University of Twente

EEMCS / Electrical Engineering
*Control Engineering*

# Image-based Magnetic Control of Microparticles

Jasper Keuning

MSc report

**Committee:**
Prof. dr. ir. S. Stramigioli
Dr. S. Misra
Dr.ir. L. Abelmann
Dr. R.K. Truckenmüller

December 2011

Report nr. 023CE2011
Control Engineering
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

# Abstract

The use of microscopic robots for surgery or drug delivery is a very promising field of research. Being able to control exactly where drugs are delivered in the human body, allows for targeted treatment of various illnesses. Because the treatment can be localized, the medication no longer affects the whole body, thus possibly reducing side-effects. In the first part of this work, we describe the design and implementation of a setup that can manipulate superparamagnetic spherical particles. The particles used have an average diameter of 100 µm. Using a microscope with a camera attached, we determined the location of the particle. Using the magnetic fields generated by a set of four coils and a controller using the image data as feedback, we were able to position the particle within 8.5 µm of a given setpoint while achieving speeds of 235 µm $s^{-1}$. Also we were able to track a circular and figure-eight shaped path by providing the controller with a sequence of waypoints.

This controller did not function well when tracking a multi-segment path. The motion along the path was not smooth and on some occasions the particle had to go back to a certain waypoint it had missed. Therefore a more specific controller needed to be designed.

In the second part of this work we describe a control algorithm that allows the particle to track a straight line. Combined with an improved waypoint switching algorithm, the particle can track a multi-segmented path accurately and the motion is smooth. In addition to the controller a potential field motion planning algorithm has been implemented to maneuver the particle towards a target while avoiding obstacles. Also a grid based least cost path planner has been implemented that pre-computes the least cost path towards a target and generates the necessary path segments for the controller to maneuver the particle towards the target around all obstacles.

# Contents

# Introduction

Minimally invasive surgery (MIS) aims to reduce patient surgical trauma while enabling clinicians to reach deep-seated locations within the human body. Next to the reduced trauma, patients may spend less time in the hospital saving time and money. A good example of minimally invasive surgery used today is the so-called laparoscopic surgery. Here the instruments needed for surgery are inserted in the body through small incisions in the body and the operation is done using images taken by a camera attached to the surgical instruments. Figure 1 shows one such surgical procedures[1].

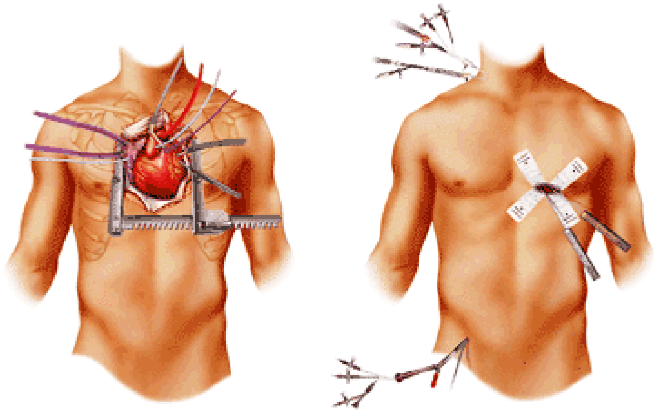


Figure 1: The left image shows the traditional open heart surgery. The right image shows the (MIS) heart surgery, performed using laparoscopic instruments.

The use or robots in MIS aims to benefit medicine by further reducing invasiveness of MIS and enable treatment of previously inoperable patients. Such robotic systems can be used to accurately guide needles to a given location in the body, or the whole robot may be inserted into the body. Once inside, it can use natural pathways within the body, such as arteries and veins or the gastrointestinal tract, to reach its target for treatment, drug delivery or diagnosis. Further reducing the size of these robots will increase their potential penetration depth inside the body. The smaller size means they will be able to travel through smaller pathways to reach their target. This work focuses on the manipulation of these small robots.

## Thesis outline

This thesis consists of two parts. The first part describes the design and implementation of a setup that can be used to manipulate small magnetic robots using magnetic fields. These robots can either be manipulated using the forces generated by the field, or they can be self-propelled where the magnetic field is used to manipulate their orientation. Some preliminary results are presented using this setup to manipulate a small 100 µm particle. The second part describes the design of a control algorithm to improve the results of controller implemented in the first part. Also some practical obstacle avoidance schemes are implemented to demonstrate the capabilities of the newly implemented controller. The thesis concludes with conclusions and recommendations for future work on the magnetic manipulation of even smaller objects.

[1]Picture courtesy: Inova Heart and Vascular Institute

# Thesis Contributions

[1] J.D.Keuning, J. de Vries, L. Abelmann andS. Misra, "Image-based magnetic control of paramagnetic microparticles in water", *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, San Francisco, USA, pp. 421 - 426, Sep-Oct 2011.

# Part I

## Image-Based Magnetic Control of Paramagnetic Microparticles in Water

# Image-Based Magnetic Control of Paramagnetic Microparticles in Water

Jasper D. Keuning[*], Jeroen de Vries[†], Leon Abelmann[†] and Sarthak Misra[*]
University of Twente, The Netherlands

*Abstract*— **This paper describes the design of a system for controlling the position of spherical paramagnetic microparticles that have an average diameter of $100\,\mu m$. The focus of this study lies in designing and implementing a system that uses microscopic images and electromagnets. Preliminary experiments have been done to verify the feasibility of the system to track and control the position of these particles. A vibrating sample magnetometer was used to determine the magnetic moment of the particles. Finite element method simulations were used to verify the magnetic behavior of the designed setup. The system was used to position the particles within $8.4\,\mu m$ of a setpoint, achieving speeds of up to $235\,\mu m\,s^{-1}$. We also demonstrated that the particle could follow a circular and a figure-eight path.**

## I. INTRODUCTION

Minimally invasive surgery (MIS) focuses on reducing patient surgical trauma while enabling clinicians to reach deep-seated locations within the human body. Further, robotic MIS aims to benefit medicine by further reducing invasiveness of MIS, improve clinical outcomes, minimize patient trauma, and enable treatment of inoperable patients. Such robotic systems can be inserted into the body, and use natural pathways within the body, such as arteries and veins or the gastrointestinal tract, to reach their target for drug delivery or diagnosis [1]. Reducing the size of these robots will increase their potential penetration depth inside the body. The smaller size means they will be able to travel through smaller pathways to reach their target.

A significant amount of research has been done on miniaturizing robots for a variety of applications [2]. An important aspect of these robotic systems is their propulsion mechanism. When robots are scaled down, the space available to store energy for propulsion within the robot reduces. In most microrobots the energy needed to move around is provided externally. There are many different ways propulsion energy can be delivered to the robot. Within the domain of untethered microrobotic systems, a widely used method to deliver this energy, is by means of (electro)magnetic fields [3].

There are several ways magnetic fields can be utilized to control microrobotic systems. Kummer et al. [4] demonstrated a system using eight electromagnets to control a robot. They showed that this setup can position a $500\,\mu m$ soft magnetic body using both open and closed loop control. Similar systems that use magnetic forces to position hard magnetic [5], soft magnetic [6], [7] and paramagnetic bodies [8] have been presented in literature. A helical-shaped robot can be propelled forward by rotating around its axis. This rotation can be achieved by periodically changing the rotation of the magnetic field at the location of the robot [9], [10].

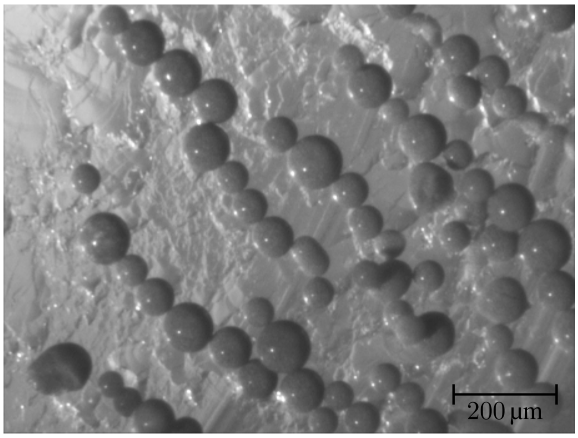



Fig. 1. The paramagnetic particles used in the experiments. The particles have diameters ranging from $60\,\mu m$ to $110\,\mu m$.

The use of magnetic fields to provide energy to the robots is however extremely inefficient. Magnetic coil systems for propulsion generate alternating fields in large volumes of at least several cubic centimeters. Out of that field, only a tiny fraction of at most a cubic millimeter is used by the robot, resulting in efficiencies smaller than $1 \times 10^{-3}$. Rather than extracting energy from the field, the robots can extract energy from the liquid into which they are residing. A static magnetic field can then be used for steering only, which requires far less energy than an alternating field. Martel and Mohammadi [11] demonstrated the possibility to use magnetotactic bacteria to position microscopic structures for self-assembly purposes. The bacteria provide the propulsion whereas the magnetic field is used to guide the bacteria to the desired positions.

Alternatively, one can use catalytic motors that extract energy from their surroundings. The energy for the propulsion comes from a chemical reaction between material that is part of the robot and the fluid it is immersed in [12]–[15].

In this paper, we describe the implementation of a compact experimental setup that will be used to study the control of self-propelled microrobots. We present the design of the setup and demonstrate its capabilities by doing preliminary experiments on the closed-loop control of paramagnetic microparticles (Fig. 1). Similar setups have been designed by the previously mentioned researchers, though the results obtained from our work will serve as a foundation for future

[*]MIRA–Institute of Biomedical Technology and Technical Medicine
[†]MESA+ Institute for Nanotechnology

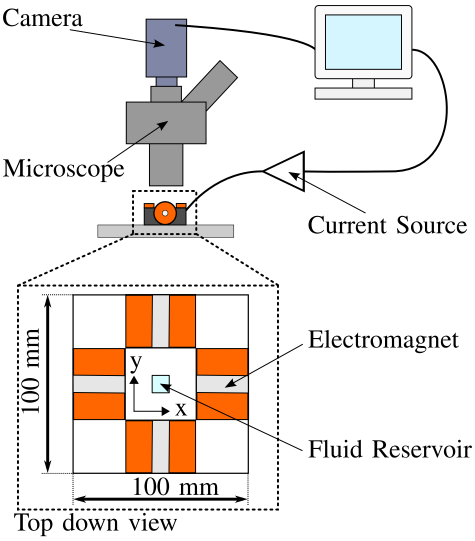



Fig. 2. The camera takes images of the particles in the reservoir using a microscope. The images are processed and control signals for the electromagnets are generated. The size of the fluid reservoir is 10 mm×10 mm.

work on self-propelled microrobots. The paper is organized as follows: Section II describes the various aspects involving the design of the setup, the tracking and the control of the paramagnetic microparticles. Section III discusses the experimental results obtained. Finally, Section IV provides conclusions and possible directions for future work.

## II. EXPERIMENTAL SETUP

The proposed setup to observe and control the position of the particles under the influence of magnetic fields is shown in Fig. 2. The main components are a fluid reservoir for the paramagnetic particles and the magnets surrounding this reservoir. A Sony XCD-X710 (Sony Corporation, Tokyo, Japan) 1024×768 pixels FireWire camera is mounted on a Mitutoyo FS70 microscope unit (Mitutoyo, Kawasaki, Japan) using a Mitutoyo M Plan Apo 2× / 0.055 Objective. A control computer is used to acquire the images and track the particles, as well as to control their position by means of Proportional-Integral (PI) controllers. The input provided to the controllers is the difference between the current location, and the desired position of the particle. The output of the controllers is used to set the current through the coils.

### A. General design considerations

The reservoir needs to be viewable underneath the microscope. There should be space available for the microscope objective to be positioned properly above the target. Also since the setup needs to be placed underneath the microscope, the total footprint is limited in its size. With the available microscope setup, the camera will have a field of view of 2.4 mm × 1.8 mm (2.34 µm per pixel). The size of the reservoir will be equal or larger then the field of view. The particles that are going to be controlled will be floating in the water-to-air boundary layer. Near the edges of the reservoir it is expected that due to surface tension, the surface of the liquid will form a meniscus. Increasing the size of the reservoir will reduce the magnitude of this meniscus near its center. Based on these considerations we have decided to design a setup with a footprint of 100 mm × 100 mm. The fluid reservoir will be 10 mm × 10 mm with a depth of 5 mm.

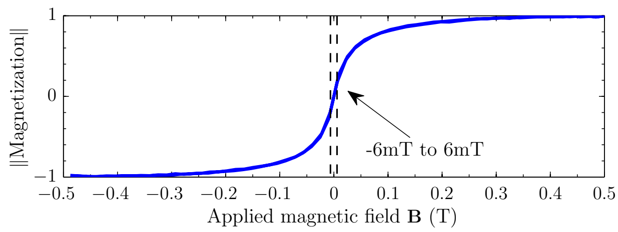



Fig. 3. Normalized mass magnetization of the paramagnetic particles.

### B. Paramagnetic microparticles

We are using paramagnetic spherical microparticles, consisting of iron-oxide in a poly(lactic acid) matrix (PLA-Particles-M-redF-plain from Micromod Partikeltechnologie GmbH, Rostock-Warnemuende, Germany). These particles have an average diameter of ∼100 µm. Paramagnetic particles are preferred over ferromagnetic particles because they were readily available and because they have a more favorable magnetization curve. We assume that the only force opposing the magnetic force is the viscous drag force. The velocity of the particle in the water is therefore determined by the balance between the magnetic force and viscous drag force.

