GEORGE MASON UNIVERSITY | Bioengineering

Weinberg MEDICAL PHYSICS

MAGNETO

Control Interface for Autonomous Delivery of Magnetically Stimulated Particles

Fall 2018 - Spring 2019

Weinberg Medical Physics Sponsors
Dr. Irving Weinberg, Dr. Lamar Mair
Dr. Chad Ropp, Olivia Hale

GMU Senior Design Team
Victor Huynh, Bassam Mutawak,
Elizabeth Ankrah, Minh Quan Do

GMU Advisors
Dr. Nathalia Peixoto, Dr. Qi Wei

Background - Magnetic Particle Delivery

- In vivo transport of magnetic nanocarriers via external magnets

- Potential for noninvasive surgery (carriers deliver therapeutic payloads)

- Current limitations:
  - Physiological barriers
  - Magnetic field attenuation
  - Manual operation of magnet arrays

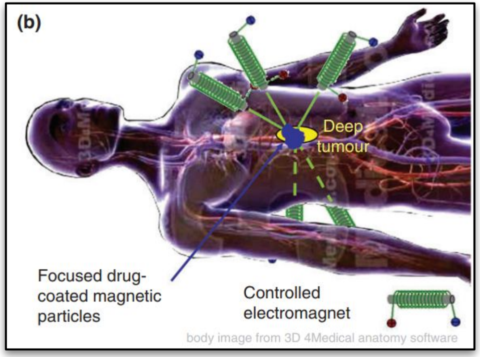


Fig 1. Concept to use magnetic fields for focusing therapy to a deep tumour [1].

Problem Definition

To overcome delivery limitations, Weinberg Medical Physics (WMP) is designing an MRI-guided magnetic control system (MCS). Automation of system operation is key for practical MCS implementation.

Project Magneto lays MCS foundation for automation behavior, control flow, and user interaction.

Project Relevance

Magneto is a C++ software interface that controls WMP’s 4-coil array system for autonomous delivery of a magnetic object. It serves to:

1. Be a software precursor to WMP’s future surgical MCS.

2. Improve control of the 4-coil array system used in WMP’s pilot experiments.

Project Objectives

1. Design software control interface with relevant, easily portable features for WMP’s future MCS

2. Automate and improve accuracy of particle translation using WMP’s 4-coil array system

Requirements

Graphic User Interface (GUI):

- Optical Image Streaming
- Consistent Particle Detection
- User Path Drawing
- Operation Automation and Controls
- Data Collection and Exporting

Particle Control

- Consistent Particle Translation Along Path
- Simultaneous Control of Multiple Hardware Components
- Bidirectional Communication (Hardware with GUI)

GUI Metrics

Speed	Functionality
Definition: Time needed to execute key functions	**Definition:** Ability to perform required tasks
Measured via elapsed timers placed within the code	Measured via functional testing of UI widgets and collected data
	

Particle Control Metrics

Path Traversal Consistency

Definition: Traversal deviation of the magnetic particle over a set path

Calculated from particle location data acquired over multiple path traversals

Simultaneous Command Execution Delay

Definition: Time difference between simultaneous hardware commands

Measured via elapsed timers placed within the code

System Roadmap

1. **GUI**
2. Image Segmentation
3. Hardware Description
4. Particle Translation Model

GUI Design

Objective: In addition to satisfying all established requirements, provide an intuitive user experience.

Major Areas

- UI Layout
- Setup
- Path Drawing
- Operational Control
- Data Collection

Figure 2. UI coded in C++ due to its speed and object-oriented support.

Figure 3. UI built using Qt API (The Qt Company, Espoo, Finland).

GUI Layout

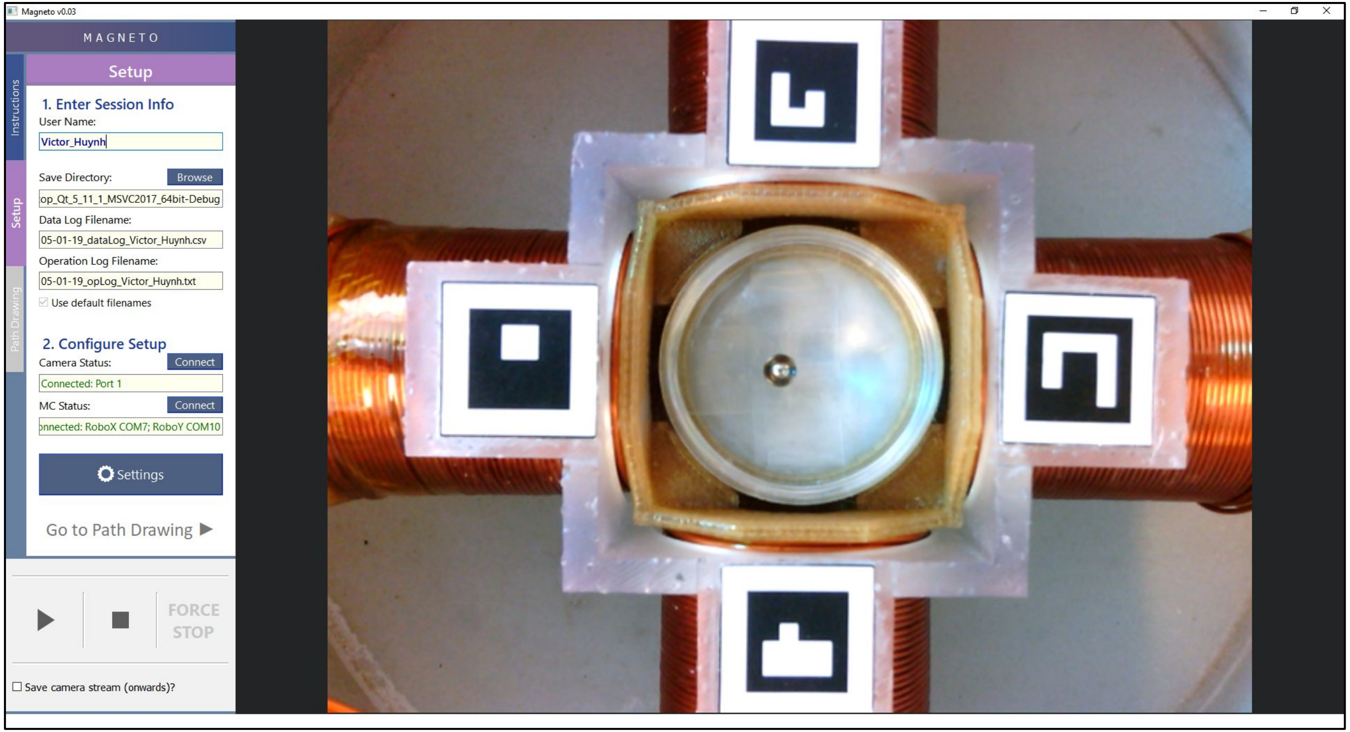


Fig 4. Layout consists 1 full-sized window (shown left) and 1 sub-window. Content is delegated to the left tab-widget. Operation control panel is to the bottom-left.

