Development of a Vision Control Feedback Control System for Control of a Micro-robot Via Magnetic Field Manipulation

by

Nisitha Jayatilleka

Submitted to the Department of Mechanical and Industrial Engineering
in partial fulfillment of the requirements for the degree of

Bachelor of Applied Science in Mechanical Engineering

at the

UNIVERSITY OF TORONTO

June 2015

Author .....
Department of Mechanical and Industrial Engineering
April 22, 2015

Development of a Vision Control Feedback Control System for Control of a Micro-robot Via Magnetic Field Manipulation

by

Nisitha Jayatilleka

Submitted to the Department of Mechanical and Industrial Engineering
on April 22, 2015, in partial fulfillment of the
requirements for the degree of
Bachelor of Applied Science in Mechanical Engineering

Abstract

In this thesis a vision feedback control system is designed for the control of a micro-robot in a 2-D environment via the manipulation of a magnetic field. The magnetic field is provided by a Helmholtz coil system.

The micro-robot's movement is tracked by a USB Webcam connected to a computer. The computer reads the position of the robot, calculates the errors made and sends the data to the Arduino which implements a PID controller and a PI controller to minimize the errors.

The system was not fully integrated to the existing physical coil system, and could not therefore be tested as a whole. However individual subsystems were proven to be working through ad-hoc testing methods.

Thesis Supervisor: E. Diller
Title: Professor (MIE)

Acknowledgments

I would like to express my deepest gratitude to my supervisor Prof. Diller, for his guidance, support and most importantly his trust. Allowing me to use the Micro-robotics lab to test out my project as well as trusting me to work on my own with minimal supervision provided me the time to enjoy the project.

I also want to thank Jiachen Zhang for helping me out by teaching me about the existing system, helping me set up my control system and most importantly for letting me use his electromagnetic coil system for testing purposes.

My thanks also go out to Adam Shaw and Ruben Larsson for their enthusiasm and energy which helped motivate me.

Also worth mentioning is Abhinav Ramakrishnan for all his help regarding LaTeX.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Micro-robotics (and nano-robotics) has been a growing field in recent years mainly due to the recognition of advantages of being able to control objects at the micro scale. Hence, micro-robots have been used in diverse fields of engineering including biomedical, manufacturing, and various other instances of general engineering (For example, crack inspection in pipes).

Micro-robots show immense potential in biomedical engineering as they provide a means for non-invasive surgery (reducing the exposure of the patient to germs, reducing the cost for the patient, and increasing the level of privacy), localized delivery of drugs in-vivo and the detection of harmful germs or even early detection of pathogens and cancer.

The OctoMag system is one such system designed for delicate retinal procedures. The system uses an assortment of electromagnets arranged around a workspace to control a micro-robot, giving the micro robot five degrees of freedom (5DOF)[4].

The ability of a micro-robot to traverse blood vessels was demonstrated for the first time in 2006 by the Nano-Robotics Laboratory at ÃĽcole Polytechnique de MontrÃĺal [5]. Cancer cell detection (or the detection of analytes of interest) has been envisioned as a future use

of micro-robots [6].

In the case of micro-assembly and micro-manufacturing using micro-robots, the micro-robot maybe designed such that it has a gripper which can grasp objects. These grippers can be manufactured by using bimorphic materials such as shape memory alloys, piezo materials, and dissimilar metals [7] [8].

An instance of micro-robots being used in a general engineering application would be the use of micro-robots to detect cracks in pipes. In this scenario, a micro-robots equipped with a micro-cameras can be sent into a system of pipes to detect cracks and defects remotely (a similar application would be a colonoscopy conducted using micro-robots [9].

It is clear that micro-robots will be helpful in the advancement of many fields and be of special significance to biomedical engineering.

1.2 Power Sources and Propulsion Methods

Micro-robots present several design challenges due to their small size. First of all it is important to note that at the micro-scale our intuitive understanding of forces can be misleading. This will be discussed in detail later in the report. The second issue is the power source for the micro-robots. Due to their small size, carrying a power source can be difficult and may add limitations such as battery life and increase in size and weight.

Accordingly different methods of energy sources for micro-robots have been identified and experimentally developed. Some of the prominent methods in existence are: through magnetic fields, electric fields, chemical propulsion, focused light beams, radio frequencies, and energy scavenging (generating electricity using the natural vibrations in the environment) among other methods [2].

Micro-robotic actuation achieved through magnetic fields has gained favour over amongst other methods as it overcomes several disadvantages exhibited by other methods. For ex-

ample chemical propulsion and electric field actuation can cause health hazards and limited operating conditions in environments(line of sight needed for light propulsion limits its capabilities in remote environments) [2].

1.3 Scope of the Thesis

When using magnetic propulsion, one of the problems encountered due to its inherently unstable nature is the need to control the micro-robot through a feedback control system. The purpose of this project is to create a vision feedback controls system to guide the micro-robot along a certain path or hold a certain position for a given amount of time.

Recognizing the importance of micro-robotics, and inspired by the success of Robocup the National Institute of Standards and Technology has come up with a competition at the micro-scale named Mobile Microrobotics Challenege (MMC). The competition would involve utilization of micro-electromechanical systems technology along with advanced controls systems to drive the micro-robots. After its initiation in 2007 with the competition name Robocup Nanogram, the MMC has been an annual event held at the IEEE International Conference on Robotics and Automation (ICRA) since 2010 [10].

The 2015 MMC would consist of two challenges. The first challenge involves the micro-robot tracking a predefined micro-scale trajectory multiples times. In this case points are avoided for how accurately the path is being followed by the micro-robot. The second challenge is for the robot to manipulate several micro-scale components inside a chamber within a certain amount of time, simulating application of micro-robotics in real world applications (such as surgery and micro-manufacturing)[10].

The first challenge of the competition has been used as guidelines to scope this thesis. Hence the objective of the thesis is to build a vision feedback controls system capable of driving the micro-robot within a set trajectory in a 2-D environment.