The magnetic force exerted on a paramagnetic microparticle can be calculated using the following equation [8]:

$$\mathbf{F} = \nabla(\mathbf{m} \cdot \mathbf{B}) \tag{1}$$

where $\mathbf{m}$ is the magnetic moment of the particle and $\mathbf{B}$ is the applied magnetic field. For a paramagnetic microparticle, $\mathbf{m}$ can be expressed as,

$$\mathbf{m} = \alpha_{\mathrm{p}} V_{\mathrm{p}} \mathbf{B} \tag{2}$$

where $V_{\mathrm{p}}$ is the volume of the particle. The value of $\alpha_{\mathrm{p}}$ was determined by measuring $\mathbf{m}$ of the particles as a function of $\mathbf{B}$ in a vibrating sample magnetometer (Fig. 3). At the expected field of 6 mT, $\alpha_{\mathrm{p}}$ can be approximated by a constant. According to the manufacturer, the saturation mass magnetization of the particle is $6.6 \times 10^{-3}\,\mathrm{A\,m^2\,g^{-1}}$ and the density is $1.4 \times 10^{6}\,\mathrm{g\,m^{-3}}$. From this we obtain the saturation magnetization ($9.24\,\mathrm{kA\,m^{-1}}$), resulting in an $\alpha_{\mathrm{p}}$ of $185\,\mathrm{kA^2\,N^{-1}}$. We focus on determining the force contribution of a single coil on a particle. We assume the particle to be a perfect sphere with a radius $r_{\mathrm{p}}$. Combining (1) and (2) results in:

$$\mathbf{F} = \alpha_{\mathrm{p}} \frac{4}{3} \pi r_{\mathrm{p}}^{3} \nabla(B^{2}) \tag{3}$$

In order to determine the drag force on the particle, we first determine the Reynolds number,

$$Re = \frac{2 \rho v r_{\mathrm{p}}}{\eta} \tag{4}$$

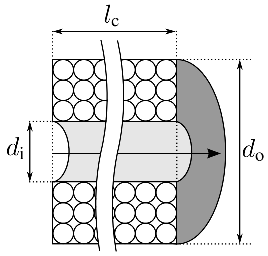



Fig. 4. Schematic of a coil used in the experimental setup. The arrow indicates the axial axis of the coil.

where $v$ is the velocity of the particle, $\eta$ dynamic viscosity ($1\,\mathrm{mPa\,s}$) and $\rho$ the density of water ($998.2\,\mathrm{kg\,m^{-3}}$), both at room temperature. Assuming that $v$ will not exceed $1\,\mathrm{mm\,s^{-1}}$, we find a Reynolds number less than 0.1 for a particle with a diameter of $100\,\mathrm{\mu m}$. We can therefore assume laminar flow conditions, and use the Stokes law for drag force $F_\mathrm{d}$ [2],

$$F_\mathrm{d} = -6\pi\eta r_\mathrm{p} v \tag{5}$$

The maximum velocity of a particle is reached when the drag force equals the magnetic force. In order to estimate the expected particle velocities, we combine (3) and (5). The maximum velocity of a paramagnetic particle ($v_\mathrm{m}$) will be,

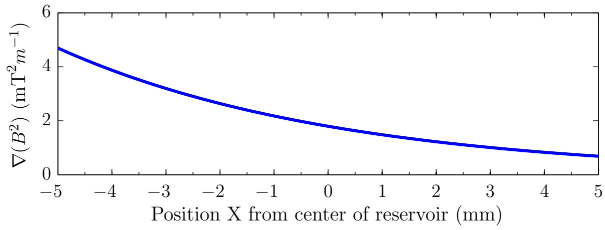
$$v_\mathrm{m} = \frac{2}{9}\frac{\alpha_\mathrm{p} r_\mathrm{p}^2}{\eta}\nabla(B^2) \tag{6}$$

The above holds for $Re < 0.1$.

### C. Coils

In order to generate the magnetic field needed to move the particles, we use four equally sized coils. Using the constraints given in Section II-A, we have chosen coils with an inner diameter $d_\mathrm{i} = 10\,\mathrm{mm}$, an outer diameter $d_\mathrm{o} = 39\,\mathrm{mm}$ and a length of $l_\mathrm{c} = 30\,\mathrm{mm}$. The coils have 1680 turns of $0.5\,\mathrm{mm}$ round enameled copper wire (magnet wire). A schematic view of the coil is shown in Fig. 4. The average self inductance and series resistance of the coils is measured to be $\sim 25\,\mathrm{mH}$ and $\sim 12\,\Omega$, respectively. The inductance and resistance were measured at 1 kHz at room temperature using a Hameg MH8118 LCR meter (HAMEG Instruments GmbH, Mainhausen, Germany).

In order to verify whether the estimated coils will be sufficient to position a particle, we performed an analysis using a finite element method (FEM) simulation. The FEM model was created in Comsol Multiphysics® (COMSOL, Inc., Burlington, U.S.A). We modeled the coil as a thick hollow cylinder. The maximum current applied to the coils will be $0.8\,\mathrm{A}$. The current density in the cylinder is assumed to correspond to a current of $0.8\,\mathrm{A}$ in the coil. Since we are interested in the gradient inside our reservoir, we assume the center of this reservoir is $20\,\mathrm{mm}$ from the coil. Fig. 5 shows the gradient of $B^2$ around the center of the reservoir in the axial direction of the coil. In this case the coil is located on the negative side of the X-axis. Fig. 5 shows that the gradient in the center of the container is $\sim 1.8\,\mathrm{mT^2\,m^{-1}}$. Using (6) for a $100\,\mathrm{\mu m}$ particle we predict a maximum velocity of $149\,\mathrm{\mu m\,s^{-1}}$ due to the force of one magnet.




(1)

Fig. 5. Gradient of $B^2$ in the axial direction of the coil around the center of the reservoir for a current of $0.8\,\mathrm{A}$.

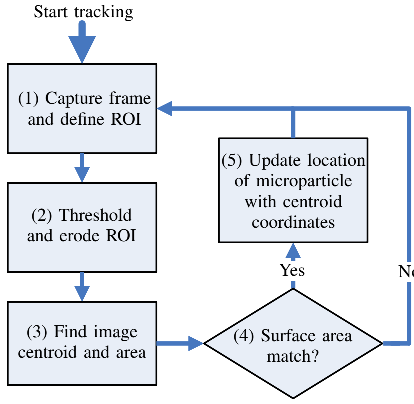



Fig. 6. Software flow with: (1) Frame capture and ROI definition (2) Adaptive thresholding and filtering (3) Centroid and area determination (4) Area checking (5) Updating the particle coordinates. The loop is run every 100 ms

### D. Image-based particle tracking

Tracking of the particles is done by use of image feedback from the black and white camera mounted on the microscope. The images captured by the camera are imported to a computer (Microsoft Windows XP SP3) running software capable of image processing. The software is written in C using the open-source computer vision libraries *OpenCV* [16]. The particle detection algorithm captures a frame every 100 ms. It then locates the particle using the steps depicted in Fig. 6 and explained below:

- The program captures a frame and defines a Region Of Interest (ROI). The center of the ROI can either be selected by the user or it is the last known location of the particle.
- The gray-scale image within the ROI is converted to a binary (monochrome) image using an adaptive thresholding algorithm. The algorithm takes the mean pixel value and subtracts the standard deviation of the pixel values in the ROI to determine a good threshold value. The adaptive nature of the thresholding makes the algorithm less dependent on lighting conditions.
- The resulting binary image is filtered using three iterations of an erosion filter to remove unwanted noise (the assumption is made that the particle is the only large object in the ROI).
- The image moments for the filtered binary image are determined. From these image moments the centroid coordinates and the area of the image are determined.

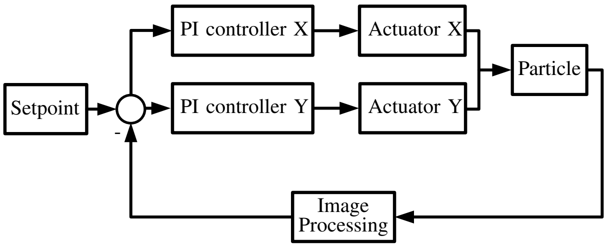



Fig. 7. Schematic diagram of the control loop. Two PI controllers are used to control the particle in both X and Y directions

- Finally, the calculated area is compared to the expected area of a particle. If the area is too large or too small to be a particle, it is assumed that no particle was detected and the particle coordinates are not updated. Thus, losing track of the particle. If the algorithm decides a particle was detected then the centroid coordinates become the new coordinates for the particle.

A limitation of the tracking algorithm is that it has a low immunity for errors that occur when multiple particles are inside the ROI. The algorithm assumes that there is only one particle in the image. If a second particle (or some other dark colored object) is in the ROI, the calculated centroid will be that of the combined particles. If the image with two particles does pass the area size test, the particle location is updated with incorrect coordinates. If the area size test fails, the tracker assumes it has lost the particle and the coordinates will no longer be updated. Since the preliminary experiments will be done in a controlled environment, most particle disturbances can be eliminated by careful preparation.

### *E. Magnetic control*

In order to control the movement of the particle to a given setpoint, a PI controller was implemented for each axis. The controllers run in a closed loop with the tracking software depicted in Fig. 7. Both controllers use equivalent settings for the proportional and and integral gains. The tracking algorithm provides the PI controllers with the difference between the actual position and the desired position of the particle. The update rate of the control loop is 100 ms, which is equal to the frame capturing rate of the particle tracking algorithm. Each axis is controlled independently and only one magnet is active per axis at the same time, depending on the direction in which the force should be applied to the particle. We have found no critical timing issues running the controllers at 10 Hz.

### *F. System integration*

The electromagnets are integrated into the setup that can be seen in Fig. 8. This setup allows the electromagnets to be positioned around the fluid reservoir. They are powered by Elmo 'Whistle' 1/60 servo controllers (Elmo Motion Control, Petach-Tikva, Israel) that are used as digitally controlled current sources. Each coil has its own dedicated controller to maximize flexibility of the system. The output current of these controllers is determined by the PI controllers. The computer uses an Arduino microcontroller board

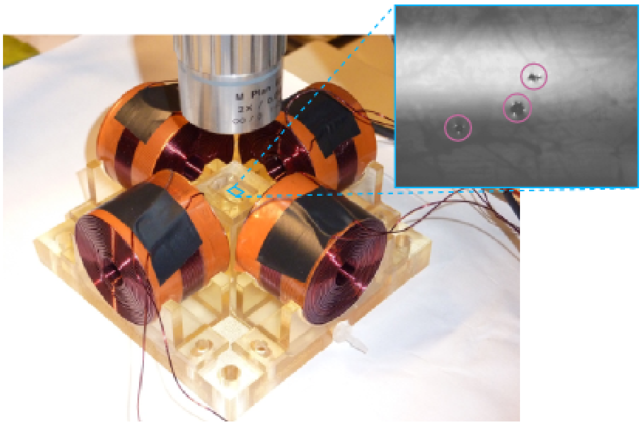


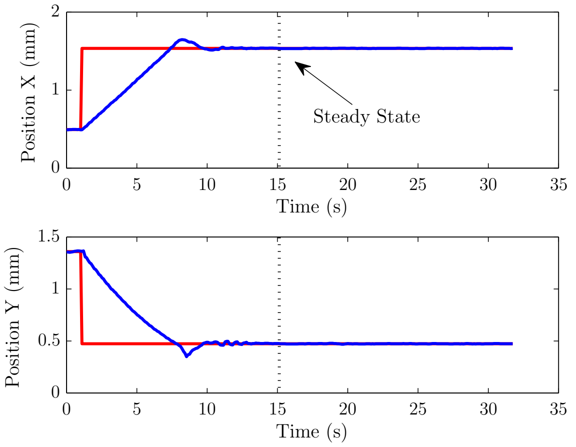
Fig. 8. Picture of the designed and assembled setup. Four coils fixed in a frame around a reservoir for liquids. The microscope objective is positioned above the reservoir. The image captured by the camera shows three particles.

(http://www.arduino.cc) to interface with each of the four current sources. The power supplied to the current sources comes from a 48 V, 5.2 A switched mode power supply unit.

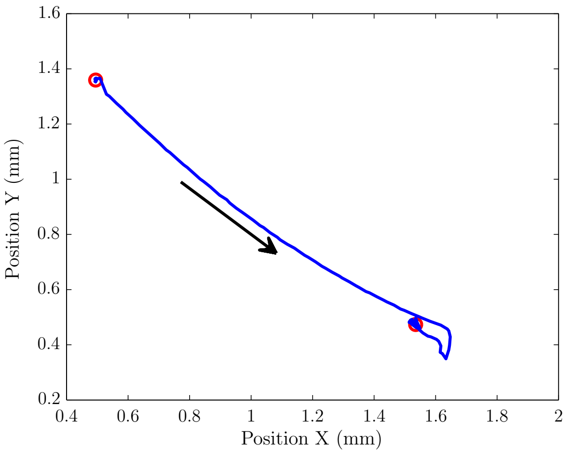
## III. RESULTS

In order to evaluate the positioning and tracking performance of the system, the following two experiments were conducted. First, an experiment to demonstrate the ability of the system to position the particle at a given setpoint. Second, an experiment to demonstrate the possibility to move the particle around a preset path of setpoints. For each experiment a single particle was selected. This particle had to be floating in the water-to-air boundary layer so it would not be affected by the surface friction at the bottom of the reservoir. The particle tracking algorithm performed adequately in locating the particles. Variations of contrast in the image did not cause the tracker to lose the particle. Several preliminary experiments were done with respect to positioning the particle. For each experiment, the gains were chosen such that the response of the system seemed suitable to perform the desired task. All resulting figures are based on the output provided by the tracking algorithm. Fig. 9(a) shows a step response of the system. The controller gains were set to $K_\mathrm{p} = 13$ and $K_\mathrm{i} = 0.5$. As can be seen in Fig. 9(b), there is a significant amount of overshoot in the particle trajectory. We argue this overshoot is mostly due to the integral action of the controller that needs time to recover and the movement of the fluid in which the particle resides. Changing the gains of the controller did reduce the overshoot on some occasions, however disturbances such as the flow of the fluid due to heat from the microscope light and air movement at the surface were compensated poorly.