Setup

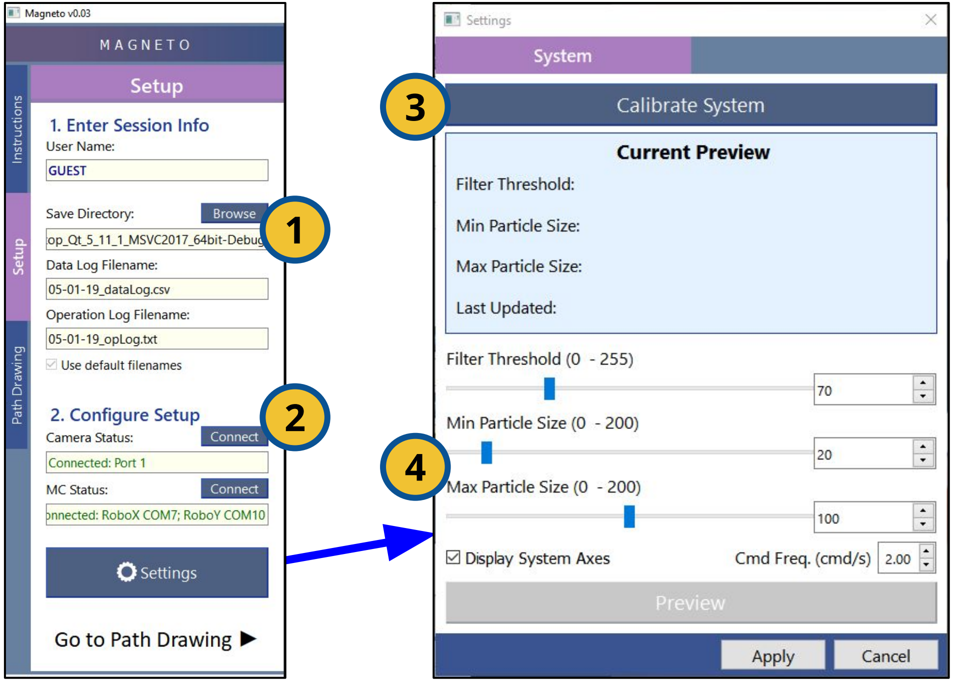


Fig 5. Setup is contained within the Setup tab and Settings subwindow. The procedure readies internal data for path drawing.

Main Setup Procedure

1) Specify directory to export data.

2) Connect GUI to camera and motor controllers.

3) Calibrate system coordinates and field-of-view.

4) Configure particle detection parameters.

Path Drawing

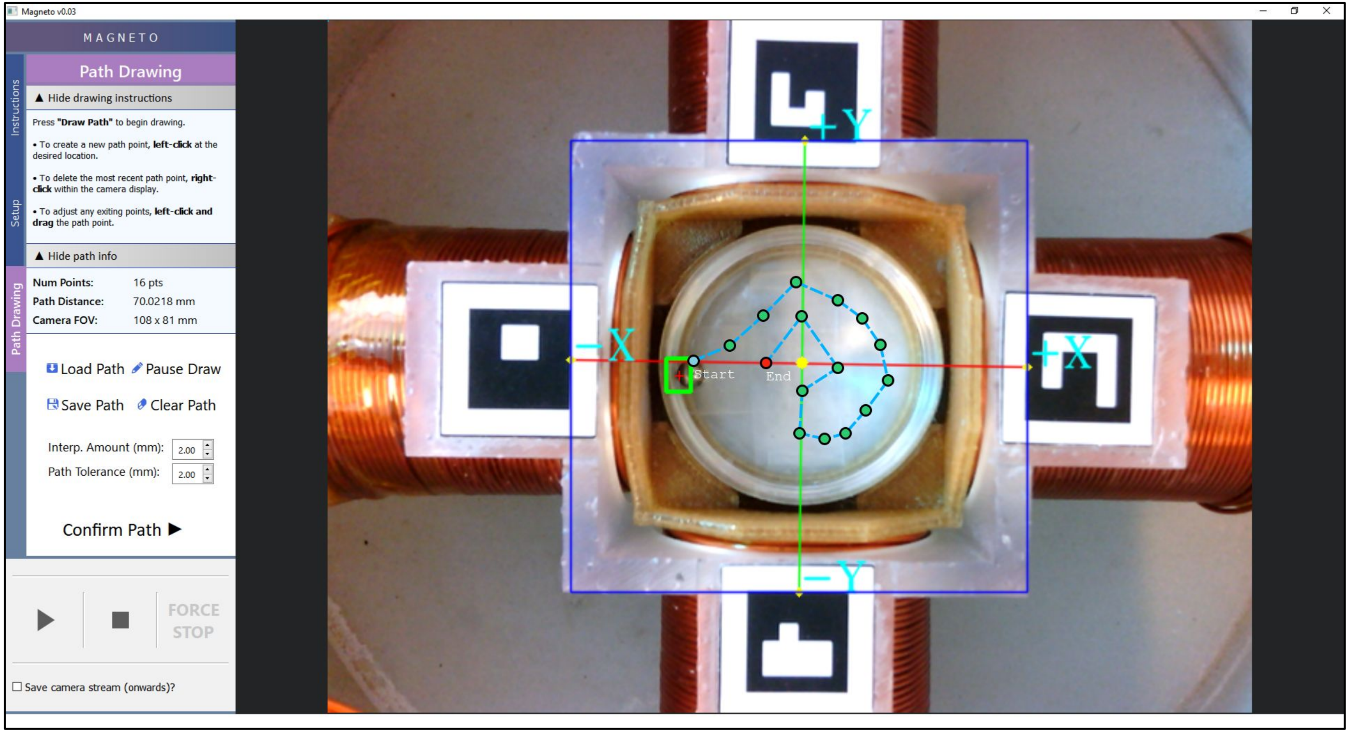
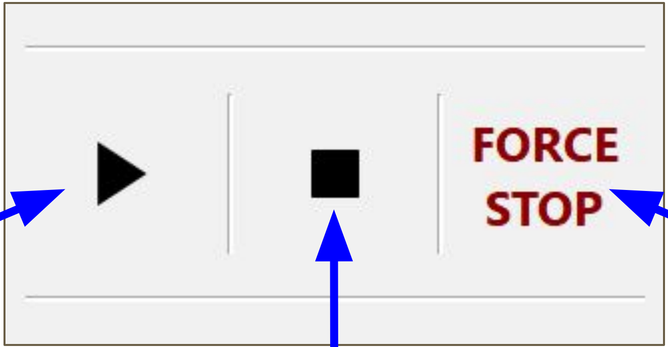


Fig 6. User can load a previously saved path or draw a new one via mouse-clicks. Path markers can be easily manipulated (i.e. drag and drop).

