349–354, 2007.

[20] Hiroo Iwata. Pen-based haptic virtual environment. In *Virtual Reality Annual International Symposium, 1993., 1993 IEEE*, pages 287–292. IEEE, 1993.

[21] Yvonne Jansen, Thorsten Karrer, and Jan Borchers. Mudpad: tactile feedback and haptic texture overlay for touch surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 11–14. ACM, 2010.

[22] Roland S Johansson, Charlotte Häger, and Ronald Riso. Somatosensory control of precision grip during unpredictable pulling loads. *Experimental brain research*, 89(1):192–203, 1992.

[23] Leonard E Kahn, Michele L Zygman, W Zev Rymer, and David J Reinkensmeyer. Robot-assisted reaching exercise promotes arm movement recovery in chronic hemiparetic stroke: a randomized controlled pilot study. *Journal of neuroengineering and rehabilitation*, 3(1):12, 2006.

[24] Mitsuo Kawato. Internal models for motor control and trajectory planning. *Current opinion in neurobiology*, 9(6):718–727, 1999.

[25] Hyunjung Kim, Seoktae Kim, Boram Lee, Jinhee Pak, Minjung Sohn, Geehyuk Lee, and Woohun Lee. Digital rubbing: playful and intuitive interaction technique for transferring a graphic image onto paper with pen-based computing. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pages 2337–2342. ACM, 2008.

[26] Ki-Uk Kyung, Jun-Young Lee, and Junseok Park. Haptic stylus and empirical studies on braille, button, and texture display. *BioMed Research International*, 2008, 2008.

[27] Ki-Uk Kyung, Jun-Young Lee, and Mandayam A Srinivasan. Precise manipulation of gui on a touch screen with haptic cues. In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pages 202–207. IEEE, 2009.

[28] Denise Lam. *A model predictive approach to optimal path-following and contouring control*. PhD thesis, 2012.

[29] Denise Lam, Chris Manzie, and Malcolm Good. Model predictive contouring control. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 6137–6142. IEEE, 2010.

[30] Peter Langhorne, Fiona Coupar, and Alex Pollock. Motor recovery after stroke: a systematic review. *The Lancet Neurology*, 8(8):741–754, 2009.

[31] Johnny C Lee, Paul H Dietz, Darren Leigh, William S Yerazunis, and Scott E Hudson. Haptic pen: a tactile feedback stylus for touch screens. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*, pages 291–294. ACM, 2004.

[32] Rui Loureiro, Farshid Amirabdollahian, Susan Coote, Emma Stokes, and William Harwin. Using haptics technology to deliver motivational therapies in stroke patients: Concepts and initial pilot studies. In *Proceedings of EuroHaptics*, volume 2001, page 6, 2001.

[33] Peter S Lum, Charles G Burgar, Peggy C Shor, Matra Majmundar, and Machiel Van der Loos. Robot-assisted movement training compared with conventional therapy techniques for the rehabilitation of upper-limb motor function after stroke. *Archives of physical medicine and rehabilitation*, 83(7):952–959, 2002.

[34] Laura Marchal-Crespo, Stephanie McHughen, Steven C Cramer, and David J Reinkensmeyer. The effect of haptic guidance, aging, and initial skill level on motor learning of a steering task. *Experimental brain research*, 201(2):209–220, 2010.

[35] Stefano Masiero, Andrea Celia, Giulio Rosati, and Mario Armani. Robotic-assisted rehabilitation of the upper limb after acute stroke. *Archives of physical medicine and rehabilitation*, 88 (2):142–149, 2007.

[36] Rochdi Merzouki, Arun Kumar Samantaray, Pushparaj Mani Pathak, and Belkacem Ould Bouamama. *Intelligent mechatronic systems: modeling, control and diagnosis*. Springer Science & Business Media, 2012.

[37] Gérard Meunier. *The finite element method for electromagnetic modeling*, volume 33. John Wiley & Sons, 2010.

[38] Matjaz Mihelj, Janez Podobnik, and Marko Munih. Henrie-haptic environment for reaching and grasping exercise. In *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*, pages 907–912. IEEE, 2008.

[39] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[40] James Mullins, Christopher Mawson, and Saeid Nahavandi. Haptic handwriting aid for training and rehabilitation. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 3, pages 2690–2694. IEEE, 2005.

[41] Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. Real-time planning for automated multi-view drone cinematography. 2017.

[42] Tobias Nägeli, Lukas Meier, Alexander Domahidi, Javier Alonso-Mora, and Otmar Hilliges. Real-time planning for automated multi-view drone cinematography. *ACM Transactions on Graphics (TOG)*, 36(4):132, 2017.