From Fig. 9(a), we determine that the velocity of the particle was 182 µm s$^{-1}$ in the X-direction and 148 µm s$^{-1}$ in the Y-direction. This gives a total particle speed of 235 µm s$^{-1}$. We estimate the size of the particle used in this experiment to be $\sim$100 µm. Using (6) we find a theoretical maximum velocity of 149 µm s$^{-1}$ in the center of the reservoir due to a single coil. We see that the velocity in the Y-direction

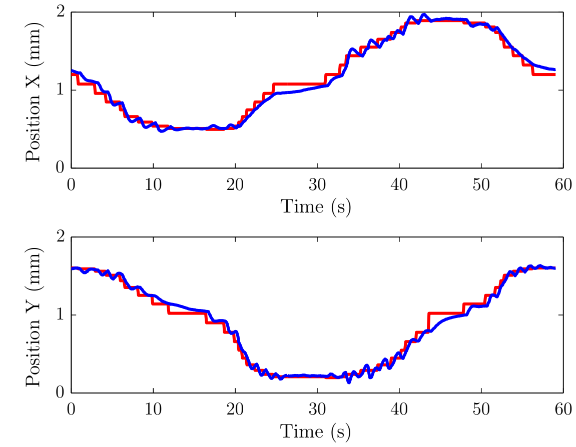



(a)

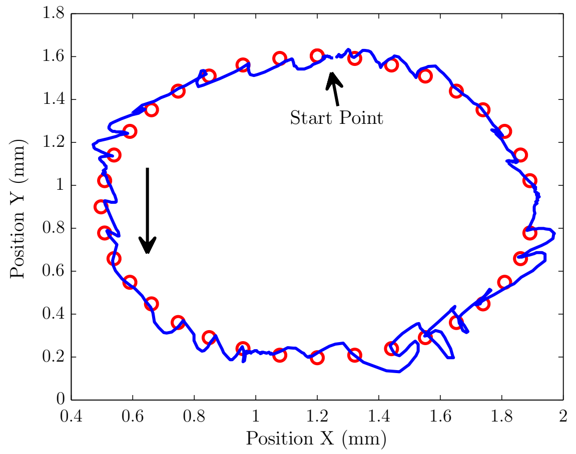



(b)

Fig. 9. Tracking performance to a single setpoint: (a) Each axis separately. (b) 2D trajectory of the combined axis. The red lines and circles represent the setpoints given to the controller. The arrow indicates the direction of movement.




(a)




(b)

Fig. 10. Tracking performance of a circular path: (a) Each axis separately. (b) 2D trajectory of the combined axis. The red lines and circles represent the setpoints given to the controller. The arrow indicates the direction of movement. The controller gains were $K_\mathrm{p}$ = 3 and $K_\mathrm{i}$ = 1.8.

corresponds to the theoretical value quite well. The velocity in the X-direction however is significantly larger than expected. We attribute this difference to discrepancies between the ideal model and the experimental implementation. The particles are not completely submerged, which reduces their drag. Additionally, the magnetic field is not entirely uniform, resulting in coupling between forces along X and Y. Finally, the aforementioned flow of the liquid also influences the observed motion of the particle.

In order to determine the positioning accuracy we define the error as the difference between the actual and the desired position of the particle. Fig. 9(a) shows that the particle reached a steady state around the desired position after 15 s. The standard deviation of the error and the maximum error in steady-state were calculated from a set of 174 data points (Table I).

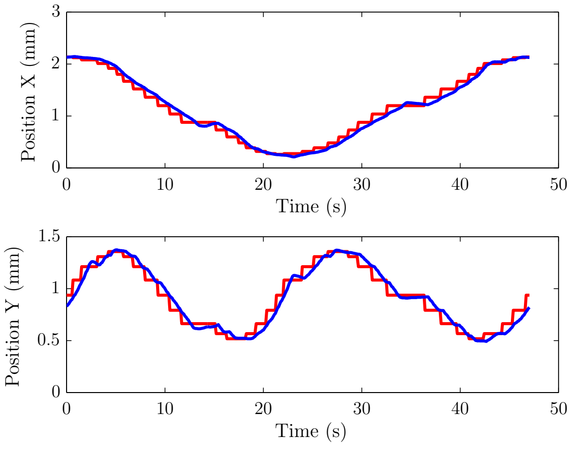
Figs. 10 and 11 show the results when a series of setpoints are given that describe a circular path and a figure-eight path, respectively. For these experiments a new setpoint is given when the particle is within 10 pixels ($\sim$23 µm) of the desired position. In the circular path, the average velocity of the particle was 83 µm s$^{-1}$. For the figure-eight path, the average velocity of the particle was 122 µm s$^{-1}$.

Most of the time it was not possible to use the same particle for multiple experiments. In our setup, the particles dried out and disintegrated after some time. New particles were taken for each experiment. We found, as expected from (6), that the size of the particle has a significant influence on the particles velocity. Therefore, for all experiments done, we have tried to manually select particles, that appeared have an equal size.

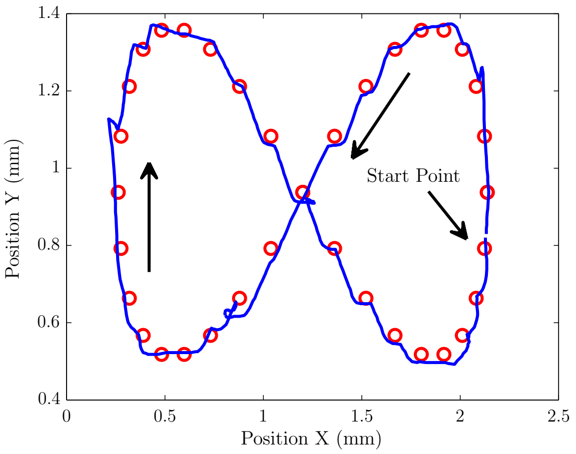
TABLE I

TABLE SHOWING THE STANDARD DEVIATION AND THE MAXIMUM VALUE OF THE ERROR IN STEAD-STATE ($>$ 15 s)

| Axis | Standard Deviation (µm) | Maximum Error (µm) |
|---|---|---|
| X | 1.8 | 4.7 |
| Y | 2.0 | 7.0 |




(a)




(b)

Fig. 11. Tracking performance of a figure-eight path: (a) Each axis separately. (b) 2D trajectory of the combined axis. The red lines and circles represent the setpoints given to the controller. The arrows indicate the direction of movement. The controller gains were $K_p$ = 15.5 and $K_i$ = 1.2.

## IV. CONCLUSIONS AND FUTURE WORK

We have shown that it is possible to control the position of paramagnetic microparticles suspended in water. This was accomplished using four electromagnets in a closed control loop with image-based feedback. We have demonstrated that with a fairly basic setup and a simple control algorithm that we were able to position a particle, with a average diameter of $100\,\mu m$, within $8.4\,\mu m$ of a desired position. The particle was observed to be traveling with a velocity of $235\,\mu m\,s^{-1}$. We have also demonstrated that we were able to steer the particle along a circular and figure-eight path.

The experimental setup and the preliminary results are the basis for future work to control self-propelled microrobots. To improve the positioning accuracy of the microrobot we aim to expand the controller to a model-based approach. The microrobots will be made out of platinum and cobalt, and will be placed in a reservoir that contains a solution of hydrogen peroxide ($H_2O_2$). The propulsion will come from the catalytic reaction that occurs between platinum and $H_2O_2$. Propulsion forces from this reaction can, for instance come in the form of oxygen bubbles that exert a force upon the microrobot [12]. When a combination of gold and platinum is used, propulsion can come from a process called self-electrophoresis [17]. When the microrobot will have its own propulsion, only control of the orientation of the robot is required. In this case, the magnetic fields will be utilized to exert a torque on the microrobot rather then a force. This can be achieved by integrating a magnetic strip inside the microrobot and exert a torque ($\tau$) on it according to $\tau = \mathbf{m} \times \mathbf{B}$. One of the advantages of using self-propelled robots over passive robots is the fact that the amount of magnetic energy needed to rotate an object is significantly less than to actually displace it. As a result, the size and cost of the required magnets will reduce considerably.

**Acknowledgements** - The authors would like to thank Gerben te Riet o/g Scholten (Control Engineering, University of Twente) and Martin Siekman and Thijs Bolhuis (Transducers Science and Technology, University of Twente) for their assistance and advice regarding the design of the setup.

## REFERENCES

[1] Menciassi *et al.*, "Microrobotics for future gastrointestinal endoscopy," *Minimally Invasive Therapy Allied Technologies*, vol. 16, no. 2, pp. 91–100, 2007.
[2] Abbott *et al.*, "Robotics in the small, part I: Microrobotics," *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, pp. 92–103, 2007.
[3] Abbott *et al.*, "How should microrobots swim?," *Int'l. J. Robotics Research*, vol. 28, no. 11-12, pp. 1434–1447, 2007.
[4] Kummer *et al.*, "Octomag: an electromagnetic system for 5-dof wireless manipulation," in *Proc. IEEE Int'l. Conf. Robotics and Automation*, (Anchorage, Alaska), pp. 1610–1616, May 2010.
[5] Choi *et al.*, "Two-dimensional actuation of a microrobot with a stationary two-pair coil system," *Smart Materials and Structures*, vol. 18, no. 5, 2009.
[6] Yesin *et al.*, "Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields," *Int'l. J. Robotics Research*, vol. 25, pp. 527–536, 2006.
[7] Mathieu *et al.*, "Method of propulsion of a ferromagnetic core in the cardiovascular system through magnetic gradients generated by an mri system," *IEEE Trans Biomed Eng*, vol. 53, no. 2, pp. 292–299, 2006.
[8] J.-B. Mathieu and S. Martel, "Steering of aggregating magnetic microparticles using propulsion gradients coils in an MRI scanner," *Magnetic Resonance Medicine*, vol. 63, no. 5, pp. 1336–1345, 2010.
[9] Fountain *et al.*, "Wireless control of magnetic helical microrobots using a rotating-permanent-magnet manipulator," in *Proc. IEEE Int'l. Conf. Robotics and Automation*, (Anchorage, Alaska), pp. 576–581, May 2010.
[10] Peyer *et al.*, "Non-ideal swimming of artificial bacterial flagella near a surface," in *Proc. IEEE Int'l. Conf. Robotics and Automation*, (Anchorage, Alaska), pp. 96–101, May 2010.
[11] S. Martel and M. Mohammadi, "Using a swarm of self-propelled natural microrobots in the form of flagellated bacteria to perform complex micro-assembly tasks," in *Proc. IEEE Int'l. Conf. Robotics and Automation*, (Anchorage, Alaska), pp. 500–505, May 2010.
[12] Ismagilovw *et al.*, "Autonomous movement and self-assembly," *Angewandte Chemie*, vol. 114, no. 4, pp. 674–676, 2002.
[13] Mirkovic *et al.*, "Nanolocomotion - catalytic nanomotors and nanorotors," *Small*, vol. 6, pp. 159–167, January 2010.
[14] Sundararajan *et al.*, "Catalytic motors for transport of colloidal cargo," *Nano Letters*, vol. 8, no. 5, pp. 1271–1276, 2008.
[15] Solovev *et al.*, "Catalytic microtubular jet engines self-propelled by accumulated gas bubbles.," *Small*, vol. 5, no. 14, pp. 1688–92, 2009.
[16] G. Bradski and A. Kaehler, *Learning OpenCV: computer vision with the OpenCV library*. O'Reilly Media, first ed., 2008.
[17] Paxton *et al.*, "Motility of catalytic nanoparticles through self-generated forces," *Chemistry A European Journal*, vol. 11, no. 22, pp. 6462–6470, 2005.

# Part II

## Improved Magnetic Control and Motion Planning of Paramagnetic Microparticles in Water

# Improved Magnetic Control and Motion Planning of Paramagnetic Microparticles in Water

Jasper D. Keuning, L. Abelmann and S.Misra

*Abstract*— **This paper describes the design of a controller to manipulate spherical paramagnetic microparticles that have an average diameter of $100\,\mu m$ using magnetic fields. A control algorithm was realized to move the particle along a straight line. Defining a path as a sequence of straight lines, allows us to have the particle track any arbitrary shaped path. The direction and magnitude of the magnetic force were determined using Finite Element Model (FEM) calculations. To demonstrate the capabilities of the controller, obstacle avoidance using the potential field method and path planning using the A* algorithm were implemented. Also a combination of both methods was proposed and demonstrated. It is shown that the new controller allows for an increase in path tracking speed of the particle with respect to the previously implemented controller [1]. A circular path was tracked with an average speed of $286\,\mu m\, s^{-1}$, compared to $83\,\mu m\, s^{-1}$ in [1]. A figure-eight path was tracked with an average speed of $197\,\mu m\, s^{-1}$ compared to $122\,\mu m\, s^{-1}$ in [1].**

## I. INTRODUCTION

The use of microrobots is a new promising field within minimally invasive surgery. Being able to maneuver small robots to hard-to-reach places in the human body for the purpose of drug delivery or treatment, can reduce the need for traumatic invasive surgery that would otherwise be needed. These microrobots can be manipulated using magnetic fields. If the microrobot has no propulsion of its own, a force can be exerted on it using the gradients of a magnetic field. When it is self-propelled, the magnetic field can be used to exert a torque on the robot, thus changing the orientation of the thrust vector. In earlier work we designed and implemented a system to manipulate superparamagnetic particles [1]. We had implemented a crude controller to demonstrate the capabilities of the setup in manipulating particles with an average diameter of $100\,\mu m$. The major drawback of this controller was that the path the particle would take was never well defined. Also the waypoint switching that had been implemented would sometimes require the particle to move backward along a path to reach a waypoint it had already passed.