Operation Control

Fig 7. Operation control panel consists of 3 buttons: Start/Pause, Stop, and Force Stop.



Start/Pause

Resumes/pauses delivery. When paused, certain settings are allowed to be modified.

Stop

Stopping delivery ends the operation with proper closeout. This occurs automatically when the particle reaches the path end.

Force Stop

Force stopping delivery immediately ends the operation with improper closeout. Used only for emergencies.

Data Collection

	A	B	C	D	E	F	G
1	Time (s)	Init. X-Pos (mm)	Init. Y-Pos (mm)	Final X-Pos (mm)	Final Y-Pos (mm)	Dist. Moved (mm)	Velocity (mm/s)
2	0.906	-14.2589	-1.75795	-14.2589	-1.75795	0	0
3	1.717	-13.0511	-0.595749	-13.0511	-0.595749	0	0
4	2.527	-12.5335	-0.097664	-12.5335	-0.097664	0	0
5	3.343	-12.0192	0.231138	-12.0192	0.231138	0	0
6	4.158	-11.5015	0.729223	-11.5015	0.729223	0	0

Traversal Data Fields
(e.g. particle location, distance moved)

Hardware Data Fields
(e.g. current scale, current duration)

H	I	J	K	L	M	N	O	P
Plus X Current Scale	Plus X Current Duration (ms)	Minus X Current Scale	Minus X Current Duration (ms)	Plus Y Current Scale	Plus Y Current Duration (ms)	Minus Y Current Scale	Minus Y Current Duration (ms)	Command #
77	100	0	0	35	100	0	0	1
77	100	0	0	35	100	0	0	2
77	100	0	0	35	100	0	0	3
77	100	0	0	35	100	0	0	4
77	100	0	0	35	100	0	0	5

GUI Speed Testing Results

Lag Time Measurements of Significant Functions		
Function	**Avg Lag Time (ms)**	**Standard Deviation (ms)**
Software Startup	1287.1280 ms	16.5940 ms
Connect Camera Connection	987.3840 ms	182.7573 ms
Connect Motor Controllers	10.8091 ms	0.5863 ms
Open Settings Subwindow	52.3760 ms	2.1065 ms
Start/Pause Delivery	1.9845 ms	0.5634 ms
Stop Further Motor Controller Execution	0.0932 ms	0.0808 ms
Simultaneous Execution Delay	0.5166 ms	0.5940 ms

Table 1. Lag times for significant GUI functions. Most intensive processing took ~1300 ms, while majority fell below 100 ms. Activation of multiple motor controller commands occurs within 1 ms of each other.

System Roadmap

1. GUI
2. **Image Segmentation**
3. Hardware Description
4. Particle Translation Model

Image Segmentation Design

Objective: Detect particle and coil locations within streamed optical images consistently with minimal delay.

1. **Particle Localization Alternatives:**
    - Image Subtraction
    - Template Matching
    - Hough Circles

2. **Coil Localization Alternatives:**
    - Color Thresholding
    - ArUco Markers

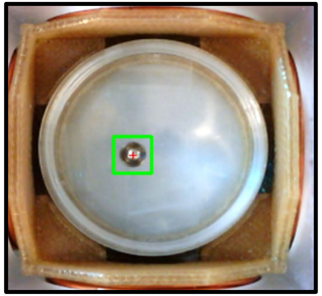


Figure 8. Particle localization display in petri-dish setup.

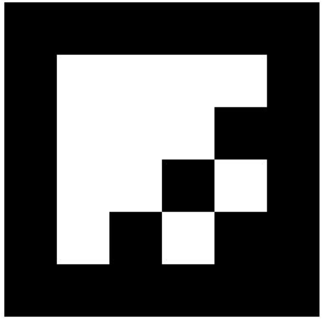


Figure 9. ArUco fiducial marker example.

Image Segmentation Final Design

1. **Particle Localization**
   - Synthesize a background image
   - Subtract from all subsequent images
   - Filter for particle location

2. **Coil Localization**
   - Affix fiducial marker structure
   - Segment images for binary marker pattern
   - Initialize coordinate system based on coil marker locations

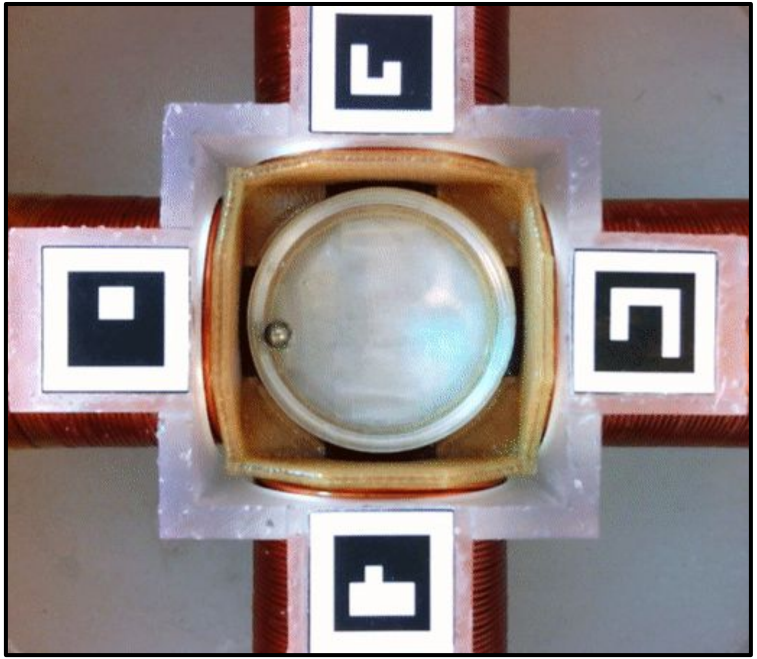
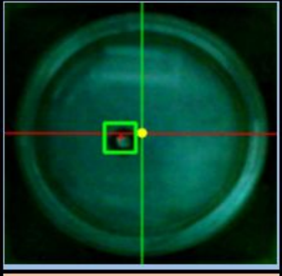
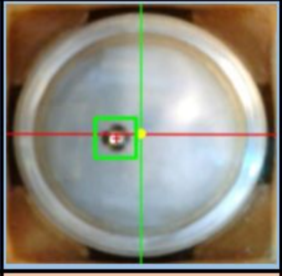
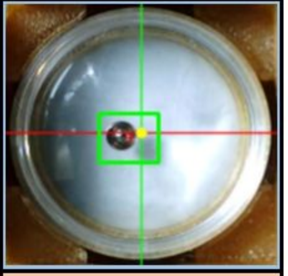


Figure 10. Example coil and particle localization calibration procedure.