Parameter	Size
d_1	2000 ± 20 µm
d_2	3000 ± 20 µm
d_3	1250 ± 20 µm
t	10 ± 5 µm
b	20 ± 5 µm
θ	45° ± 2° µm

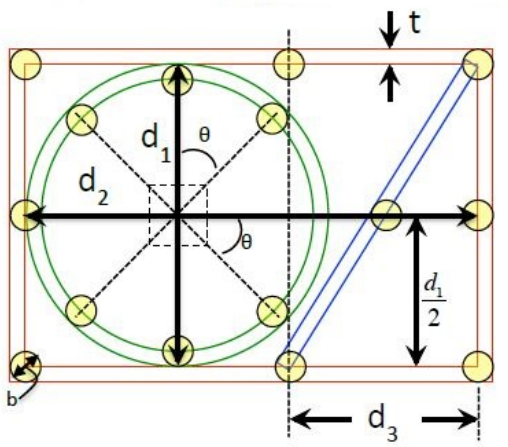


Figure 1-1: Figure showing the arena of the competition [1]. The circles represent the path the robot is to follow

1.3.1 Competition set up

The first challenge of the competition (Autonomous Mobility and Accuracy Challenge) involves an arena within which the basic shapes or trajectories the micro-robot has to traverse are marked by circular markers (Figure 1-1). The surface of the arena will be flat and it is recommended the arena has walls on its boundaries preventing the robot from moving outside. A complete set of rules and guidelines can be found here:[1]

1.4 Plan and Execution

Towards achieving the above goal, research was conducted into forces experienced by the micro-robot under a given set of conditions (assumptions made where necessary to simplify the system). A theoretical model was developed based on the research.

Next a controls system is modelled using Simulink.

Two cameras will be directed at the workspace of the micro-robots to provide feedback on it's current position and trajectory. At first the project will only consider a 2-dimensional (2D) (one camera directed at the X-Y plane) as the system used for this 2D system can be

used for the X-Z plane in a similar manner to give an overall 3D environment.

Afterwards, methods of magnetic field creation, available options for implementing the controls system, available cameras, and vision algorithms suited for the system will be explored. Next the systems will be integrated and implemented. Finally, the system is tested and results are obtained for it's accuracy.

The project conclusion will be given at the end of this report and future recommendations will be provided on how the system can be improved.

Chapter 2

Theoretical Modelling

An analysis of the forces that act upon the micro-robot will be conducted in this section. It is understood that forces acting on bodies can be size-dependant due to scaling factors. For example, the surface area and volume of a robot depends on L^2 and L^3 if its length is L, meaning that the surface area-to-volume ratio would increase at the micro-scale. Hence, any and all forces that vary with area and volume will have a larger effect on the micro-robots motion [2]. These forces are recognized in Figure 2-1.

Even though the competition only requires 2-D motion, the scope of the thesis is to build a system which can transcend the 2-D world and move into the 3-D world in the future. Keeping this in mind, fluid drag forces, weight and buoyancy forces were recognized as the

Length dependent ($\propto L$)	Surface tension
Area dependent ($\propto L^2$)	Surface forces, fluid drag, friction, Reynolds number (Re), evaporation, fluid drag transient time τ (low Re), heat conduction, electrostatic forces
Volume dependent ($\propto L^3$)	Mass, inertia, heat capacity, buoyancy

Figure 2-1: Scaling of different forces dependent on length (perimeter), surface area, and volume. Image taken from [2].

important forces. Due to the fact that robot would be controlled through application of magnetic fields, the force exerted by these fields are identified and explained here.

2.1 Drag Forces

In the real world application of micro-robots, the operating environment can range from air to blood to various other liquids.

The drag force is expressed as

$$F_d = \frac{1}{2} C_d \rho_f A v^2 \tag{2.1}$$

where C_d is the drag coefficient which is dependent on the shape of the object, ρ_f is the density of the fluid, A is the cross sectional area of the object, and v is the velocity of the object relative to the flow velocity. C_d depends on the Reynods Number, defined as the ratio of inertial forces to viscous forces (providing a measure of which force dominates). For a spherical object, experiencing laminar flow ($Re < 0.1$) C_d can be approximated as

$$Re = \frac{\rho_f D v}{\mu_f} < 0.1 \tag{2.2}$$

$$C_d = \frac{24}{Re} \tag{2.3}$$

using the above, and substituting for the surface area of the spherical robot, Equation 2.1 can be simplified to

$$F_d = 3\pi \mu_f D v \tag{2.4}$$

Further, it is important to note that the velocity of the object relative to the flow, v would depend on flow velocity greatly. While we understand laminar flow, we can only roughly approximate the flow for turbulent conditions. Even though laminar flow has been well studied and understood, this thesis will assume a stationary environment (no flow conditions). This would mean that v is the velocity of the robot itself.

Similarly, drag torque will be considered negligible due to the fact that the spherical shape of the robot would mean that rotation would be meaningless in most scenarios with the notable exception of rolling on a surface. On considering micro-robots with other geometries, the drag torque experienced could be substantial.

Brownian motion and non Newtonian fluids are considered outside the scope of this thesis.

2.2 Weight and Buoyant Force

As the micro robot is travelling within a fluid, it is necessary to recognize the weight and the buoyant force experienced by the robot in order to apply a suitable magnetic force to move or hold position in a 3-D environment. For a 2-D system, the buoyant and weight forces do not hold meaning.

The weight of the robot will depend on the size and the density of the material it is made from. The buoyant force will depend on the density of the surrounding fluid. Hence the total downward force acting on the micro-robot submerged in a fluid can be calculated using

$$\vec{F_b} = V_m(\rho_r - \rho_f)\vec{g} \tag{2.5}$$

where $V_m = \frac{4\pi r^3}{3}$ is the volume of the spherical robot, ρ_r is the density of the robot, ρ_f is the density of the fluid, and g is the gravitational acceleration.

2.3 Friction

Friction acting on the micro-robot can be a considerable force in a 2-D system. Friction between two surfaces occur due to variations in the evenness of the surface. A highly even surface is considered smooth. The surface quality, represented by μ is an intrinsic property for each surface which depends on the surface finish. The frictional force acting on an object (always in the direction opposite to it's motion) is hence given by

$$F_f = \mu_f R = \mu_f mg \tag{2.6}$$

where R is the normal reaction force to the surface. It is important to note that if the micro-robot is placed in a dense liquid, the buoyancy force would come into play and the above equation will no longer hold.