In this work we propose an improved control method for controlling the trajectory of a microparticle with an new algorithm for waypoint switching. The method can also be directly applied to microrobots that are self-propelled. In addition to the controller we also describe the implementation of several motion planning algorithms to maneuver the particle to a given location while avoiding a number of obstacles.

In section II we describe the design of the controller. In section III we describe the implementation of path planning algorithm to maneuver the robot while avoiding a number of virtual obstacles. Section IV shows the results of several experiments demonstrating the controller and obstacle avoidance schemes.

## II. CONTROLLER

To improve the trajectory following behavior of the particle, a new control scheme was implemented. The control scheme is used to track a path consisting of multiple line segments.

### A. Basis of the trajectory following control

Fig.1 shows the overview of the control implementation. In this image P is the position of the particle in the global coordinate frame. $W_1$ and $W_2$ are the way points on the desired path. $W_2$ is the point towards which the robot should move. $\chi_f$ is the angle the path makes with respect to the global coordinate frame and $\chi_d$ is the angle between the velocity vector of the robot and the desired path. The tracking error ($e$) is defined as the shortest distance between P and the path. The factor s ($0 \leq s \leq 1$) is a measure for the progress along the path, measured by the projection of P on the path (P'). The control angle $\chi_d$ determines how fast the error is reduced. In order to determine this angle we first define a boundary ($b$) around the path that gives the maximum allowed deviation from the path. At and beyond this boundary all available force should be used for returning the robot towards the path ($\chi_d = \frac{\pi}{2}$) . Using this boundary condition, we can define a function for the angle [2]:

$$\chi_d = \frac{e}{|e|}\frac{\pi}{2}\left(\frac{|e|}{b}\right)^n \tag{1}$$

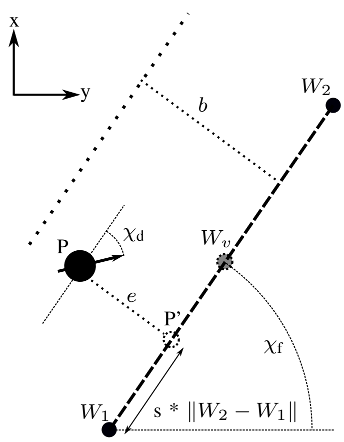



Fig. 1. Control problem parameters

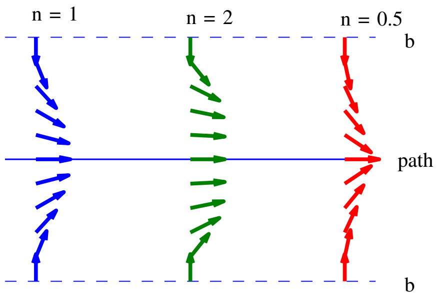



Fig. 2. $\chi_\mathrm{d}$ with respect to the value of n

Here n determines how aggressive the reduction of the error should be: n = 1 gives a linear relation between e and $\chi_d$. n $<$ 1 gives a large change of $\chi_d$ with respect to e and thus a strong counter action on small deviations from the path with respect to the linear case. Of course n $>$ 1 results in a weak counter action on a small deviation (Fig. 2).
Since environmental disturbances are present, a steady state error will most likely be present. In order to deal with this error, an integral term is added to the control angle

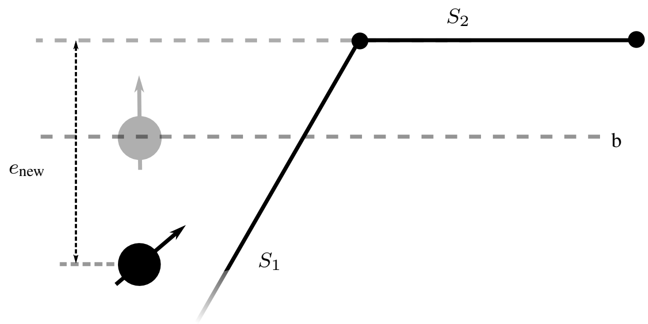
$$\chi_\mathrm{d} = \frac{e}{|e|}\frac{\pi}{2}\left(\frac{|e|}{b}\right)^n + K_\mathrm{i}\int_0^t e\mathrm{d}t \tag{2}$$

Here $K_\mathrm{i}$ is a gain factor that determines how much the integrated error influences the control angle.

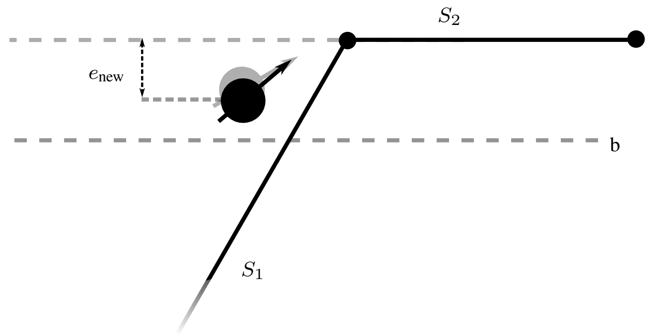
### B. Way Point switching

We assume every path is a set of waypoints connected by straight lines. The most simple method is to switch when $s \geq 1$. This is at the location where both line segments cross. It is also possible to switch when the distance between P' and $W_2$ is less than or equal to b. As a result, the particle will already be inside the boundary for the new path section. This does however not always give a smooth transition between waypoints. A better choice would be to determine what the control angle for next line segment would be if the switch was made, based on the current position P. The switch to this segment is made when when the current control angle with respect to the new segment is larger than or equal to the calculated angle for the new segment. As a result, the orientation of the force does not have to change between segments, thus allowing for a smooth transition. In order not to switch too soon, the particle should at least be within the boundary of the next segment. This is illustrated in Fig. 3. In Fig. 3(a), the calculated control angle for the new line segment ($S_2$) is perpendicular to this segment. Because the particle is still outside the boundary (b), the switch is not made. In Fig. 3(b), the calculated control angle for the new segment is almost equal to the current control angle. Since the particle is within the boundary of the segment, a switch is made to the next segment, allowing a smooth transition. In the global coordinate frame the control angle of the particle is defined as:

$$\chi_\mathrm{global} = \chi_\mathrm{d} + \chi_\mathrm{f} \tag{3}$$




(a)




(b)

Fig. 3. Smooth waypoint switching. The black circle is the current control angle. The gray circle represents the calculated control angle for the new line segment $S_2$. In (b) both control angles match up, and when a switch is made from tracking $S_1$ to tracking $S_2$ the transition will be smooth.

In order to make the decision to switch, the current $\chi_d$ should be expressed as an angle in the coordinate frame of the next segment of the path.

$$\chi_{\mathrm{d,old}\,\rightarrow\,\mathrm{new}} = \chi_\mathrm{d,old} + \chi_\mathrm{f,old} - \chi_\mathrm{f,new} \tag{4}$$

$$\chi_\mathrm{d,new} = \frac{e_\mathrm{new}}{|e_\mathrm{new}|}\frac{\pi}{2}\left(\frac{e_\mathrm{new}}{b}\right)^n \tag{5}$$

If the angle calculated in (4) is larger then the calculated angle in the new frame (5), the switch should be made.

### C. Magnetic Fields

In order to accurately manipulate the direction of movement of the robot, it is desirable to be able to accurately impose a vector of force on the robot. In order to do this, the magnetic fields and their interaction with the robot need to be accurately characterized. In earlier work [1] we calculated the force on a paramagnetic particle to be

$$\mathbf{F} = \alpha V \nabla(\mathbf{B}\cdot\mathbf{B}) \tag{6}$$

Where $\alpha$ is a material dependent constant with a value of $185\,\mathrm{kA}^2\,\mathrm{N}^{-1}$, V the volume of the particle and $\mathbf{B}$ the magnetic field. This can be rewritten as

$$\mathbf{F} = 2\alpha V\begin{bmatrix} B_x\frac{\partial B_x}{\partial x} + B_y\frac{\partial B_y}{\partial x} \\ B_x\frac{\partial B_x}{\partial y} + B_y\frac{\partial B_y}{\partial y}\end{bmatrix} \tag{7}$$

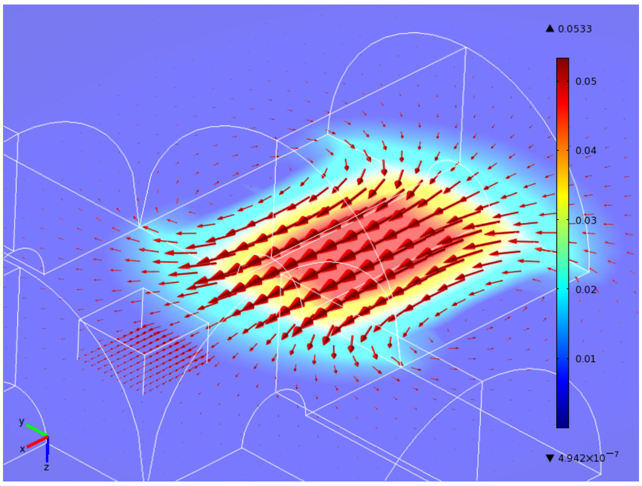



Fig. 4. FEM simulation results for a 1 Ampere current through one coil. Only the planar case for z = 0 is shown. The arrows represent $\mathbf{B}$ and the colored plane represents $\|\mathbf{B}\|$.

for a planar situation. $\mathbf{B}$ at a position $\mathbf{P}$ inside the reservoir is the summation of the field contributions of each coil at this point

$$\mathbf{B}(\mathbf{P}) = \sum_{i=0}^{n} \mathbf{B}_i(\mathbf{P}). \tag{8}$$

In our case we use four coils, so $n = 4$. The magnetic field of one coil is linearly proportional to the current (I) that is passed through the coil

$$\mathbf{B}_i(\mathbf{P}) = \bar{\mathbf{B}}(\mathbf{P})\mathrm{I}_i. \tag{9}$$

where $\bar{\mathbf{B}}$ is position dependent constant that relates the field strength to the current. The values of $\bar{\mathbf{B}}$ are determined using an FEM analysis of a single coil (Fig. 4). The FE model was created in Comsol Multiphysics® (COMSOL, Inc., Burlington, U.S.A). More information on the model used can be found in Appendix A. We fit a polynomial surface to the results from the FEM analysis (Fig. 4) using MATLAB Surface fitting toolbox (MATLAB 7.10, The MathWorks Inc., Natick, U.S.A.). We found that a 5th order polynomial surface fit yielded the smallest fitting error. Because the B field is a vector, we fit a polynomial to both the x component of the field ($B_x$) and the y component ($B_y$). The gradient of $\mathbf{B}$ can be easily calculated using the fitted function by either directly differentiating the polynomials for $B_x$ and $B_y$, or using a numerical approach.
In order to find the currents needed to give a given $\mathbf{F}$ at a certain $\mathbf{P}$, (7) needs to be solved. It is a non-linear problem with 4 variables and two equations. It can be solved using a non-linear solver using the Levenberg-Marquardt algorithm.

### D. Velocity feedback

The tracker used in this system only provides position feedback. In order to control the velocity vector, a good estimate of the velocity should be found. A simple method is to simply differentiate the position information. For low speeds, this can be quite noisy due to the quantization noise from the camera. To reduce the noise, it is possible to filter the position information before differentiating. This does however result in a delay between actual velocity and the calculated velocity.

Another way to get this information is to use a state estimator to predict the velocity based on the input, the measurements and a model of the system.

*1) State Space Model:* The particle is assumed to be a perfect sphere. It is actuated by a magnetic force $\mathbf{F}_\mathrm{m}$. A drag force $\mathbf{F}_\mathrm{d}$ that is dependent on the particles speed with respect to the liquid. If the liquid is stable the drag is only dependent on the particles speed with respect to the fixed world. The continuous time state space model of this system is given by

$$\dot{\mathbf{x}} = \mathbf{A}_\mathrm{c}\mathbf{x} + \mathbf{B}_\mathrm{c}\mathbf{F}_\mathrm{m}. \tag{10}$$

Where

$$\mathbf{A}_\mathrm{c} = \begin{bmatrix} -C_\mathrm{d}/m & 0 \\ 1 & 0 \end{bmatrix} \tag{11}$$

$$\mathbf{B}_\mathrm{c} = \begin{bmatrix} 1/m \\ 0 \end{bmatrix} \tag{12}$$

and $C_\mathrm{d}$ is a drag constant determined by Stokes drag at low Reynolds numbers, m is the mass of the particle and $\mathbf{x}$ is the matrix that contains the velocity vector ($\vec{v}(x,y)$) and the position ($p(x,y)$) of the particle $\begin{bmatrix} v & p \end{bmatrix}^T$.