Image Segmentation Testing Results

Table 2. Detected coil and particle localization deviation over different lighting conditions. Particle was detected with low particle deviation in all but one lighting condition (6 Lux). Coil markers were detected with exceptionally low deviation.

	Lighting Condition		
	6 Lux (Dark Room)	80 Lux (Ambient Light)	220 Lux (Direct Light)
Detected Feature			
Particle Location Deviation (mm)	0.3432	0.0521	0.1337
Coil Marker Deviation (mm)	0.0049	0	0.0266

System Roadmap

1. GUI

2. Image Segmentation

3. **Hardware Description**

4. Particle Translation Model

Hardware Description

1. GUI sends instructions simultaneously to both motor controllers

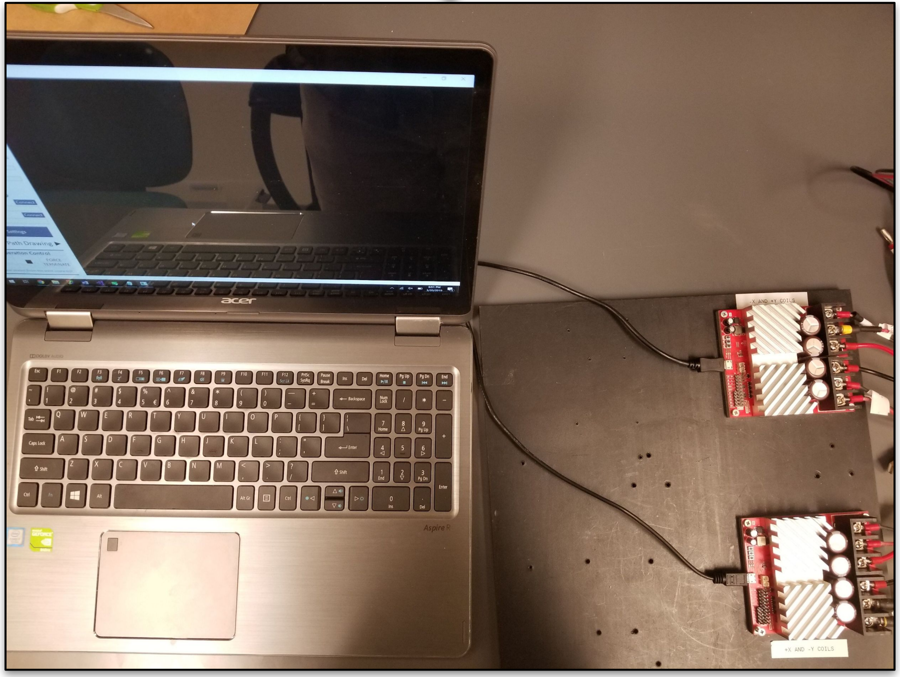


Fig 19. GUI (left) and motor (right) controllers connected via USB

Hardware Description

2. Motor controllers run current through connected solenoid coils

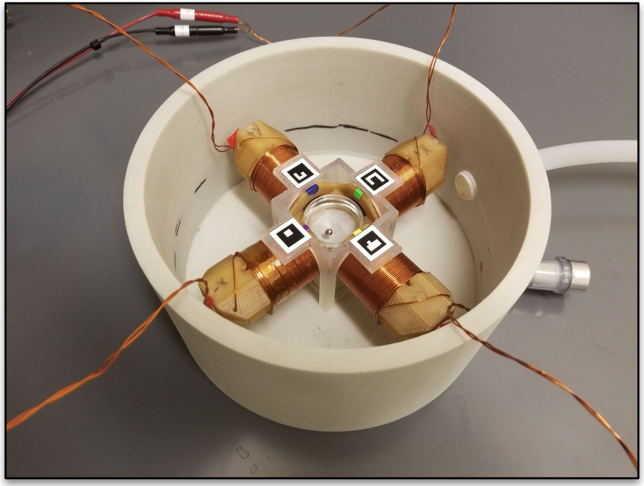


Fig 20. Four coil array with solenoid connections placed within fluid housing

3. Induced magnetic fields translate permanent magnets

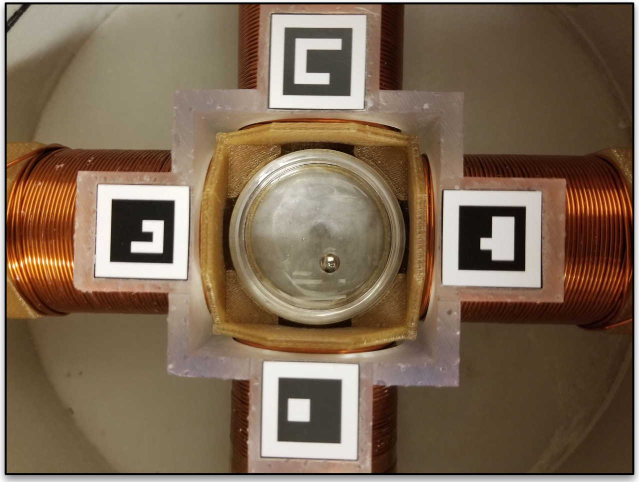


Fig 21. Neodymium sphere magnet suspended in a solution within a petri-dish

System Roadmap

1. GUI
2. Image Segmentation
3. Hardware Description
4. **Particle Translation Model**

Translation Model Design

Objective: Establish a methodology for accurately translating the magnetic particle through a user-defined path using a solenoid coil array.