Due to the small size of the micro-robot, the surface it 'sees' can be considered very rough to the point where the surface protrusions might be of a similar scale as the robot itself. One way to overcome this issue somewhat, is to apply a torque on the micro-robot such that it rolls forward rather than gets dragged. The difference between static friction and dynamic friction is considered negligible throughout this project.

2.4 Magnetic Force

Magnetic materials are separated into three categories depending on susceptibility (or the response to externally applied magnetic fields): diamagnetic, paramagnetic and ferromagnetic materials. Diamagnetism is the opposition to an applied magnetic field, hence the forces acting on these materials due to external magnetic forces are very weak. Diamagnetism is marked by a very low negative susceptibility. Paramagnetic materials have positive susceptibility values larger than those of diamagnetic materials and as such experience larger forces due to magnetic fields. Lastly, ferromagnetic materials have the highest susceptibility.

Ferromagnetic materials can be subdivided into hard and soft materials. This is due to the fact that some materials (hard) exhibit a hysteresis effect by retaining their magnetization after the external field has been removed. Soft materials have a higher permeability and are easier to magnetize and demagnetize than hard materials. The hysteresis effect can lead to undesired application of force on the robot.

The flux density of a magnetic field is given by

$$\vec{B} = \mu_o(\vec{M} + \vec{B}) \tag{2.7}$$

where $\mu_o = 4\pi \times 10^{-7} Tm/A$ is the magnetic permeability of free space, $\vec{B}(A/m)$ is the external magnetic field strength, $\vec{M}(A/m)$ is the magnetization of the micro-robot, and $\vec{B}(Tesla)$ is the magnetic flux density. For a linear, isotropic and homogeneous media $\vec{M} = \chi\vec{H}$ and this simplifies Equation 2.7 as

$$\vec{B} = \mu_o(\chi\vec{H} + \vec{H}) = \mu_o\mu_r\vec{H} \tag{2.8}$$

The magnetic force and torque exerted on the micro-robot can be given by

$$\vec{F_m} = V_m(\vec{M} \cdot \nabla)\vec{B} \tag{2.9}$$

$$\vec{\tau_m} = V_m\vec{M} \times \vec{B} \tag{2.10}$$

$$\tag{2.11}$$

It is of importance to note that the force experienced by the robot depends on the gradient of the flux $\vec{B}$ and the torque depends on the magnitude of the curl $\vec{B}$.

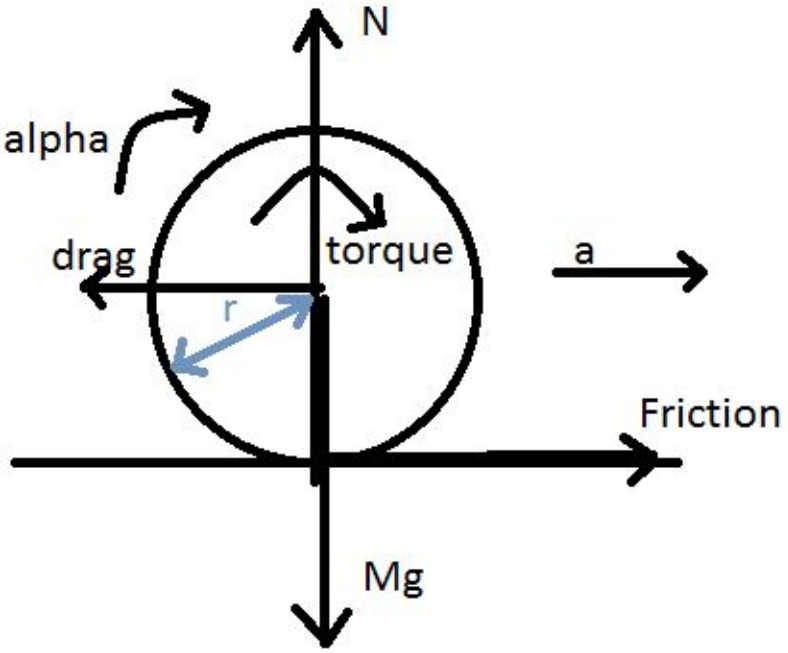


Figure 2-2: Figure showing the force diagram of a micro-robot made to roll via aapplication of a magnetic torque τ_m. It's weight is shown by Mg, acceleration by a, angular acceleration by $alpha(\alpha)$, fluid drag by $drag(F_d)$, friction by F, normal force by N, and radius by r

2.5 Resultant System

The resultant system from these forces on a micro-robot made to roll by applying a magnetic torque is as follows Figure 2-2.

$$N - Mg = 0 \tag{2.12}$$

$$F - F_d = Ma \tag{2.13}$$

$$\tau_m - FR = I\alpha \tag{2.14}$$

where I is the polar moment of inertia of the sphere.

Substituting $I = 2MR^2a/5$ and $\alpha = a/R$ and simplifying the above equations gives,

$$a = \ddot{x} = \frac{5(\tau_m - F_d rag R)}{7RM} \tag{2.15}$$

Rotational damping torque and adhesive forces are neglected here to simplify the model.

Calculation of the necessary magnetic field strengths $\vec{B}$ necessary to apply a certain force $\vec{F_{total}}$, is taken care of by a third party and is hence outside the scope of this thesis.

Chapter 3

Control System Design and Simulation

The control system is what guides the micro-robot along its desired path. Therefore, it is first necessary to determine how the path would be input or implemented in the system and the algorithm that would be used to guide the robot as the design of the control system would be dependent on this.

3.1 Path Following Algorithm

For now the project will assume that the path is input by the user as a function $y = f(x)$, a continuous function Figure 3-1. The control system operates at a given frequency, (depending on the frequency of the tracking camera, the frequency of the computer, the frequency of the Arduino Board and the frequency of the other circuitry used to drive the coil system. Therefore obtaining discrete points of data at given intervals of time so that the real position of the micro-robot can be matched against those is essential.

It is therefore necessary to obtain the set of data points in x, y coordinates, that lie on the input function. In the figure, these points are shown as blue dots. The micro-robot needs to follow the path mapped out by these data points as closely as possible. To achieve this, the following algorithm was designed.