The tracking algorithm only returns the estimated coordinates of the particle and gives no information about the velocity. The output of the system can be written as

$$y = \begin{bmatrix} 0 & 1 \end{bmatrix} \mathbf{x}. \tag{13}$$

We can then implement a discrete state observer

$$\hat{\mathbf{x}}(k+1) = \mathbf{G}\hat{\mathbf{x}}(k) + \mathbf{H}\mathbf{u}(k) + \mathbf{K}(\hat{\mathbf{y}}(k) - \mathbf{y}(k)) \tag{14}$$

$$\hat{y}(k) = \mathbf{C}\hat{\mathbf{x}}(k) \tag{15}$$

Where $\mathbf{G}$ and $\mathbf{H}$ represent the discretized system. When the error (e) is defined as

$$\mathbf{e}(k) = \hat{\mathbf{x}}(k) - \mathbf{x}(k) \tag{16}$$

The observer is asymptotically stable when e converges to zero for $k \to \infty$. For a Luenberger observer, this is the case when the eigenvalues of $(\mathbf{G} - \mathbf{KC})$ are inside the unit circle. The system from (10) needs to be discretized before it can be used in the discrete time observer. Discretization is done by computing

$$\mathbf{G} = e^{\mathbf{A}_\mathrm{c}T} \tag{17}$$

$$\mathbf{H} = \left( \int_{\tau=0}^{T} e^{\mathbf{A}_\mathrm{c}\tau} d\tau \right) \mathbf{B}_\mathrm{c} \tag{18}$$

Where $T$ is the sample time. Since matrix exponentials are computationally intensive, an approximation can be made, for instance using a bilinear transform

$$e^{\mathbf{A}_\mathrm{c}T} \approx \left( \mathbf{I} + \frac{1}{2}\mathbf{A}_\mathrm{c}T \right) \left( \mathbf{I} - \frac{1}{2}\mathbf{A}_\mathrm{c}T \right)^{-1} \tag{19}$$

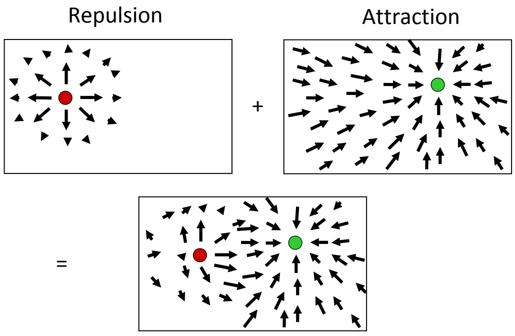



Fig. 5. The idea of potential fields with repulsive and attractive forces. The red circle represents an obstacle and the green circle is the target.

Using $C_d = 0.94 \times 10^{-6}$ and $m = 1.27 \times 10^{-6}$ in (10), the discretized system becomes

$$\mathbf{G} = \begin{bmatrix} 0.9218 & 0 \\ 0.1056 & 1 \end{bmatrix} \tag{20}$$

$$\mathbf{H} = 1 \times 10^4 \times \begin{bmatrix} 8.3182 \\ 0.4637 \end{bmatrix} \tag{21}$$

Lastly the observer gain $\mathbf{L}$ be computed, depending on the desired location of the poles of the observer. It was found that both the moving average filter as the state estimate were too slow or noisy to be useful in the controller. So the speed calculations were not used.

## III. PATH FINDING

In order to demonstrate the capabilities of the controller, two path generation methods are implemented to steer the particle towards a target avoiding a number of virtual obstacles. The obstacles are specific areas that the particle should avoid while moving towards its target. For this work the obstacles are chosen to be circular with varying radii and positions. Though any obstacle shape can be used, as long as their edges and the normal vectors on these edges are known.

### A. Potential Field Control

The potential field method is based on the idea that the obstacles exert a repulsive force onto the particle, thus pushing the particle away. The target has an attractive force pulling the particle in its direction. This method only needs information about the obstacles that are near to the particle (Fig. 5). One important feature is that the attraction of the target is homogeneous and the magnitude does not depend on the location with respect to the target. The obstacles only have a local field of repulsion, that reduced linearly to zero at the boundary. It therefore does not require a lot of computing power and can quite easily deal with a changing environment. This can be useful in applications where only a limited amount of information about the surroundings is available. This can be a camera with only a limited field of view, or feedback from an ultrasound device that only observes a small portion of the space in which to maneuver the particle.

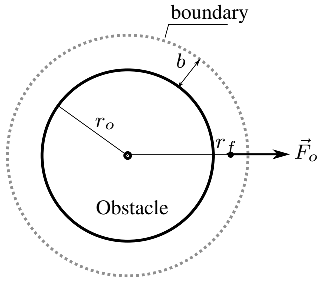



Fig. 6. Schematic representation of a circular obstacle with a radius $r_o$

In this work the obstacles are circular as shown in Fig. 6. A boundary has been defined around the obstacle in which the obstacle has an influence on the particle. Inside the obstacle, all vectors point outwards from the center at maximum length. Between the edge of the obstacle and the boundary the force vector ($\vec{F_o}$) points outward with a magnitude that is determined by (23).

$$d = r_\mathrm{f} - r_\mathrm{o} \tag{22}$$

$$\vec{F}_\mathrm{o} = (b - d)\frac{F_\mathrm{r}}{b}\vec{n} \tag{23}$$

where $F_\mathrm{r}$ is the repulsive force on the edge of the obstacle and $\vec{n}$ is the vector perpendicular to the edge of the circle. The total repulsive force at a location in the grid is a summation of the force contributions from all obstacles.

$$\vec{F}_\text{o,tot} = \sum_{i=1}^{k} F_\mathrm{o}^i \tag{24}$$

Where $k$ is the total number of obstacles in view. The main drawback of this method is that without modifications of the basic algorithm, the particle can get stuck in local minima [3].

### B. Path finding based on A*

The A* path finding method is a graph based search that tries to find a minimal cost route towards the target. [4] The method guaranties to find a path if one exists.

We have implemented the algorithm as follows: The screen with all virtual obstacles is mapped onto a squared grid . For this squared grid a weight map is calculated based on the locations of the obstacles (Algorithm 1). If a grid point is part of an obstacle, its weight is given the highest possible value. If it is outside the obstacle, but within a certain boundary, the weight is inversely proportional to the distance from the obstacle edge. Outside the boundary the weight is 0. The boundary is user defined. A large boundary will keep the path far away from the obstacles, resulting in a larger overall path length. An advantage of this, is that when obstacles move after the path is planned, the larger margin around the obstacles lowers the chance of collision. A small boundary will result in a shorter path, but the path will be closer to the obstacles. This increases the chance of hitting an obstacle when an unexpected disturbance affects the trajectory of the particle. If multiple obstacles influence the weight at a certain

TABLE I

VARIABLES AND FUNCTIONS FOR PATH FINDING

| Variable or Function | Description |
|---|---|
| $O_\mathrm{r}$ | Obstacle radius |
| $O_\mathrm{b}$ | Obstacle boundary |
| $O_\mathrm{p}$ | Obstacle position at coordinate (x,y) |
| $G_\mathrm{p}$ | Grid position at coordinate (x,y) |
| $G_\mathrm{w}$ | Grid weight at coordinate (x,y) |
| $G_\mathrm{w,max}$ | Maximum grid weight |
| d$(P_1, P_2)$ | Absolute distance between points $P_1$ and $P_2$ |

**Algorithm 1** Weight mapping based on circular obstacles

For all (x,y) and all Obstacles do:
d = $d(G_\mathrm{p}, O_\mathrm{p})$
**if** d $\leq O_\mathrm{r}$ **then**
  $G_w$ := $G_\mathrm{w,max}$
**else**
  **if** d $\leq (O_\mathrm{r} + O_\mathrm{b})$ **then**
    wt := $((O_\mathrm{r} + O_\mathrm{b})$ - d) * $G_\mathrm{w,max}$ / $O_\mathrm{b}$
    $G_\mathrm{w}$ := $G_\mathrm{w} \geq$ wt ? $G_\mathrm{w}$ : wt
  **end if**
**end if**

grid point, the highest value for the weight is assumed. The cost to move from a node to a horizontal or vertical neighbor is 10 plus the weight of the neighbor ($G_\mathrm{w,neighbor}$). The cost for a diagonal step is 14 + $1.4 \times G_\mathrm{w,neighbor}$. Several heuristics have been tried: Manhattan-distance [5], Chebyshev-distance and Euclidean-distance. All of these heuristics give similar performance in computation time ( $< 3$ ms on a $64{\times}48$ grid) for our situation. Usually the Manhattan-distance returns the fewest nodes after smoothening, therefore this heuristic is used. It is calculated by calculating the costs ($H$) of the path if only horizontal and vertical steps are taken from the node towards the target:

$$\mathrm{H} = 10 \times (|\mathrm{node}_x - \mathrm{target}_x| + |\mathrm{node}_y - \mathrm{target}_y|) \tag{25}$$

The algorithm will return a list of nodes that, when connected, describe the path towards the target. Since we run A* on a squared grid, the path described by the nodes only has 8 possible directions (top, bottom, left, right and the four diagonal directions). Also, straight lines can contain a number of nodes. In order to reduce the number of nodes, a line-of-sight (LOS) algorithm [6] is used. Starting at the location of the particle, it determines the cost for a straight line drawn from the starting point to the nodes in list returned by the A* algorithm. When the cost of the line is less then the cost to reach a certain node, the node is removed from the list. This is done for all resulting nodes, thereby reducing the nodes in the list (Algorithm 2).

Where NodeList[] is the list of nodes returned by the A* algorithm and SNodelist[] is the list of nodes containing the smoothened path. This algorithm does not check all possible combinations of smoothing and can therefore not guarantee the best reduction in points. It does however reduce

**Algorithm 2** Path Smoothening

N := NodeList[0]
SNodeList.add := NodeList[0]
**for** i $<$ (number of nodes in NodeList) **do**
  **if** cost(LOS(N,NodeList[i])) $>$ cost(NodeList[i]) **then**
    {Add previous node to smooth node list}
    SNodeList.add := NodeList[i-1]
    {Continue looking from last node}
    N := NodeList[i-1]
  **end if**
**end for**
return{SNodeList now contains the smoothened path}

the cost of the path significantly in most cases, at little added computational cost. The various steps involved in path planning are shown in Fig 7.

### C. Using a combination of A* and the Potential method

When looking at a realistic situation where obstacles are rarely stationary, the path calculated by A* may no longer exist after some time. As long as the path can be recalculated in one program cycle, it is possible to update the path based on the new situation on-line. If the calculations become too intensive and recalculating is not possible on-line, it is possible to use the path calculated as a guide and to avoid the moving obstacles using the potential method from section III-A. The current implementation is fairly simplistic but effective. Basically, the potential method is applied in the motion planning of the robot. However, instead of choosing the endpoint as the target, the waypoints provided by the path planning algorithm serve as sub-targets. One added condition is that a sub-target should not be inside an obstacle. If this happens, the first sub-target that is outside the obstacles should be selected.

## IV. EXPERIMENTAL RESULTS

For this experiment similar microparticles have been used as in our earlier work ((PLA-Particles-M-plain from Micromod Partikeltechnologie GmbH, Rostock-Warnemuende, Germany) as shown in Fig 8. Although the container holds particles of different sized (Fig. 8), all experiments below have been performed with manually selected particles that have a diameter of approximately 100 µm. The control loop for these experiments has a loop time of 100 ms. Unless otherwise specified, the boundary for the controller is set at 40 pixels (94 µm) and the boundary around the obstacles is set at 50 pixels (117 µm). Also the exponent $n$ from eq 1 was found to give good results when given a value of 2.

### A. Controller

Fig. 9 shows the particle tracking a squared path. Although overshoot occurs at the corners, it can be seen that the particle trajectory stabilizes. And it tracks the line with a decreasing error. The overshoot is represented by the positive peaks in the error graph (Fig. 9(b)). The large overshoot is caused by the fact that the particle will start tracking the

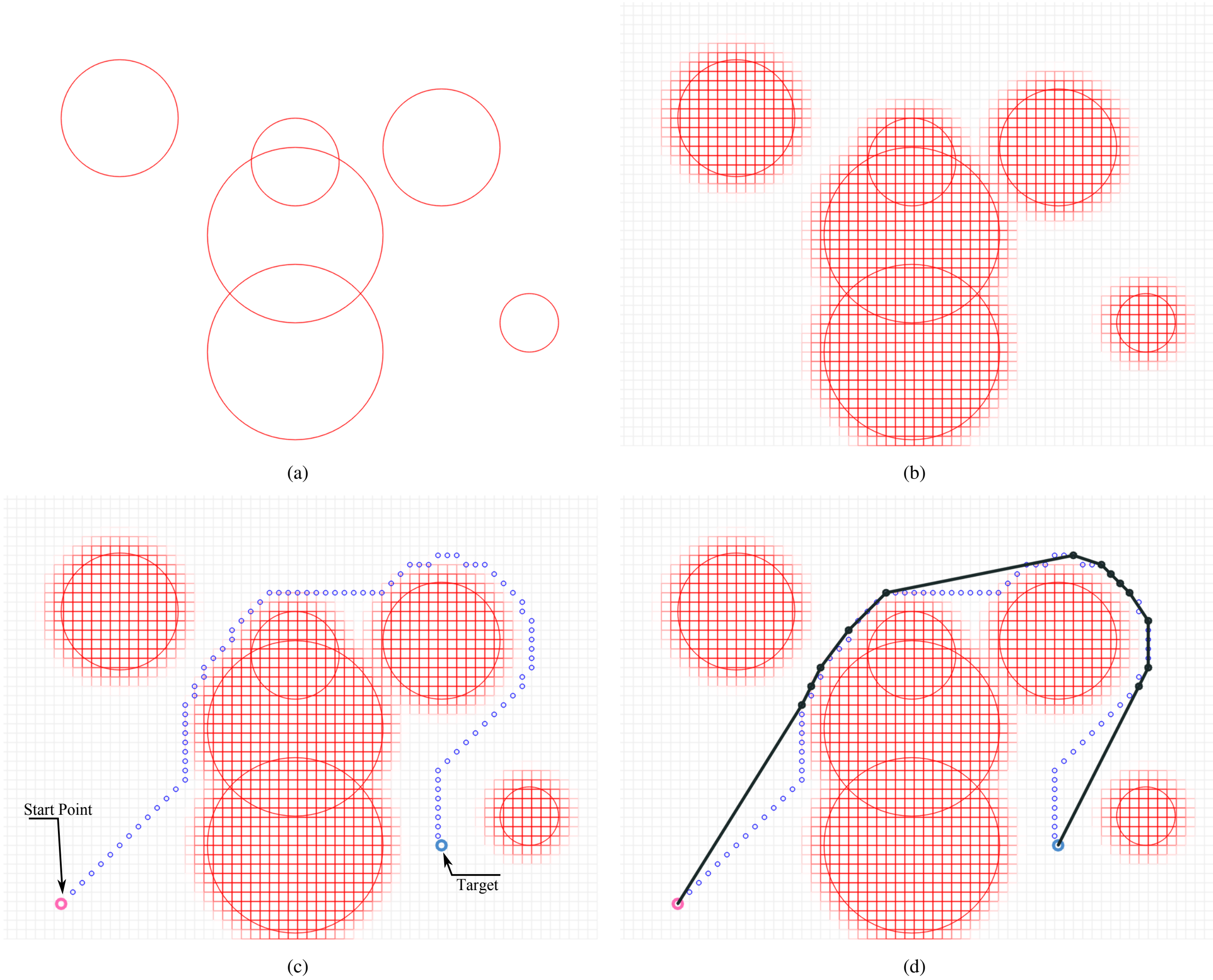



Fig. 7. (a) The obstacles around which a path needs to be planned. (b) S grid is defined and cost map is created. The cost map shows the cost to reach a certain area. The shade of red determines the cost to reach that point on the grid. The darkest red is unreachable space and the faded red is reachable but at an added cost. (c) The A* algorithm determines a low cost path through the grid. Where the thick green circle is the starting position and the thick blue circle is the target position. (d) The smoothening algorithm reduces the previously calculated set of grid points to a smooth path for the particle to follow.