Options:

- Surface Mapping
- Regression Neural Network

Figure 11. Particle translation concept

Data Collection

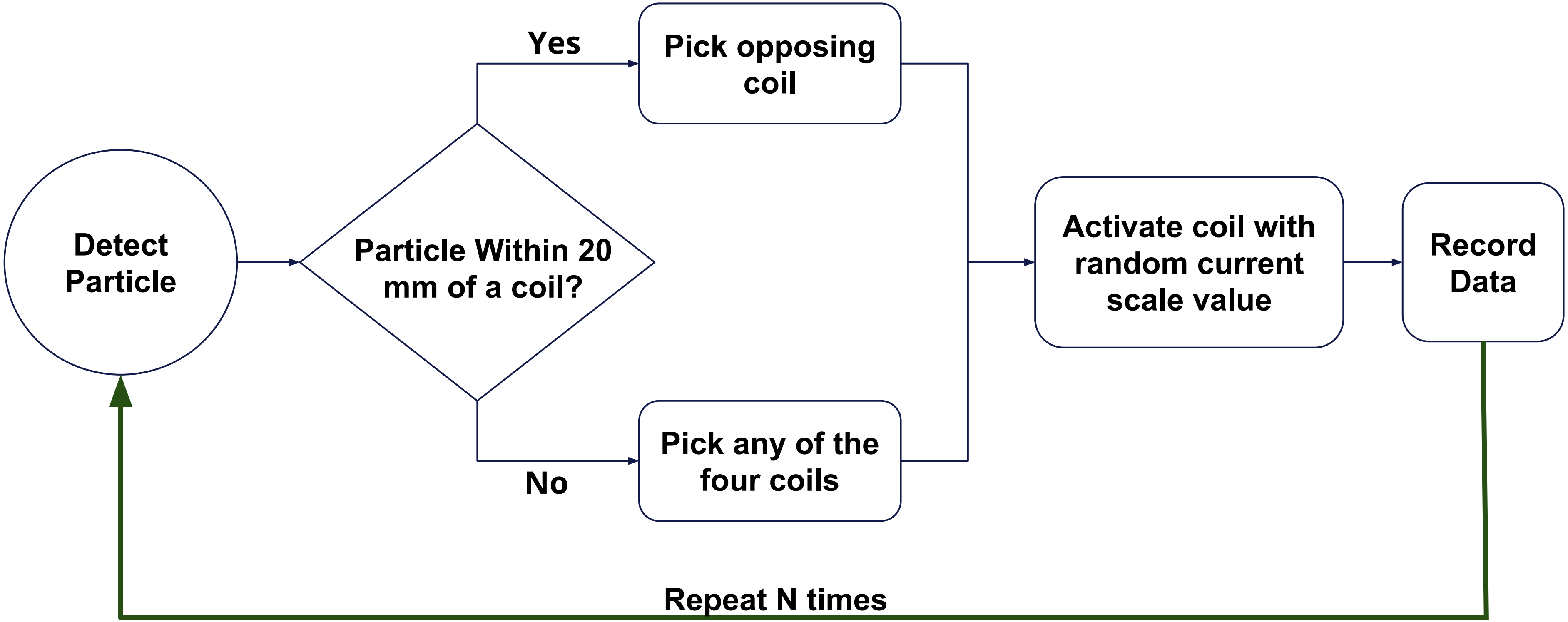


Figure 12. Automated data collection procedure

Surface Fitting - Architecture

Inputs: Desired travel distance (mm), Distance from coil (mm)

Output: Necessary current scale (0 - 127 unitless)

Used 3D surface to map current scale as a function of:

- Desired travel distance (mm)
- Distance from coil (mm)

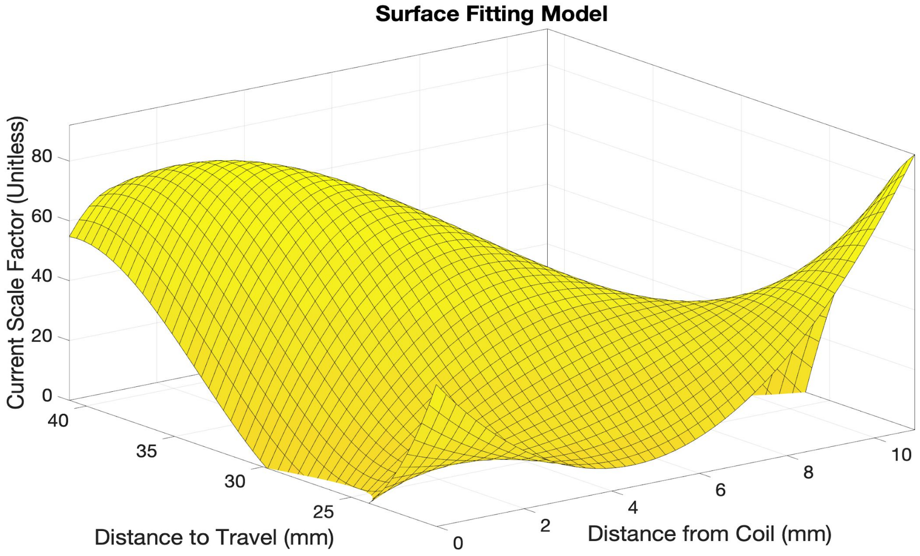


Fig 13. Surface fitting tested architecture.

Neural Network - Preprocessing

- Dataset of size 11,000 containing particle translation data, 80% used for training and 20% testing

- Normalize all values using Z-score

$$z = \frac{x - \mu}{\sigma}$$

Neural Network - Architecture

	Mutual Information Regression	Empirical Evaluation
Determined Features (mm scale)	- Initial Axis Coord (X or Y) - Final Axis Coord (X or Y) - Distance to Axis Coil - Distance from Opposing Axis Coil	- Initial X Coord - Initial Y Coord - Final X Coord - Final Y Coord - Distance to Axis Coil

Table 3. Relevant features determined by mutual information regression (left) and empirical evaluation (right). Two network architectures, one with four inputs and the other with five, were tested using these features.

Neural Network Chosen Architecture

Inputs: Initial coordinates (mm), Desired coordinates (mm), distance from coil (mm)

Output: Necessary current scale (0 - 127 unitless)

1 hidden layer, 10 hidden neurons

4 total networks were trained (one per solenoid coil)

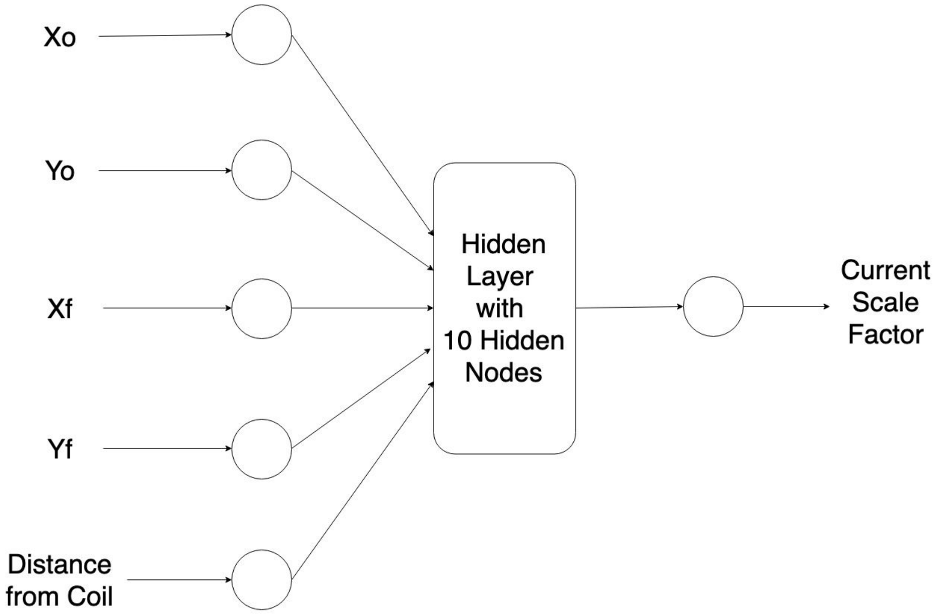


Figure 15. Diagram of chosen neural network (5-Input) architecture.