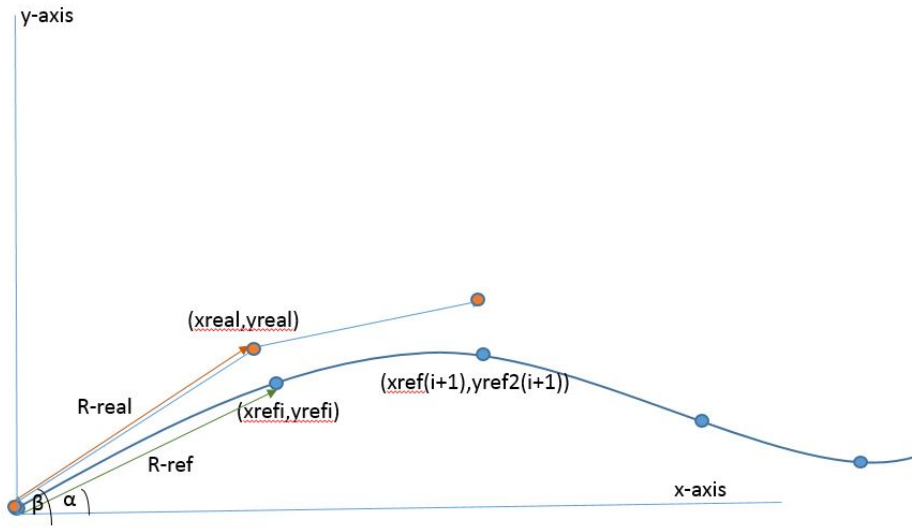


Figure 3-1: Figure showing the path or the data points being followed by the micro-robot. $(xreal, yreal)$ denotes the position of the robot now where as $(xref(i), yref(i)$ denotes where the robot should be. Likewise, $(xref(i+1), yref(i+1)$ shows the next target position of the micro-robot.

The algorithm accounts for the error in two ways. First it calculates the magnitude of the overshoot or the undershoot (i.e. $RErr = Rref - Rreal$) and feeds this value to the controller (which will be discussed later). Second, it calculates the directional error made by the micro-robot by taking the difference of the angles, i.e. $dirErr = \alpha - \beta$. This value is also sent to the controller.

3.1.1 Initialization and Termination

It should be noted that the user should also input the starting coordinates and the finishing coordinates of the path along with the function that defines the path. These would help guide the robot in a certain direction along the path. However it should be noted that the starting coordinates of the micro-robot input to the system, cannot be matched exactly by the initial positioning of the micro-robot due to the fact that it is introduced the system manually.

To overcome this, the system generates consequent closely timed torques which would cancel each other out. This would vibrate the micro-robot allowing the camera to determines its initial position. Next the robot would be guided towards the starting point defined by the user along a straight line. It should also be noted that this vibration will reduce the adhesive forces between the micro-robot and the surface and thereby reduce errors [2].

After all the data points have been parsed by the algorithm, it will generate the end point coordinates repeatedly, bringing the micro-robot to the desired end point, similar to the method discussed above.

3.2 Control System Selection

The error $RErr$ is a result of the micro-robot overshooting its desired position. This happens due to the fact that it does not slow down until it reaches the target position (arrives with a velocity) which is very similar to the speed control used with vehicles. Applying a Proportional -Integral-Derivative (PID) feedback controller here seems the best option as it takes care of the magnitude of the previous error (through proportional control), the accumulative error over the entire time (through integral control) and prevents overshoot (through derivative control). From here on, this controller will be referred to as the Magnitude controller

Similarly, a Proportional-integral (PI) feedback controller can be implemented to control the directional error $dirErr$. The reason for excluding the derivative controller part here is due to the fact that the micro-robot used is a sphere. Hence the robot never really turns or rotates, but rather, starts rolling in another direction. This controller will be referred to as the Directional controller from

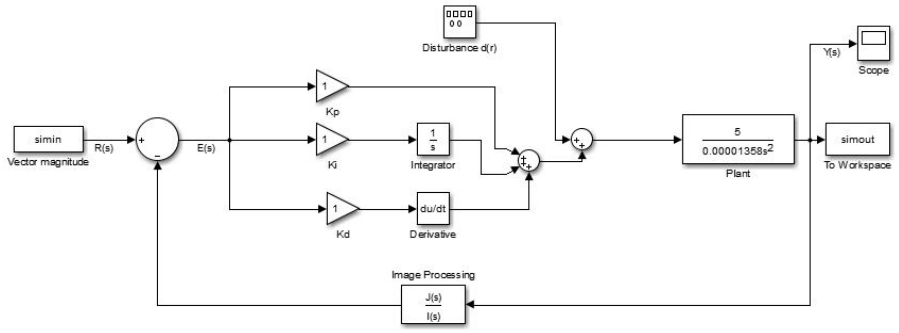


Figure 3-2: The Simulink model for the Magnitude controller

3.3 Model

3.3.1 The Magnitude Controller PID

Based on the theoretical model constructed in 2.15, the transfer function can be derived by taking the Laplace Transformation of the equation (notice that the fluid drag force term F_dragR) can be removed and considered as an external disturbance in the system) and rearranging it as:

$$H(s) = \frac{X(s)}{\tau_m(s)} = \frac{5}{7RMs^2} \tag{3.1}$$

Here, an assumption can be be made that the micro-robots density was $\rho_r = 7400kg/m^3$ (assuming the micro-robot is made of neodymium) and along with $R = 500\mu m$, which is the size of the robot allowed in the competition, the mass of the micro-robot was calculated to be $M = 3.88mg$.

A Simulink model is created and the values are substituted Figure 3-2.

INSERT PICTURE OF MODEL rerror

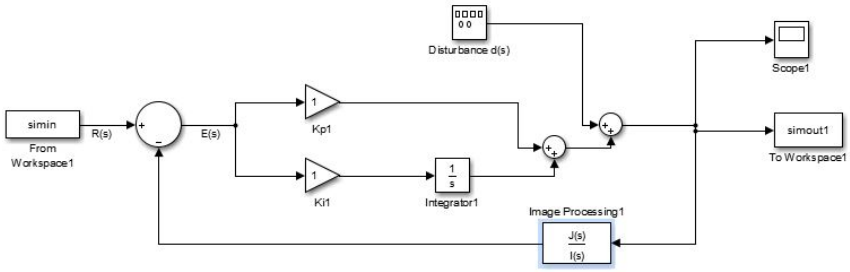


Figure 3-3: The Simulink model for the Directional controller

3.3.2 The Directional Controller PI

In this model, the PI controller will compare the two angle values α and β and then send the error through the PI controller. The user set controller values of K_p and K_i will determine the 'importance' or the weight applied to each error source (proportional or cumulative respectively).