new line segment when it touches this line. The controller needs time to react to the new situation. During this time the particle continues to move away from the line, causing the observed overshoot. When smooth waypoint switching is used, the overshoot at the corners is decreased from 200 µm to 50 µm as is shown in Fig. 10. The negative spikes shown in Fig. 10(b) are due to the way point switching. Because the controller changes the line-segment to be tracked before the particle reaches it, it will still have some distance from this segment, thus a in the error spike right after switching. In order to compare the current controller with the controller from the previous work, we tracked a circle (Fig. 11) and a figure-eight path (Fig. 12). The figures show a clear improvement in the path followed by the particle. It is smoother and it tracks the prescribed path more accurately. The fact that this path is smooth is most likely also the reason why the average speed of the particle increases using the new controller. The particle does not have to slow down or even move backwards to reach a certain setpoint. Saving a lot of time, which is clearly visible in Fig. 11. Below are the results of the potential field method and he path planning we proposed. For these experiments a fixed virtual obstacle configuration is used for easy comparison.

*1) Potential Method:* Fig. 13 shows the result of an experiment using the potential field method. As can be seen in the image, when the particle is near to an obstacle, it gets rejected and pushed away. When no local minima exist, the particle will gradually move towards the target. If the target is chosen at a position such as in Fig. 14, the particle will get stuck in the potential field and will be unable to reach its target.

*2) Path planning:* Fig. 15 shows the result of an experiment done avoiding static obstacles. The path is successfully planned around the obstacles. The particle follows the path, reaching its target without colliding with any obstacles.

*3) Combined planning and potential:* Fig. 16 shows the result when the potential method is combined with the path planning. As can be seen, after planning the path, the large

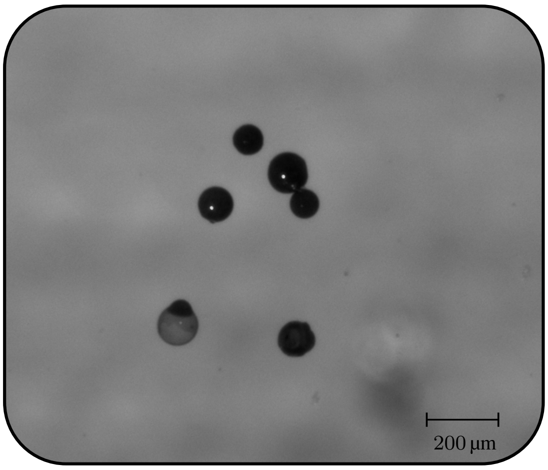
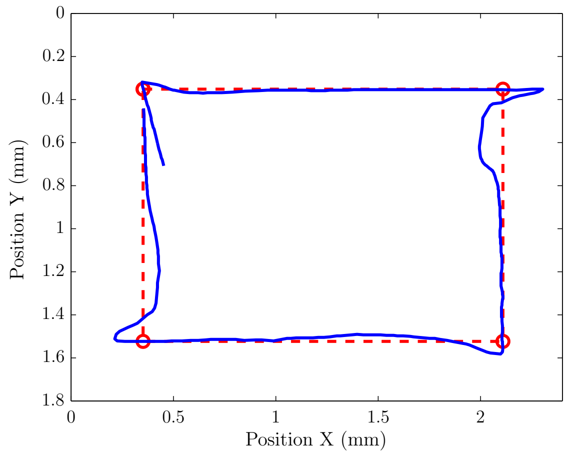
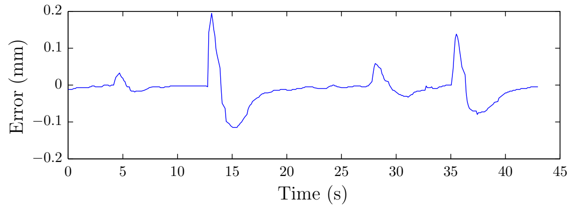



Fig. 8. Superparamagnetic microparticles floating on the surface of water as seen by the camera in the setup.




(a)




(b)

Fig. 9. Particle trajectory tracking a squared path. The red dashed line represents the path defined by the setpoints. The blue line represents the actual path taken by the particle. The particle moves in a clockwise direction starting at the top left corner.

obstacle that used to be in the center of the field of view has moved, blocking the planned path. The particle safely avoids the obstacles. For this experiment, the repulsive force of the obstacle has been set to 40, the attractive force of the (sub) target has been set to 40 and a large boundary of 80 pixels (188 µm) has been chosen. The reason for the large boundary is to clearly show the effect of the method.

## V. CONCLUSIONS

A controller has been proposed and implemented for the magnetic manipulation of paramagnetic particles in a 2D space. The focus of the controller is steer the particle towards a line and to track this line within a set boundary. Any path followed is a sequence of straight line segments. The new controller shows significant improvement in tracking performance over the controller used in the previous work. A circular path was tracked with an average speed of $286\,\mu m\,s^{-1}$, which is $\sim$ 3.5 times faster then the $83\,\mu m\,s^{-1}$ from the previous work. Also the path followed by the particle is smoother. Similar results were observed when comparing the tracking of a figure-eight path. The figure-eight path was tracked with an average speed of $197\,\mu m\,s^{-1}$, which is almost $\sim$ 1.6 times faster then the $122\,\mu m\,s^{-1}$ from the previous work. To further demonstrate the capabilities of this controller a path planning algorithm has been implemented that plans a path around a number of obstacles towards a given target. We demonstrated the tracking capability of the controller using a predefined path, or a path determined by the path planner. Also a potential field method has been successfully used to maneuver the particle towards a target while avoiding a number of obstacles. Lastly a combination of path planning and potential field control has been demonstrated to avoid obstacles that have moved after the path had been planned.

We could accurately predict the magnetic forces by a fit to FEM results. However the non-linear solver did not provide a constant and stable solution for the force on the particles. The reason for this is currently not clear. Also the velocity estimation was not usable in this work because the response was too slow, limiting the bandwidth of the controller too much.

## VI. RECOMMENDATIONS & FUTURE WORK

The next step to take, in the manipulation of the particles, is to move to manipulation in a 3D environment. This will require a number of adaptations in the current setup. Most significantly, that more magnets need to be added in order to be able to control the particles in the all directions and the camera setup needs to be modified so it can track particles in all three dimensions.

The bandwidth of the controller is fairly small with a sampling time of 10 Hz. For accurate speed measurements or estimates and a fast controller response an increase in bandwidth is desired. Also the Luenberger observer is not an optimal observer. Implementing a better and optimal observer such as a Kalman observer can give better results.

When moving towards clinical applications where microrobots will be used for drug delivery, the size of these robots will have to go down by several orders of magnitude. The magnetic force on a particle is proportional to its volume (eq. 6). The drag force on a particle is proportional to its radius [1]. As a result, when the particles get smaller the magnetic field strength needs to go up significantly to achieve similar behavior at a smaller scale.

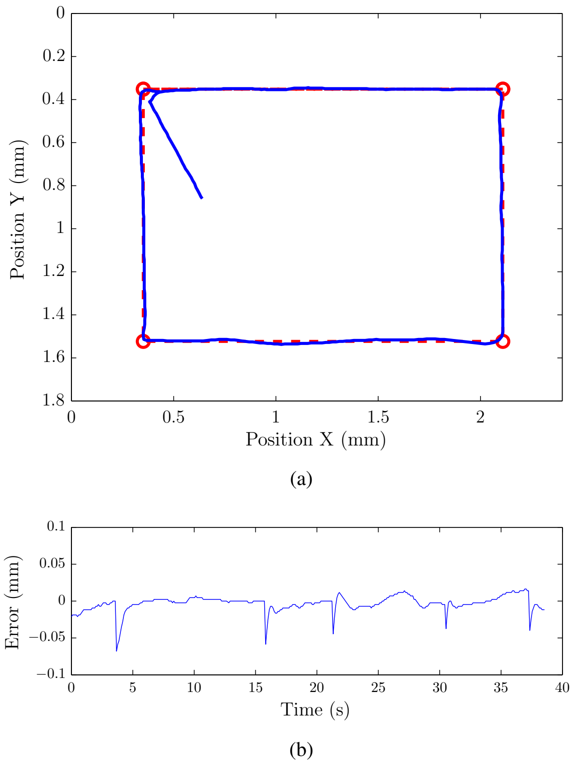



Fig. 10. Particle trajectory tracking a squared path using the improved waypoint switching algorithm. The red dashed line represents the path defined by the setpoints. The blue line represents the actual path taken by the particle. The particle moves in a clockwise direction starting at the top left corner.

For the magnetic torque, this is different. The amount of torque exerted on the particle is proportional to its volume, the drag torque the particle experiences is also proportional to its volume [7]. As a result, the speed of rotation will remain the same irregardless of the particle size, when the same magnet configuration is used. The next step to take in the design of the microrobots, is to make them self-propelled. Then the magnetic forces to position the robot are no longer needed. Only a torque is need to change the orientation of the propulsion. This robot can then be reduced in size without the need to greatly increase the size of the magnets used for controlling it.

## Appendix

### A. FEM Data

We model the coil as a thick hollow cylinder. The current density in the cylinder is assumed to correspond to a current of 1 A in the coil. All mesh elements are tetrahedral. The mesh for the coil had a maximum element size of 1 cm. A squared block was defined in the center of the model (Fig. 4). This block represents the area where the reservoir will be and was given a finer mesh with a maximum size of 1 mm. All other space was filled with a mesh with a maximum size of 2.4 cm a minimum size of 3 mm. We did a stationary analysis using the Flexible Generalized Minimum Resisudal (FGMRES) solver.

## References


[1] J. D. Keuning, J. de Vries, L. Abelmann, and S. Misra, “Image-based magnetic control of paramagnetic microparticles in water,” in *Proc. IEEE/RSJ Int Intelligent Robots and Systems (IROS) Conf*, pp. 421–426, 2011.

[2] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, “Vector field path following for small unmanned air vehicles,” in *Proc. American Control Conf*, 2006.

[3] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *Proc. Conf. IEEE Int Robotics and Automation*, pp. 1398–1404, 1991.

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE T Syst Sci Cyb*, vol. 4, no. 2, pp. 100–107, 1968.

[5] E. F. Krause, *Taxicab Geometry*. Dover, 1987.

[6] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, pp. 25–30, 1965.

[7] U. Lei, C. Y. Yang, and K. C. Wu, “Viscous torque on a sphere under arbitrary rotation,” vol. 89, no. 18, pp. 181908–181908–3, 2006.

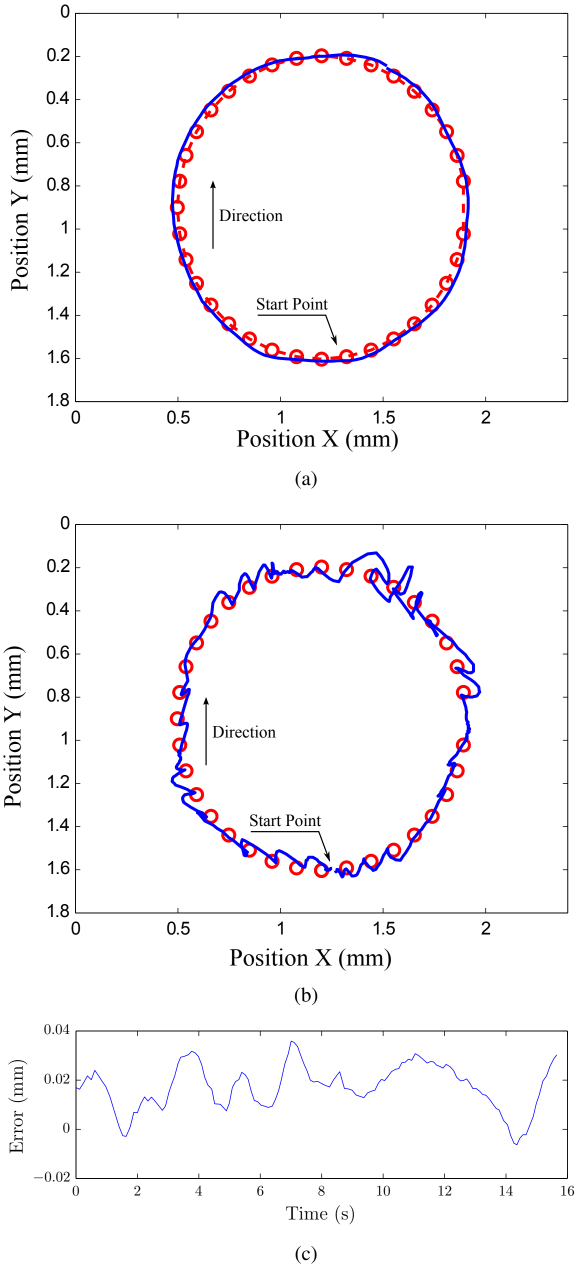



Fig. 11. Particle trajectory tracking a circular path. The red dashed line represents the path defined by the waypoints. The blue line represents the actual path taken by the particle. (a) Is the result for tracking a circular path using the current controller. The average speed achieved was $286\,\mu\mathrm{m\,s^{-1}}$. For comparison, (b) shows the result from the previous work. Here the average speed achieved was $83\,\mu\mathrm{m\,s^{-1}}$. (c) shows the deviation from the circular path for (a) assuming a perfect circle. A positive error means the particle is outside the circle, a negative error means it is inside.