Path Comparison (5-Input vs. 4-Input NN)

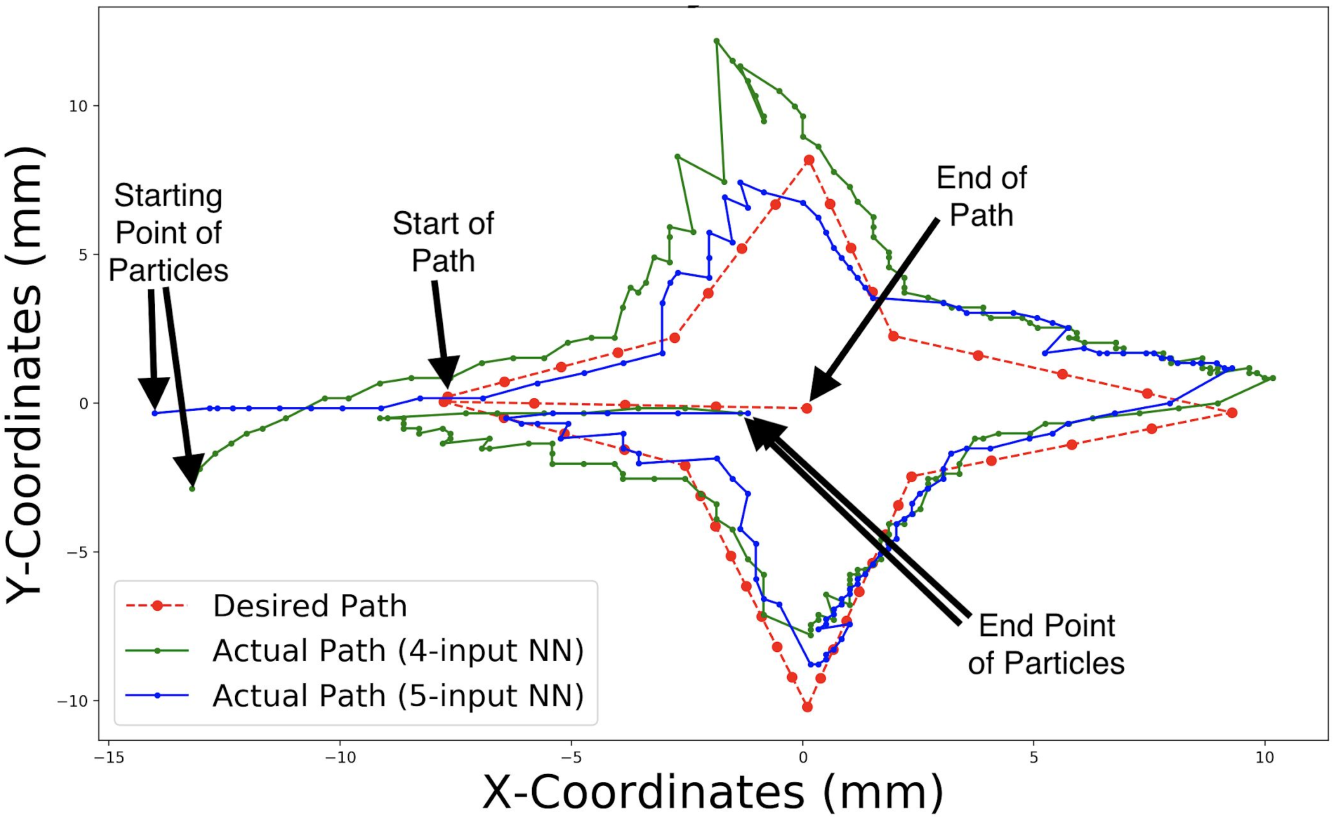


Figure 14. Path traversal comparison between 4-Input (green) and 5-Input neural network architecture (blue) across the same desired path (red).

Path Comparison (NN vs. SF)