INSERT PICTURE OF MODEL direrror

Chapter 4

Available Methods of Implementation

4.1 Magnetic field generation

Magnetic fields are the energy source of the micro-robot in this project. Hence, creating a reliable system that produces a magnetic field is of paramount importance. The existing methods for generating magnetic fields are quite extensive. Since the idea is to integrate the control with a magnetic field generation system built by a third party, the project will only be exploring the two possible coil systems that it will eventually be integrated with.

4.1.1 Helholtz Coils

A Helmholtz coil system is one way of producing a uniform magnetic field over a workspace for micro-robots. The coil system uses pairs of coils positioned along the X,Y and Z axes. The system is shown in Figure 4-1. One major drawback of the system is that the field created is uniform throughout the workspace and hence exerting a force (push or pull) on the micro-robot is not an option given that that requires a gradient in the field as proven by 2.9. Therefore, micro-robot locomotion is achieved by applying torques using the uniform field created, according to 2.10.

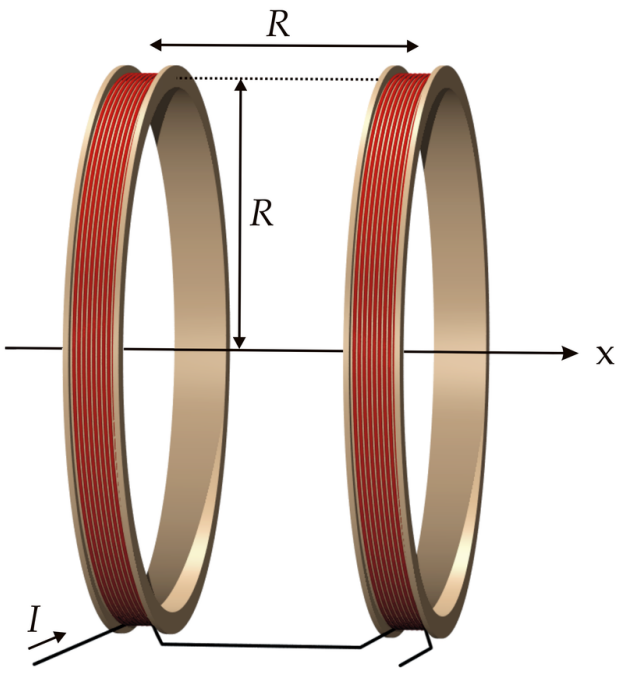


Figure 4-1: Figure showing the schematic of a Helmholtz coil pair [3].

The uniformity of the field can be further improved by adding more coils to the system along the same axes. This would lead to a set up known as the a Maxwell coils configuration. The trade off is that the system becomes more complex. The coil configuration found in the Micro-robotics Laboratory is a Helmholtz configuration.

The field generated can be changed by varying the current through each of the three pairs of coils. The robot can be rolled along a certain axis by providing sine wave and cosine wave currents to the coils along the other axes.

$$B(\vec{x}) = \frac{(\mu_o I R^2)}{2(R^2 + x^2)^{3/2}} \tag{4.1}$$

where $\mu_o = 4\pi \times 10^{-7} T \cdot m/A$ is the magnetic permeability of free space,

4.1.2 Permanent Magnet System

Another method of generating a variable magnetic field is through the manipulation of permanent magnets. The system currently being built at the Micro-robotics lab at University of Toronto uses permanent magnets mounted on stepper motors. The stepper motors can rotate the magnets with high precision and thereby create minute changes in the magnetic field as necessary.

The upside of using this system is that it has the ability to generate forces as well as torques. However, optimizing the system (amount of magnets used, mounting positions, mounting orientations) is time consuming and calculating the forces and torques applied on the micro-robot can be a complex process. Further, since the system uses motors, magnetic noise will be greater than that of the Helmholtz coil system.

4.2 Controller Implementation

The next decision to be made is the selection of the controller for the implementation of the control system. The PID and PI controllers used in this project can be implemented by a various number of devices with varying capabilities, such as Programmable Interrupt Controllers (PICs), Arduino Boards, Field Programmable Gate Arrays (FPGAs), and computer controlled systems through the use of software like MATLAB and LABVIEW.

Out of these options, PICs and Arduino Boards (based on Atmel AVR chips) are the cheapest and the most available. FPGAs are more expensive in general and can be considered excessive for the task at hand. The disadvantage faced by these is the inability to process complex vision algorithms and math which might be required for the project. One way to overcome this would be to implement a small computer system like Raspberry Pi or Beaglebone which would lead to a costly system.

More complex systems using MATLAB and LABVIEW (along with other necessary equip-

ment) are very expensive. Also, a major factor against the use of these methods is the fact that these would be harder to integrate with the existing system at the Micro-robotics lab. However, it should be noted that these are industry standards and the final result can be much more accurate and easier to obtain. For example MATLAB has built-in vision algorithms (through the Computer Vision System Toolbox) and complex controls system implementation tools available through Simulink. Arduino was chosen over the PIC for cost efficiency.

Accordingly an Arduino Mega2560 board along with a personal laptop computer will be used to implement the PID controller.

4.3 Vision Feedback

Initially, it was decided that the Machine Vision Cameras already in place at the Micro-robotics lab would be used to obtain information regarding the robot. However, extracting the vision input from the cameras and integrating to the system built for the thesis proved to be time consuming as the existing system was complex. Owing to this, a USB webcam was bought in order to receive vision input (Logitech HD Webcam c270).