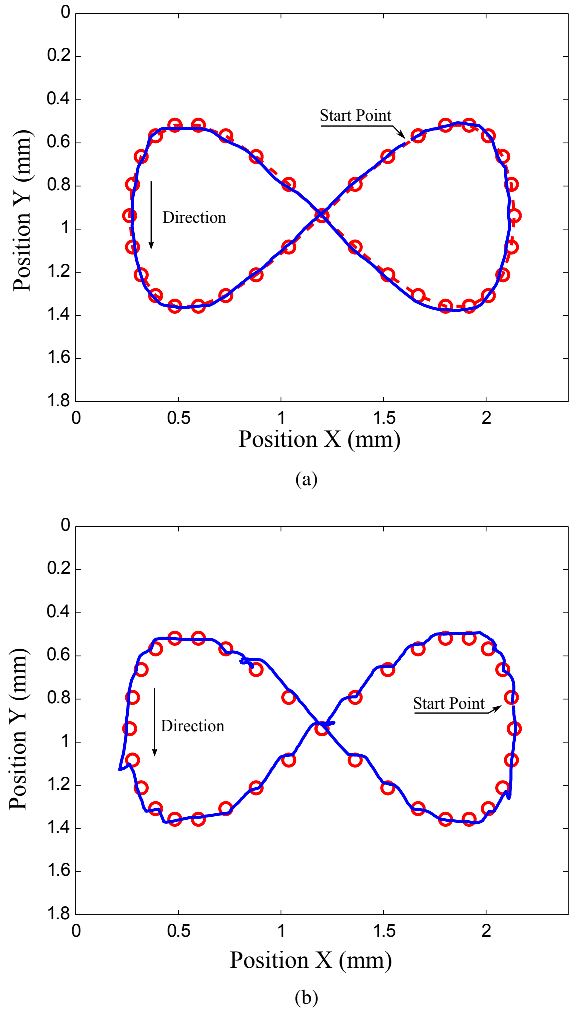



Fig. 12. Particle trajectory tracking a figure-eight path. The red dashed line represents the path defined by the waypoints. The blue line represents the actual path taken by the particle. (a) Shows the result using the current controller the average speed was measured to be $197\,\mu\mathrm{m\,s^{-1}}$. (b) Shows the result from the previous work where an average speed of $122\,\mu\mathrm{m\,s^{-1}}$ was achieved.

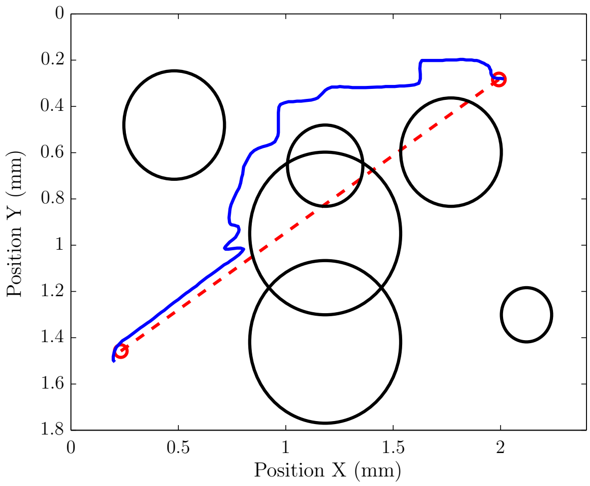



Fig. 13. Experiment using potential field method. The red dashed line represents the shortest path. The blue line represents the actual path taken by the particle. The particle starts at the bottom left and moves towards the top right in the figure.

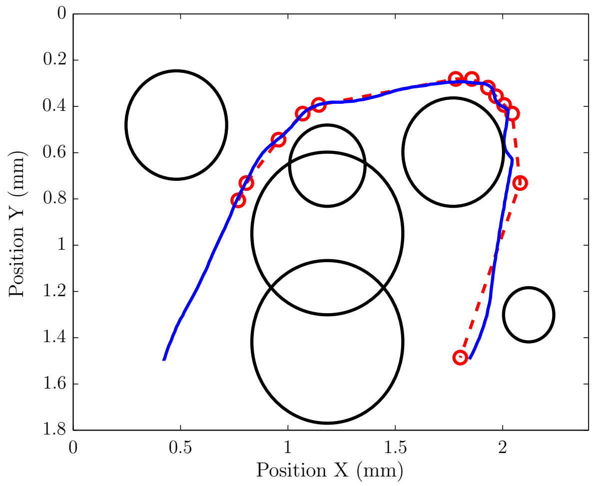



Fig. 15. Result of experiment using static obstacles and path planning. The red dashed line represents the preferred path. The blue line represents the actual path taken by the particle. The particle starts at the bottom left and moves towards the bottom right in the figure.

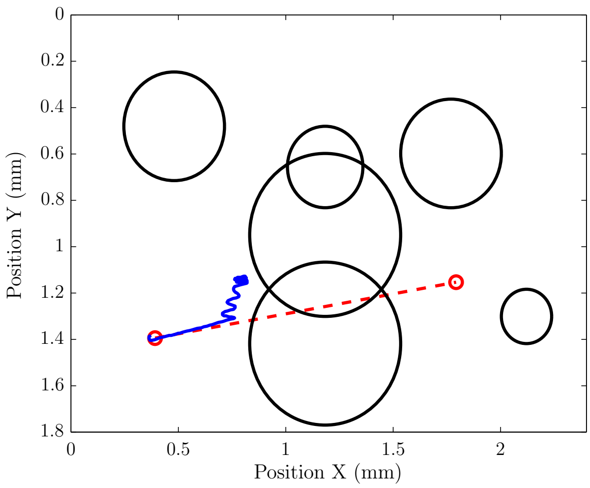



Fig. 14. Experiment demonstrating local minima in the potential field method. The red dashed line represents the preferred path. The blue line represents the actual path taken by the particle. The particle tries to move from the bottom left to the right in the figure.

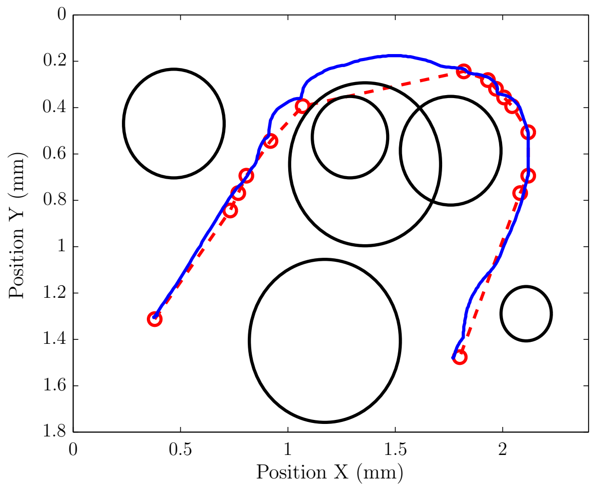



Fig. 16. The top middle obstacle has moved after planning the path (red dashed line). Due to the repulsive forces of the obstacles, the particle is forced out of its path around the obstacle (blue line). The particle starts at the bottom left and moves towards the bottom right in the figure.

# Conclusions

We have shown that it is possible to control the position of paramagnetic microparticles suspended in water in 2D space. This was accomplished using four electromagnets in a closed control loop with image-based feedback. We have demonstrated that with a fairly basic setup and a simple control algorithm that we were able to position a particle, with a average diameter of $100\,\mu\text{m}$, within $8.4\,\mu\text{m}$ of a desired position.

An improved controller has been proposed and implemented for the magnetic manipulation of the particles. The focus of this controller is steer the particle towards a line and to track this line within a set boundary. Any path followed is a sequence of straight line segments. A circular path was tracked with an average speed of $286\,\mu\text{m}\,\text{s}^{-1}$. A figure-eight path was tracked with an average speed of $197\,\mu\text{m}\,\text{s}^{-1}$.

To further demonstrate the capabilities of this controller a path planning algorithm has been implemented that plans a path around a number of obstacles towards a given target. We demonstrated the tracking capability of the controller using a predefined path, or a path determined by the path planner. Also a potential field method has been successfully used to maneuver the particle towards a target while avoiding a number of obstacles. Lastly a combination of path planning and potential field control has been demonstrated to avoid obstacles that have moved after the path had been planned.

We could accurately predict the magnetic forces by a fit to FEM results. However the non-linear solver did not provide a constant and stable solution for the force on the particles. The reason for this is currently not clear. Also the velocity estimation was not usable in this work because the response was too slow, limiting the bandwidth of the controller too much.

# Recommendations and Future Work

The bandwidth of the controller is fairly small with a sampling time of 10 Hz. For accurate speed measurements and fast controller response an increase in bandwidth is desired. Also the Luenberger observer is not an optimal observer. Implementing a better and optimal observer such as a Kalman observer can give better results.

The next step to take, in the manipulation of the particles, is to move to manipulation in a 3D environment. This will require a number of adaptations in the current setup. Most significantly that more magnets need to be added in order to be able to control the particles in the all directions and the camera setup needs to be modified so it can track particles in all three dimensions.

The coils currently have no magnetic core. Adding a core will increase the field strength considerably. This can be important when future work requires a larger workspace or smaller particles that need force manipulation.

When moving towards clinical applications where microrobots may be used for drug delivery, the size of these robots will have to go down by several orders of magnitude. The magnetic force on a particle is proportional to its volume. The drag force on a particle is proportional to its radius. As a result, when the particles get smaller the magnetic field strength needs to go up significantly to achieve similar behavior at a smaller scale. For the magnetic torque, this is different. The amount of torque exerted on the particle is proportional to its volume, the drag torque the particle experiences is also proportional to its volume. As a result, the speed of rotation will remain the same irregardless of the particle size, when the same magnet configuration is used. The next step to take in the design of the microrobots, is to make them self-propelled. Then the magnetic force to position the robot are no longer needed. Only a torque is need to change the orientation of the propulsion. This robot can then be reduced in size without the need to greatly increase the size of the magnets used to controlling it.

# Appendix Software

The code is need some cleaning up and giving full class diagrams here will not be useful. Therefore the most important classes and a description of these classes is given.

## Class: Particle

The Particle class contains all information regarding a particle being tracked. Calling the memberfunction update() will call the particles tracking algorithm and its location stats are updates (such as location and velocity based on a moving average filter)

### Important Memberfunctions

Particle(location): Creates a particle at a given location
Update(): Runs the tracking algorithm to update the location of the particle
GetLocation(): returns the current location
SetLocation(location): sets the location where to look for the particle

## Class: Obstacle

The Obstacle class creates obstacle objects. All obstacles are circular and have a location and a radius. The update() memberfunction can be called to move the obstacle. Movement will be small and will have a sinusoidal shape.

### Important Memberfunctions

Obstacle( .. ): Creates an obstacle of a certain size at a certain location.
Update():Updates the location of the obstacle.

## Class: TController

The controller class holds all information with regard to the Line tracking control and the potential field method. Calling the memberfunction update() will calculate the new controller output. Most memberfunctions deal with updating the waypoints and dealing with various controller gains. It can be linked to a list of waypoints for tracking a path and it can be linked to a list of pointers to Obstacles for potential field vector generation.

### Important Memberfunctions

TController(l): Creates a controller object
Init( .. ): Creates initializes the controller, linking it to a particle, initializing the controller gains, and setting the boundary for the line-tracking.
Update(): Updates the controller output based on the position of the particle it is linked to.

GetPotentialVector( .. ): When given a target, a repulsive force, an attractive force and a boundary condition, the calculates the potential field vector at the current location of the linked particle.

## Class: Magnetic Field

The magnetic field class is used to calculate the currents needed to give a certain force vector in the reservoir. There are internal functions that calculate the fields and gradients based on the Polynomial determined by Matlab.

### Important Memberfunctions

GetCurrents(): The argument for this function takes the position where the force vector needs to be calculated, the desired vector, and a pointer to an array in which the calculated current values should be stored. The currents are calculated using a non-linear solver.

## Class: PathPlanner

The PathPlanner class is responsible for the A* path planning. It contains the functions to calculate a walkability map and to plan the route.

### Important Memberfunctions

loadWalkabilityMap( .. ): Calculates the walkability map. Its argument should contain a pointer to an array of Obstacle pointers.
InitializePathfinder(); Sets all registers for path finding.
FindPath(); Finds the path based on a start and end location set by the user.
ReadPath(); Loads the path from the registers.
PostSmooth(); Smoothen the path.
LoadSetPointArray( .. ); The argument is an array of points, and this function copies the points to this array.
EndPathfinder(); clear all registers and free up memory.