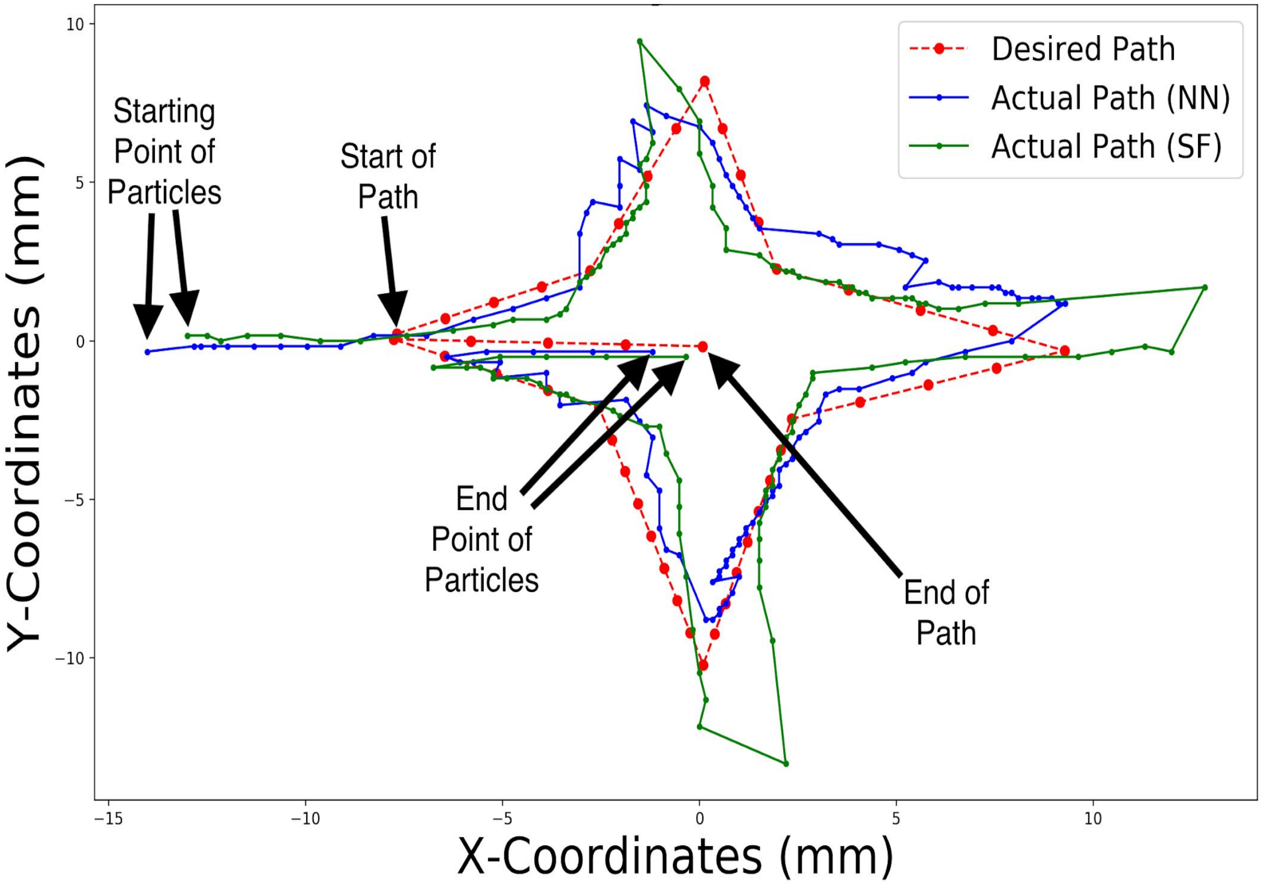


Figure 17. Path traversal comparison between 5-Input neural network (blue) and surface fitting (orange) across the same desired path (black).

Average Traversal Path (NN)

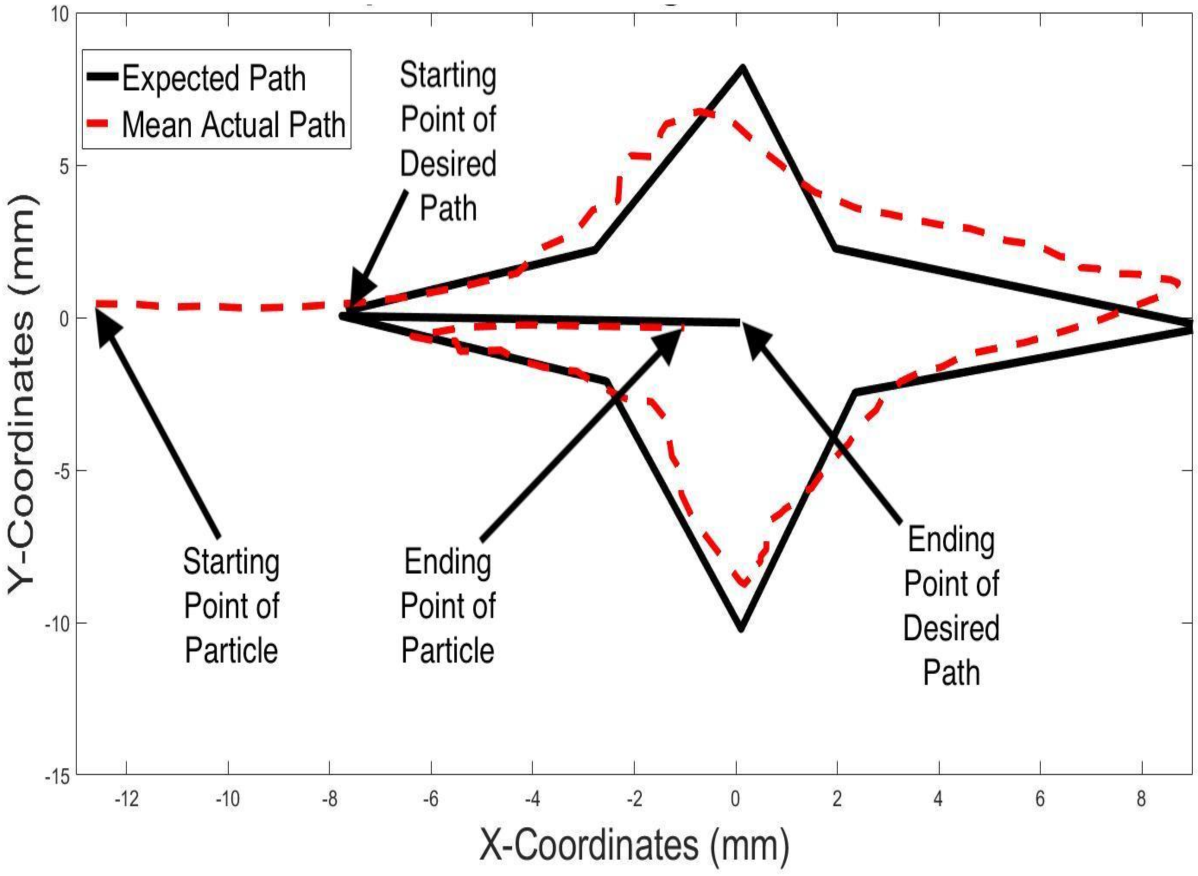
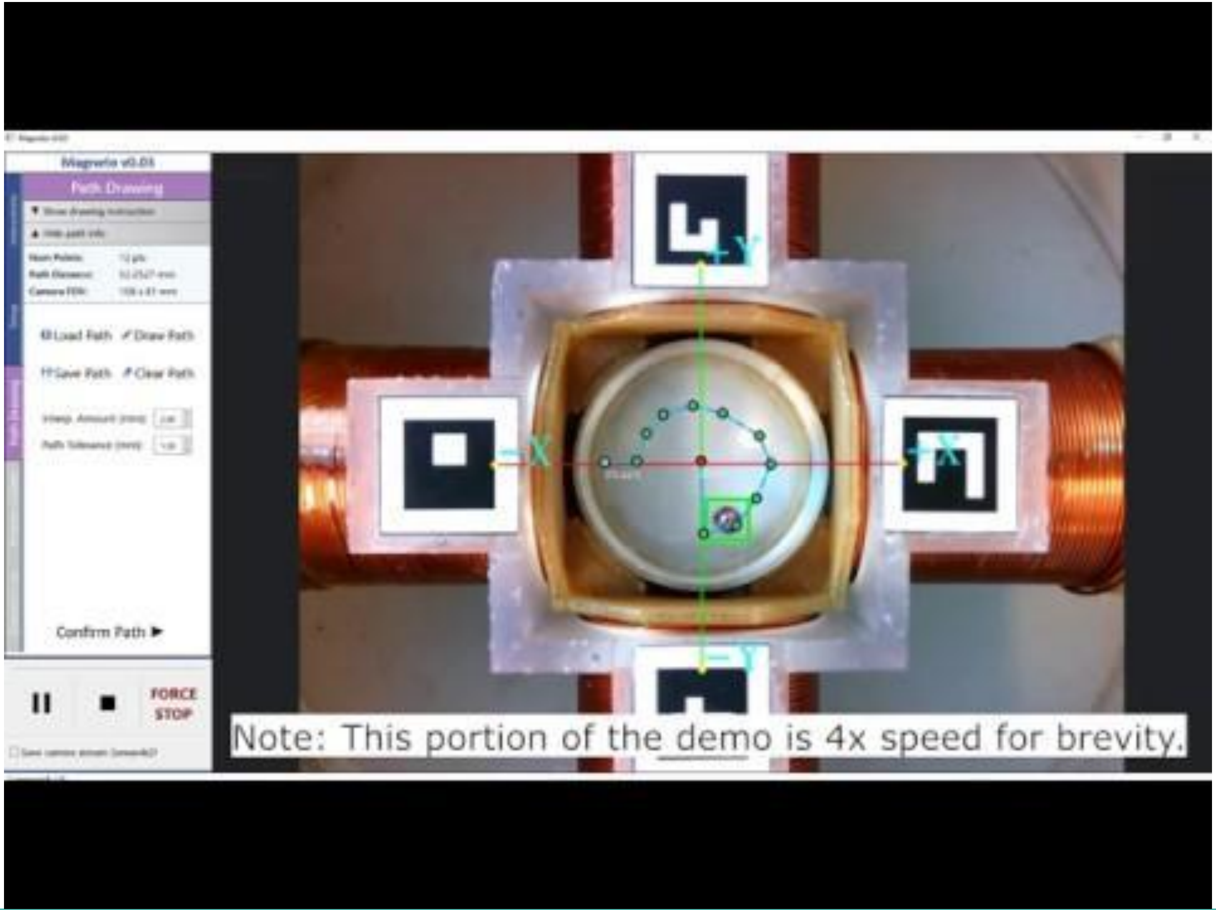