Chapter 5

Design of the System

5.1 Components of the System (Overview)

The main components of the system and how they connect can be seen in Figure 5-1. As can be seen the controls system is to be implemented alongside a Helmholtz coil system. The coil system is powered by a DC supply connected to a set of circuitry (drivers). The Arduino Mega2560 from the built system will connect to these drivers which will act as amplifiers and power up the coil system. The coil system will drive the micro-robot as commanded by the Arduino. The motion will be recognized by the Logitech Webcam connected to the computer via USB. The computer reads realtime video from the camera and tracks the robot. It also calculates the Magnitude error and the Directional error and outputs the data to Arduino which in turns runs the PID and the PI controllers and redirects the robot as necessary by providing signals to the drivers.

5.2 Computer Programming

The computer program contains four files Figure 5-2.

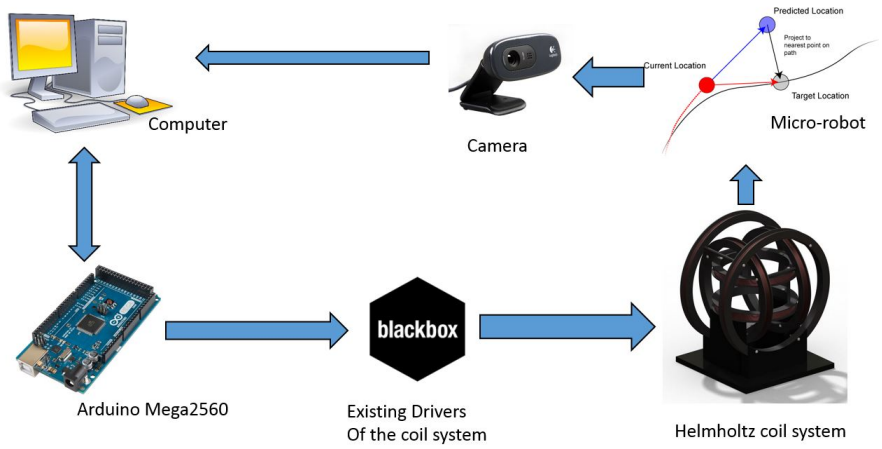


Figure 5-1: Figure showing the connectivity of the components of the system

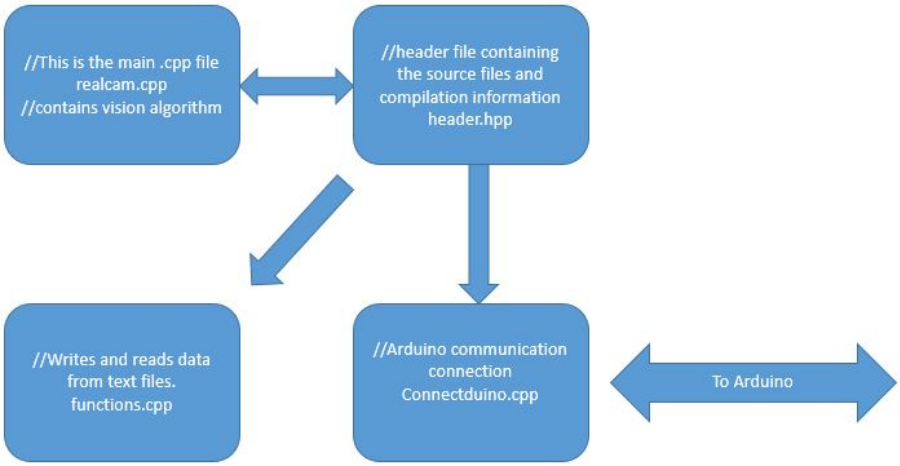


Figure 5-2: Figure showing the connectivity of the components of the system

```
//read first frame
capture.read(nisiframe1);
//convert frame1 to gray scale for frame differencing
cv::cvtColor(nisiframe1,grayImage1,COLOR_BGR2GRAY);
//copy second frame
capture.read(nisiframe2);
//convert frame2 to gray scale for frame differencing
cv::cvtColor(nisiframe2,grayImage2,COLOR_BGR2GRAY);
//perform frame differencing with the sequential images. This will output a
//do not confuse this with a threshold image, we will need to perform thres
cv::absdiff(grayImage1,grayImage2,differenceImage);
//threshold intensity image at a given sensitivity value
cv::threshold(differenceImage,thresholdImage,SENSITIVITY_VALUE,255,THRESH_B
if(debugMode==true){
    //show the difference image and threshold image
    cv::imshow("Difference Image",differenceImage);
    cv::imshow("Threshold Image", thresholdImage);
}else{
    //if not in debug mode, destroy the windows so we don't see them anymor
    cv::destroyWindow("Difference Image");
    cv::destroyWindow("Threshold Image");
}
```