# Appendix Markers

In order to determine the absolute orientation and location of the setup with respect to the camera, a marker calibration has been devised. A marker plate can be fixed onto the tower as a substitute for the reservoir. Onto this plate are several dots. Clicking the dot at the center of the plate calculates the shift of the setup with respect to the camera. Clicking a dot to the right of this center determines the angular shift.

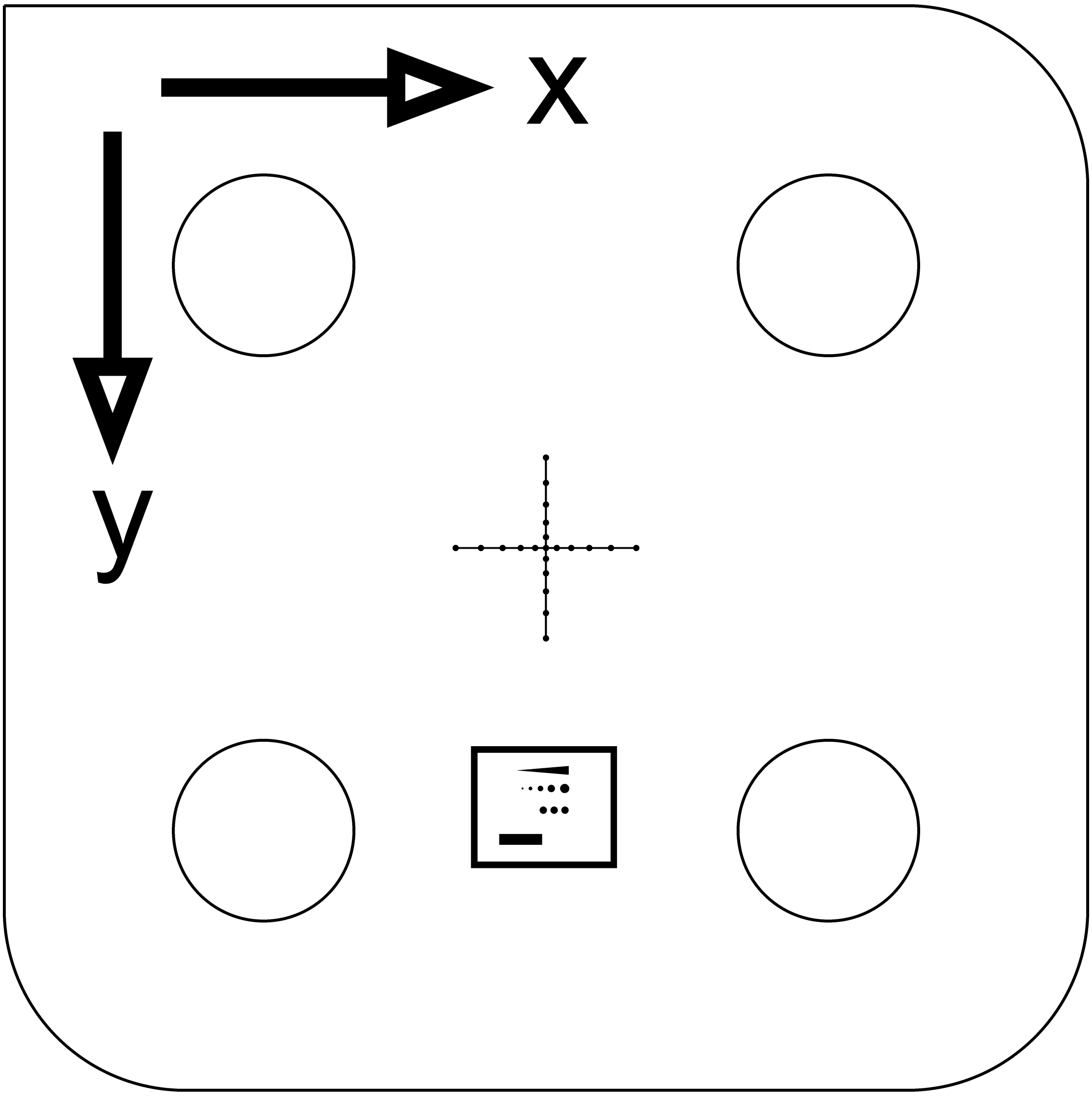



Figure 2: Marker plate used. The marker dot to the right of the center dot and the center dot 300 µm appart

# Appendix Solver issues

Solver Implementation:

Using LevenbergMarquardt algorithm

minimization vector function $F$ is defined as

$$F[0] = 2\alpha V \left[ B_x \frac{\partial B_x}{\partial x} + B_y \frac{\partial B_y}{\partial x} \right] \tag{1}$$

$$F[1] = 2\alpha V \left[ B_x \frac{\partial B_x}{\partial y} + B_y \frac{\partial B_y}{\partial y} \right] \tag{2}$$

$$F[2] = (I \cdot I) * \beta \tag{3}$$

$$F[3] = C \tag{4}$$

V is the particle volume, $\beta$ a constant that gives a weight to the importance of the magnitude of the total current. I is the current vector and C is some constant. The solution is found for the current vector I, containing the four coil currents.

Since the current estimate from Part I was not too bad, this is used as the initial guess for the solution.

$B_x$,$B_y$ and their partial derivatives can be derived from Part II. This seemed to be done successful in software.

The solver does usually find a solution to the problem, however, it is not a stable solution. The particles bounce around on the screen when it is used. Also, often a solution comes up when two opposing coils get equal current values, effectively canceling each other and resulting in no force. This particular solution is a far deviation from the initial guess and makes no sense.

# Appendix Description of reservoir and magnet holders

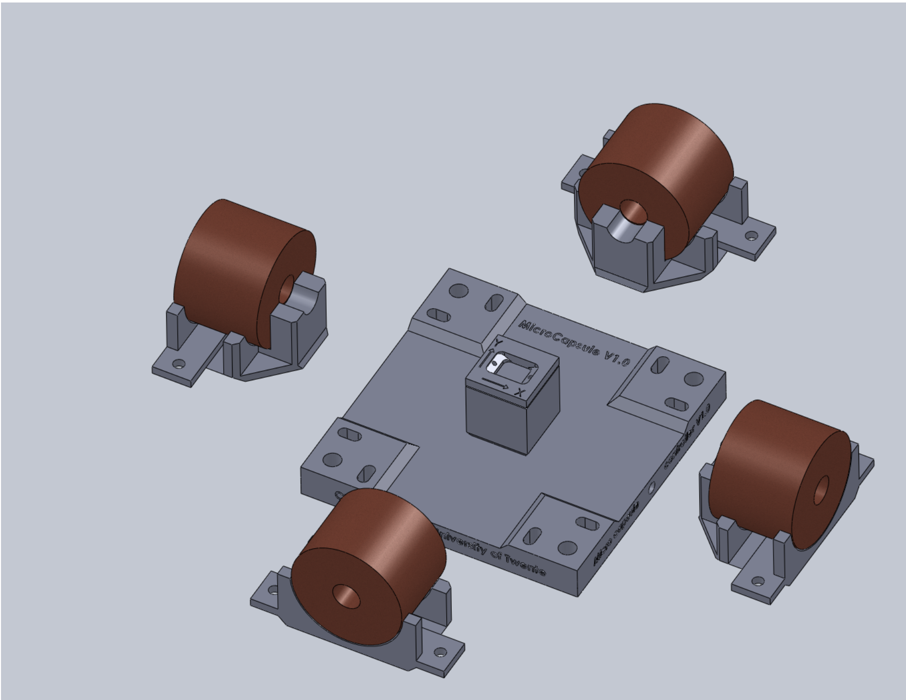
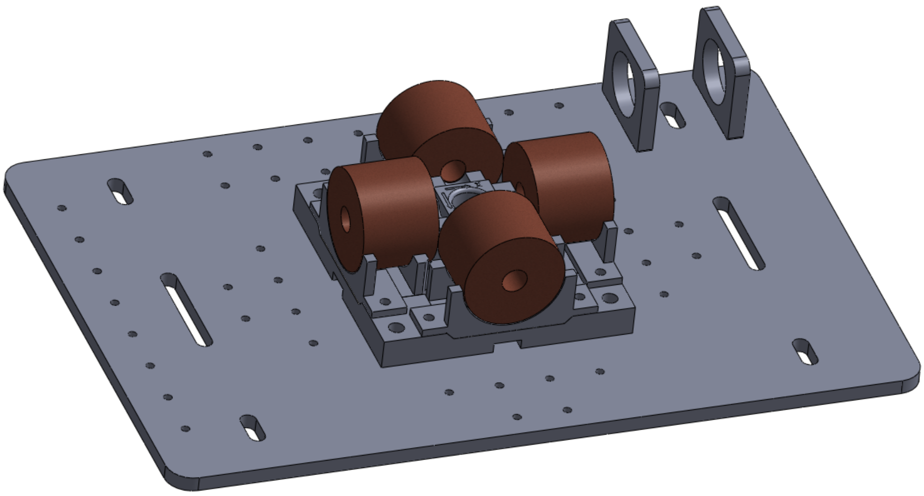
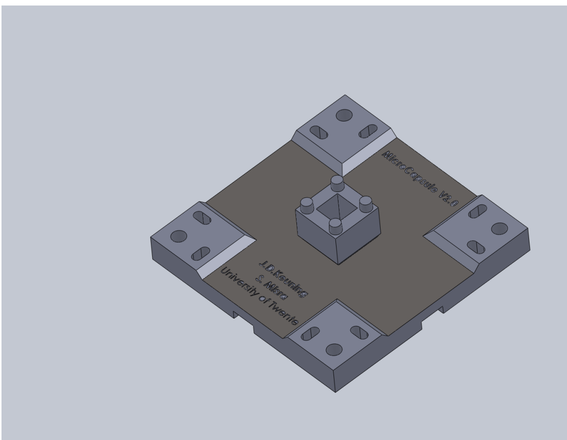


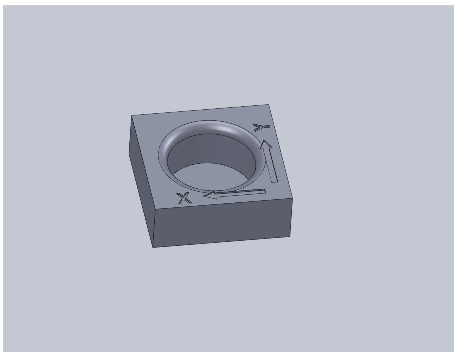

Figure 3: First iteration of the setup. The magnets are fixed in holders and can be placed on the base plate. The relevant dimensions can be found in Part I. All experiments from Part I are done using this setup.



(a)



(b)



(c)

Figure 4: Second iteration of the setup. (a) Shows the setup with reservoir, coils, coilbase and the mounting plate for the microscope. The magnet holders remain the same. The reservoir now is detachable from the base and can be exchanged. The tower on which the reservoir attaches, is hollow, so light can easily be provided from below the base. (c) Shows the reservoir used for the experiments in Part II.

# Appendix Matlab surface fits for B

The $B_x$ and $B_y$ polynomials by Matlab have the structure shown below.

$$\begin{aligned} Poly55(x,y) = & p00 + p10*x + p01*y + p20*x^2 + p11*x*y + p02*y^2 + p30*x^3 + p21*x^2*y + \cdots \\ & p12*x*y^2 + p03*y^3 + p40*x^4 + p31*x^3*y + p22*x^2*y^2 + p13*x*y^3 + p04*y^4 + \cdots \\ & p50*x^5 + p41*x^4*y + p32*x^3*y^2 + p23*x^2*y^3 + p14*x*y^4 + p05*y^5 \end{aligned}$$

Note that due to some decisions Matlab made in terms of the axis, the x in the Poly55 should be replaced with -y and the y needs to replaced with -x. (this is corrected in software)

The coefficients for $B_x$ are:

$$\begin{aligned} p00 &= 0.004088 \\ p10 &= 0.0002972 \\ p01 &= 0.3411 \\ p20 &= -9.089 \\ p11 &= 0.02495 \\ p02 &= 18.04 \\ p30 &= 5.711 \\ p21 &= -1141 \\ p12 &= 0.05776 \\ p03 &= 758.9 \\ p40 &= 1.124 \times 10^4 \\ p31 &= 394.7 \\ p22 &= -7.968 \times 10^4 \\ p13 &= -531.4 \\ p04 &= 2.64 \times 10^4 \\ p50 &= -8.879 \times 10^4 \\ p41 &= 1.546 \times 10^6 \\ p32 &= -1.424 \times 10^5 \\ p23 &= -3.674 \times 10^6 \\ p14 &= -3.522 \times 10^4 \\ p05 &= 5.881 \times 10^5 \end{aligned}$$

the coefficients for $B_y$ are:

$$\begin{aligned} p00 &= -1.233 \times 10^{-6} \\ p10 &= -0.1705 \\ p01 &= -0.0003951 \\ p20 &= -0.003807 \\ p11 &= -18.01 \\ p02 &= -0.03731 \\ p30 &= 286.1 \\ p21 &= 2.245 \\ p12 &= -1125 \\ p03 &= -0.755 \\ p40 &= -45.78 \\ p31 &= 4.017 \times 10^4 \\ p22 &= 41.03 \\ p13 &= -5.217 \times 10^4 \\ p04 &= 529.3 \\ p50 &= -3.584 \times 10^5 \\ p41 &= 4.847 \times 10^4 \\ p32 &= 2.888 \times 10^6 \\ p23 &= -7.912 \times 10^4 \\ p14 &= -1.902 \times 10^6 \\ p05 &= 4.661 \times 10^4 \end{aligned}$$