[43] Carl Nave. Pid controller. `http://hyperphysics.phy-astr.gsu.edu/hbase/Solids/hyst.html`, 2017. Accessed: 2018-14-03.

[44] Gian Pangaro, Dan Maynes-Aminzade, and Hiroshi Ishii. The actuated workbench: computer-controlled actuation in tabletop tangible interfaces. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*, pages 181–190. ACM, 2002.

[45] Ivan Poupyrev, Makoto Okabe, and Shigeaki Maruyama. Haptic feedback for pen computing: directions and strategies. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1309–1312. ACM, 2004.

[46] S Joe Qin and Thomas A Badgwell. A survey of industrial model predictive control technology. *Control engineering practice*, 11(7):733–764, 2003.

[47] Vijay Rajanna, Patrick Vo, Jerry Barth, Matthew Mjelde, Trevor Grey, Cassandra Oduola, and Tracy Hammond. Kinohaptics: An automated, wearable, haptic assisted, physio-therapeutic system for post-surgery rehabilitation and self-care. *Journal of medical systems*, 40(3):60, 2016.

[48] Gabriel Robles-De-La-Torre. The importance of the sense of touch in virtual and real environments. *Ieee Multimedia*, 13(3):24–30, 2006.

[49] Gabriel Robles-De-La-Torre. Principles of haptic perception in virtual environments. In *Human haptic perception: Basics and applications*, pages 363–379. Springer, 2008.

[50] Gabriel Robles-De-La-Torre and Vincent Hayward. Force can overcome object geometry in the perception of shape through active touch. *Nature*, 412(6845):445, 2001.

[51] David A Rosenbaum. *Human motor control*. Academic press, 2009.

[52] Robert A Scheidt, David J Reinkensmeyer, Michael A Conditt, W Zev Rymer, and Ferdinando A Mussa-Ivaldi. Persistence of motor adaptation during constrained, multi-joint, arm movements. *Journal of Neurophysiology*, 84(2):853–862, 2000.

[53] Henning Schmidt, Stefan Hesse, Rolf Bernhardt, and Jörg Krüger. Hapticwalker—a novel haptic foot device. *ACM Transactions on Applied Perception (TAP)*, 2(2):166–180, 2005.

[54] Richard A Schmidt. A schema theory of discrete motor skill learning. *Psychological review*, 82 (4):225, 1975.

[55] Richard A Schmidt, Tim Lee, Carolee Winstein, Gabriele Wulf, and Howard Zelaznik. *Motor Control and Learning, 6E*. Human kinetics, 2018.

[56] Richard Shusterman. Muscle memory and the somaesthetic pathologies of everyday life. *Human Movement*, 12(1):4–15, 2011.

[57] Roland Sigrist, Georg Rauter, Robert Riener, and Peter Wolf. Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review. *Psychonomic bulletin & review*, 20(1):21–53, 2013.

[58] Unknown. electromagnets. `http://www.ece.neu.edu/fac-ece/nian/mom/electromagnets.html`, Unknown. Accessed: 2018-27-02.

[59] Unknown. Pid controller, Unknown.

[60] OAJ Van der Meijden and MP Schijven. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical endoscopy*, 23(6):1180–1190, 2009.

[61] Liuping Wang. *Model predictive control system design and implementation using MATLAB®*. Springer Science & Business Media, 2009.

[62] Charles Warlow, Jan Van Gijn, Martin S Dennis, Joanna M Wardlaw, Peter AG Sandercock, Gabriel Rinkel, and Peter Langhorne. Stroke: practical management. 2008.

[63] Malte Weiss, Chat Wacharamanotham, Simon Voelker, and Jan Borchers. Fingerflux: near-surface haptic feedback on tabletops. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 615–620. ACM, 2011.

[64] Anusha Withana, Makoto Kondo, Yasutoshi Makino, Gota Kakehi, Maki Sugimoto, and Masahiko Inami. Impact: Immersive haptic stylus to enable direct touch and manipulation for surface computing. *Computers in Entertainment (CIE)*, 8(2):9, 2010.

[65] Junichi Yamaoka and Yasuaki Kakehi. depend: augmented handwriting system using ferromagnetism of a ballpoint pen. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 203–210. ACM, 2013.

[66] Shunsuke Yoshida, Haruo Noma, and Kenichi Hosaka. Proactive desk ii: Development of a new multi-object haptic display using a linear induction motor. In *Virtual Reality Conference, 2006*, pages 269–272. IEEE, 2006.