Figure 5-3: Figure showing the connectivity of the components of the system

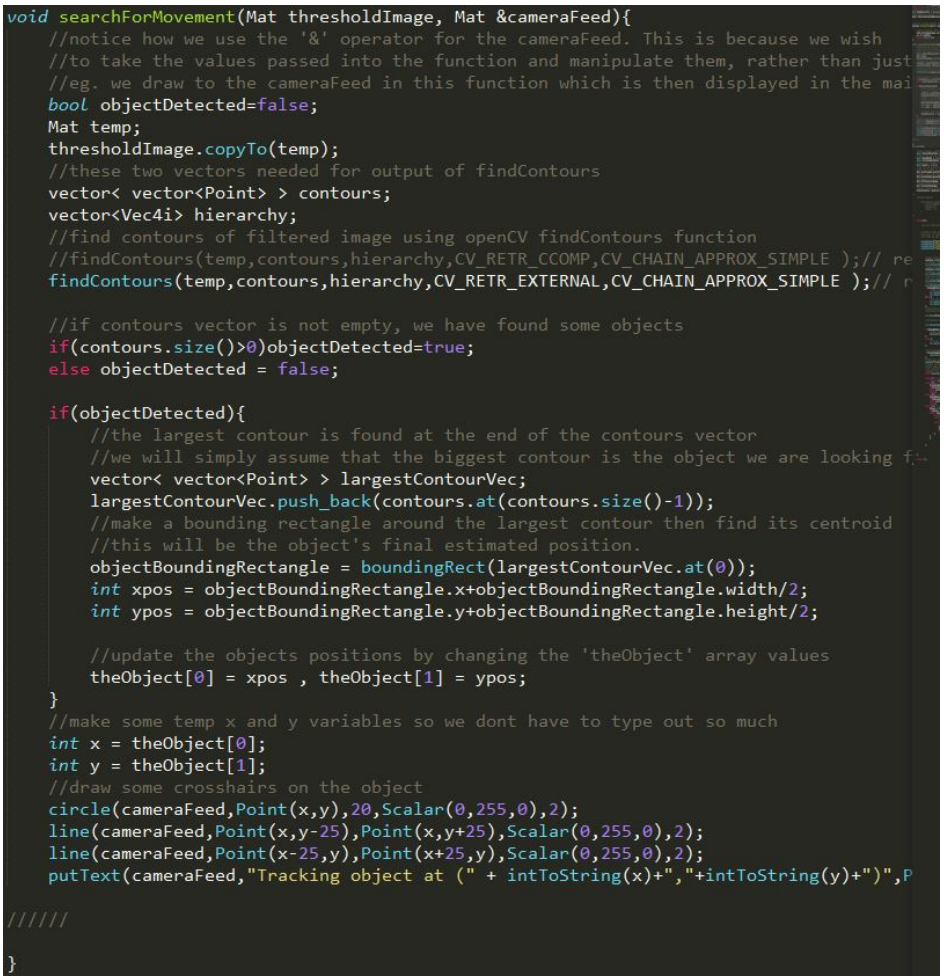
5.2.1 realcam.cpp

The main function is included in the realcam.cpp file. This file includes the vision algorithm responsible for monitoring the robot. It is linked to the opencv libraries.

The tracking algorithm used is a absolute difference of frames method. The main reason for this is the fact that its a very reliable method for tracking a single object that is moving against a still background.

The algorithm reads two consequent frames from the video and after converting the images to grey-scale (Figure 5-3), compares the matrices obtained. Here it recognizes the differences between the two matrices. The opencv function 'boundingRect' is used to obtain the vertices of a rectangle that circumscribes the micro-robot. For the ease of the user, the function draws out a target over the video Figure 5-4.

It is important to note that the centre coordinates of the robot gets written to an external variable which will be input the 'xreal' and 'yreal'.



```
void searchForMovement(Mat thresholdImage, Mat &cameraFeed){
    //notice how we use the '&' operator for the cameraFeed. This is because we wish
    //to take the values passed into the function and manipulate them, rather than just
    //eg. we draw to the cameraFeed in this function which is then displayed in the mai
    bool objectDetected=false;
    Mat temp;
    thresholdImage.copyTo(temp);
    //these two vectors needed for output of findContours
    vector< vector<Point> > contours;
    vector<Vec4i> hierarchy;
    //find contours of filtered image using openCV findContours function
    //findContours(temp,contours,hierarchy,CV_RETR_CCOMP,CV_CHAIN_APPROX_SIMPLE );// re
    findContours(temp,contours,hierarchy,CV_RETR_EXTERNAL,CV_CHAIN_APPROX_SIMPLE );// r

    //if contours vector is not empty, we have found some objects
    if(contours.size()>0)objectDetected=true;
    else objectDetected = false;

    if(objectDetected){
        //the largest contour is found at the end of the contours vector
        //we will simply assume that the biggest contour is the object we are looking f
        vector< vector<Point> > largestContourVec;
        largestContourVec.push_back(contours.at(contours.size()-1));
        //make a bounding rectangle around the largest contour then find its centroid
        //this will be the object's final estimated position.
        objectBoundingRectangle = boundingRect(largestContourVec.at(0));
        int xpos = objectBoundingRectangle.x+objectBoundingRectangle.width/2;
        int ypos = objectBoundingRectangle.y+objectBoundingRectangle.height/2;

        //update the objects positions by changing the 'theObject' array values
        theObject[0] = xpos , theObject[1] = ypos;
    }
    //make some temp x and y variables so we dont have to type out so much
    int x = theObject[0];
    int y = theObject[1];
    //draw some crosshairs on the object
    circle(cameraFeed,Point(x,y),20,Scalar(0,255,0),2);
    line(cameraFeed,Point(x,y-25),Point(x,y+25),Scalar(0,255,0),2);
    line(cameraFeed,Point(x-25,y),Point(x+25,y),Scalar(0,255,0),2);
    putText(cameraFeed,"Tracking object at (" + intToString(x)+","+intToString(y)+")",P

//////

}
```

Figure 5-4: Figure showing the connectivity of the components of the system

5.2.2 functions.cpp

function generateref

This is the user's input method to the program. The user can input the starting coordinates and ending coordinates into the program as well as the the function $y = f(x)$.

Using the GiNaC libraries [11], a symbolic function is input to the system. Taking user input for the number of reference points to be generated, the function plugs these values into the function and write the 'xref' and 'yref' values to a file called "datapoints.txt".

function readFile and writeFile

These functions are included so that the real coordinates of the micro-robot can be written to file (as the system is running) and to read the stored data from file. The readFile function is not being used for this project but was made with future uses in mind such as graphing the robots movement as it is moving.

5.2.3 connectduino.cpp

This function is responsible for opening and maintaining communications with the Arduino board through the USB serial port. This is done by using the SerailStream.h header (includes the Serial library) which provides functions for connecting to Arduino through c++ programs.

The function sets a bits per second rate (9600) which has to matched by the Arduino in order for proper communication.

Figure 5-5: Figure showing the connectivity of the components of the system

5.3 Arduino Programming

The Arduino is programmed using Arduino 1.6.3 Integrated Development Environment (IDE). External libraries used include the PID library [12] to implement the PID controller, the PI controller, and the Messenger library [13].

The Messenger library is used to read the data sent over the USB serial connection. An important function of the library is to buffer the data so as to allow for better synchronisation for data transmission.

The first bytes received will provide the starting coordinates of the robot. From here on the system will follow the path folowing algorithm outlined in section 3.1.

Chapter 6

Testing, Results, and Future Recommendations

The integration of the system with the existing equipment at the Micro-robotics laboratory was met with difficulty and eventually failed (due to lack of time). However, the vision algorithm was tested out using the existing control system while reading data from the Webcam. Further this was also tested out using a recorded video Figure 6-1.

The communication between the Arduino board and the computer was tested out using a simple program which when given a command through the Linux Terminal would make an LED blink a specified number of times.

The void generateref() function was tested out by using inputs $xstart = 0$ and $xend = 600$.