Figure 18. Average neural network path traversal for N = 10 trials. Standard path deviation was calculated to be 0.85 mm.

Particle Delivery Demo



Budget

Item	Quantity	Cost
Qt Development Environment	4	$0
Microsoft Visual Studio 2017 Community	4	$0
OpenCV	4	$0
Github Version Control	4	$0
27V 1.593kW Power Supply *	1	$360
ArUco Markers	4	$0.50
Roboclaw 2x60A Motor Controllers *	2	$400
Ipevo Ziggi USB Camera *	1	$148
4-Coil Magnetic Solenoid Array *	1	N/A
Micro USB to USB 2.0	2	$10.78
Total Operational Cost		**$919.28**

*** Supplied by Weinberg Medical Physics, LLC**

Total Hours (Fall 2018 - Spring 2019): 2050.25 hrs

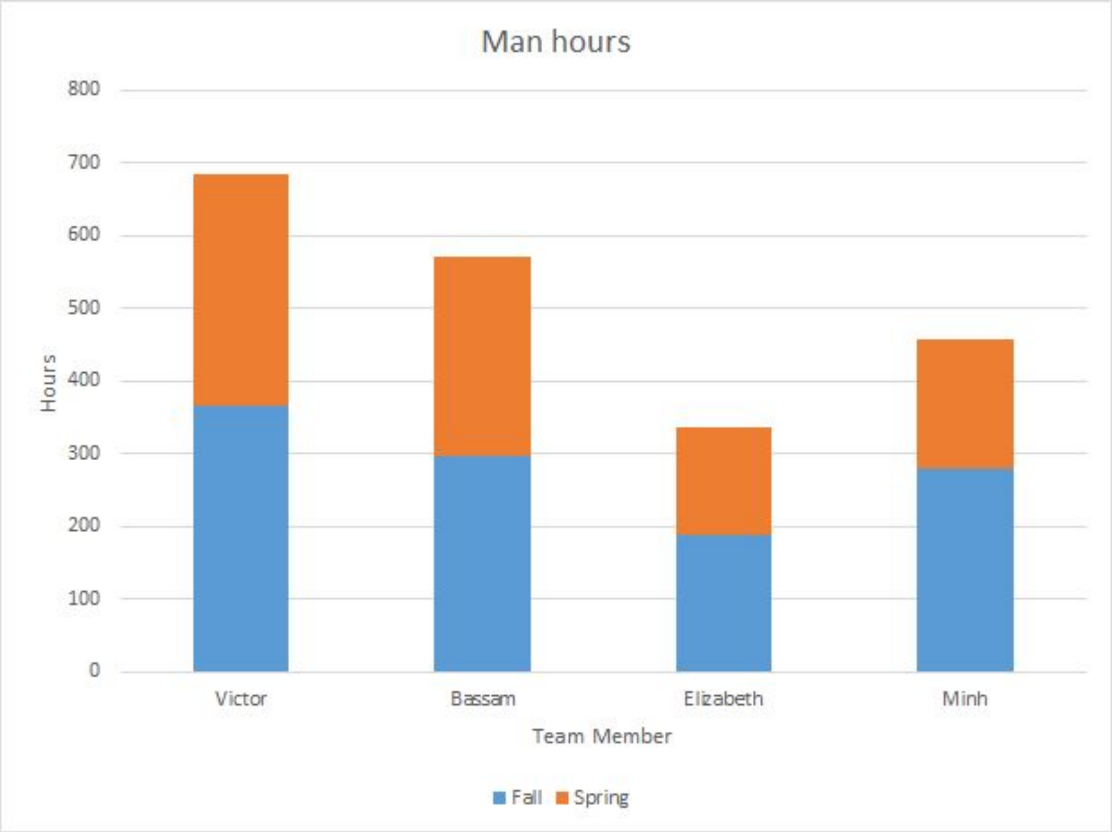


Fig 22. Hours contributed by each team member over two semesters

Conclusion

Outcome:

Project MAGNETO can successfully track and guide a magnetic object along a user-drawn path.

Limitations:

- Translation model (accuracy, “dead zone” regions, data collection)
- Current iteration (ball magnet, small traversal region, detection glitches)

Future Work:

- Test neural network on larger platforms and different magnetic objects
- Port software features to sponsor’s 3D MCS

References

[1] B. Shapiro, S. Kulkarni, A. Nacev, S. Muro, P. Y. Stepanov, and I. N. Weinberg, “Open Challenges in Magnetic Drug Targeting,” Wiley Interdiscip Rev Nanomed Nanobiotechnol, vol. 7, no. 3, pp. 446–457, May 2015.