Even though the above functions of the system were proven to be working, the overall functionality of the system could not be tested due to inability to integrate with the existing coil system.

Several ways of improving the system exist (without changing the hardware set up). First, the vision algorithm could be improved with the addition of another algorithm which would recognize the edges of either the workspace itself or obstacles. This would allow the micro-

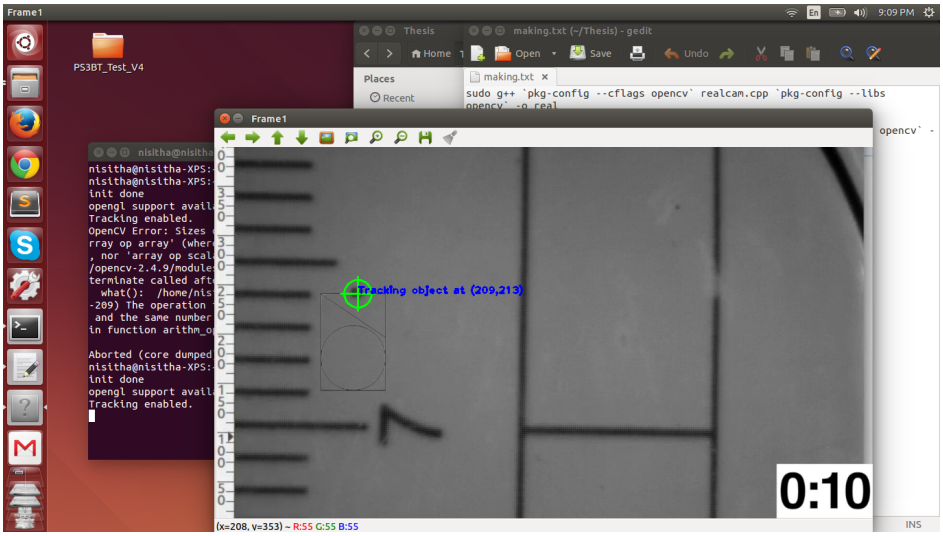


Figure 6-1: Figure showing the connectivity of the components of the system

robot to avoid collisions. Second, the control system could be improved to suit different conditions (such as edge effects and drag torque). Auto tuning of the PID controller and the PI controller can be handled through the Arduino library "Autotune PID".

Chapter 7

Conclusion

The goal of this thesis was to build a vision feedback control system for the control of a micro-robot via magnetic field manipulation. In setting the scope of the thesis, the competition requirements for the Mobile Microrobotics Challenge's Autonomous Mobility and Accuracy Challenge were used as guidelines. Hence the scope of the thesis was set to controlling a micro-robot in a 2-D environment.

After setting the scope, a theoretical model was constructed for the movement of a spherical micro-robot made from a neodymium magnet. Next, a path following algorithm was designed and using this, the strategy for the implementation of the controller was reached.

There were two controllers, one occupied with minimizing the displacement errors of the robot (overshooting the target) and the other occupied with correcting the direction of the robot. These systems were Modelled in Simulink.

Implementation of the system required selection of several components of the system including a camera for computer vision and an Arduino Board for implementing the controllers.

The vision algorithms for the system were written using the c++ opencv library. The vision algorithm takes in two frames (images) from the camera and compares them. It notices the changes between the two images and recognizes this as movement and tracks

the object. The downside to this is that the background of the image has to be motionless. Hence at high sensitivity levels vibrations in the surrounding environment might interfere with the tracking.

The coordinates of the robot's actual position is recorded by the computer and the error values for magnitude and direction are calculated using the reference values of the robots path. These eror values are then forwarded to the Arduino which implements the controllers and sends the output to the coils system's driving circuit boards.

Upon completion the system was to be implemented with a physical coil system (made by a third party) and tested out. However integration with the existing system proved to be complicated and due to lack of time, the project was stopped.

However a number of programming functions were tested out individually. The vision algorithm, communication with the Arduino, and the creation of path and thereby, reference values were proven to be working.

Recommendations for the future steps of the include the addition of another vision algorithm capable of recognizing terrain (mazes and other obstacles) and the implementation of a more flexible control system which will adapt to the environment of the robot by automatically tuning it's values.

Bibliography

[1] Sites.google.com, 2015.

[2] Eric Diller and Metin Sitti. Micro-scale mobile robotics. *Foundations and Trends in Robotics*, 2(3):143–259, 2011.

[3] Ansgar Hellwig. *Helmholtz coils*. 2005.

[4] Michael P. Kummer, Jake J. Abbott, Bradley E. Kratochvil, Ruedi Borer, Ali Sengul, and Bradley J. Nelson. Octomag: An electromagnetic system for 5-dof wireless micromanipulation. *IEEE Transactions on Robotics*, 26(6):1006–1017, 2010.

[5] J.-B. Mathieu, G. Beaudoin, and S. Martel. Method of propulsion of a ferromagnetic core in the cardiovascular system through magnetic gradients generated by an mri system. *IEEE Transactions on Biomedical Engineering*, 53(2):292–299, 2006.

[6] Sylvain Martel. Bacterial microsystems and microrobots. *Biomedical Microdevices*, 14(6):1033–1045, 2012.

[7] M.B. Khamesee, N. Kato, Y. Nomura, and T. Nakamura. Design and control of a microrobotic system using magnetic levitation. *IEEE/ASME Trans. Mechatron.*, 7(1):1–14, 2002.

[8] P Dario, R Valleggi, M C Carrozza, M C Montesi, and M Cocco. Microactuators for microrobots: a critical survey. *Journal of Micromechanics and Microengineering*, 2(3):141–157, 1992.

[9] I. Kassim, L. Phee, W.S. Ng, Feng Gong, P. Dario, and C.A. Mosse. Locomotion techniques for robotic colonoscopy. *IEEE Eng. Med. Biol. Mag.*, 25(3):49–56, 2006.

[10] Sites.google.com. Mobile microrobotics challenge, 2015.

[11] Ginac.de. Ginac, an open framework for symbolic computation within the c++ programming language, 2015.

[12] Brett Beauregard. Arduino playground - pidlibrary, 2015.

[13] Thomas Fredericks. Arduino playground - messenger, 2015.