Neural Network-based Magnetometer Sensor Model for Virtual Magnetic-based Navigation Testbed

January 6, 2024

Aditya Penumarti, Nikhil Iyer, Loi Park, Humberto Ramos, Jane Shin

University of Florida

APRILab
ACTIVE PERCEPTION AND ROBOT INTELLIGENCE LAB

SCITECH FORUM

UF UNIVERSITY of FLORIDA

What is Magnetic Navigation?

- Magnetic Navigation (**MagNav**) uses magnetic fields to navigate in **GPS-Denied** environments.
- Accomplished through the use of a Total Field Magnetometer (**TFM**).
- Often uses a particle filtering method to localize.
- Commonly performed with a pre-collected map.
  - It can be conducted without one but is difficult.

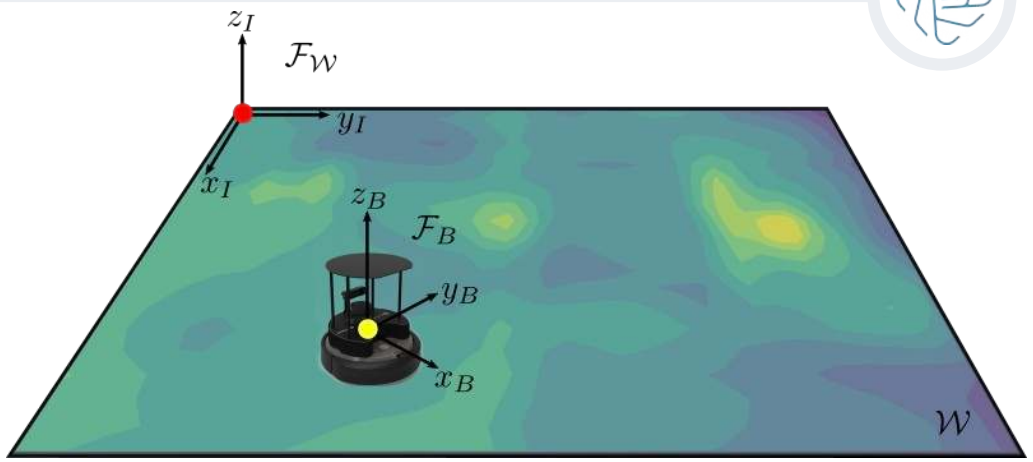


Figure 1. Navigation in Magnetic Anomalies



Figure 2. Total Field Magnetometer [1]

[1] QUSpin, “QTFM Gen-2”, https://quspin.com/qtfm-gen-2/

Problem

- MagNav has:
  - **High upfront costs**
  - **Complex calibration process**
- Distortions are **modeled linearly** [2]
- No accurate sensor models
- Naïve simulations do not represent distortions accurately
- Most readily available simulation environments cannot model a vector magnetometer.

Can we accurately model the distortions affecting a total field magnetometer without a vector magnetometer?

[2] Ramos, H. et al, "Ownship Calibration for Indoor Magnetic Navigation", AIAA SciTech 2024

Problem Formulation

- Goal is to emulate a measurement that is disturbed by:
  - **Environmental electromagnetic sources**
  - The **robot's self-structure** that include permanent, induced and Eddy effects.
- An effective dual of a **Tolles-Lawson calibration** is predicted using an **artificial neural network**
- Predicts **distortions** for a physical and virtual robot.

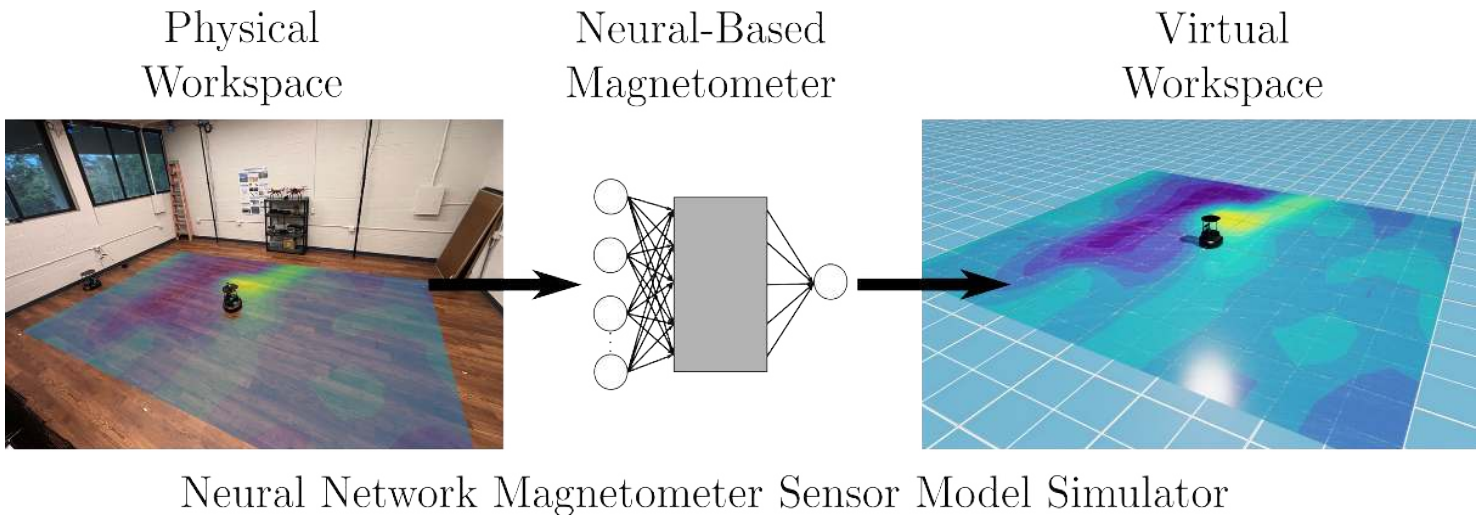


Figure 3. Neural Magnetometer Pipeline

Neural Network Architecture

- Readily available sensor data was used from the **physical** or **virtual** platform
- Highly correlated variables were included with some lesser included
- This forms **20 neuron input layer**

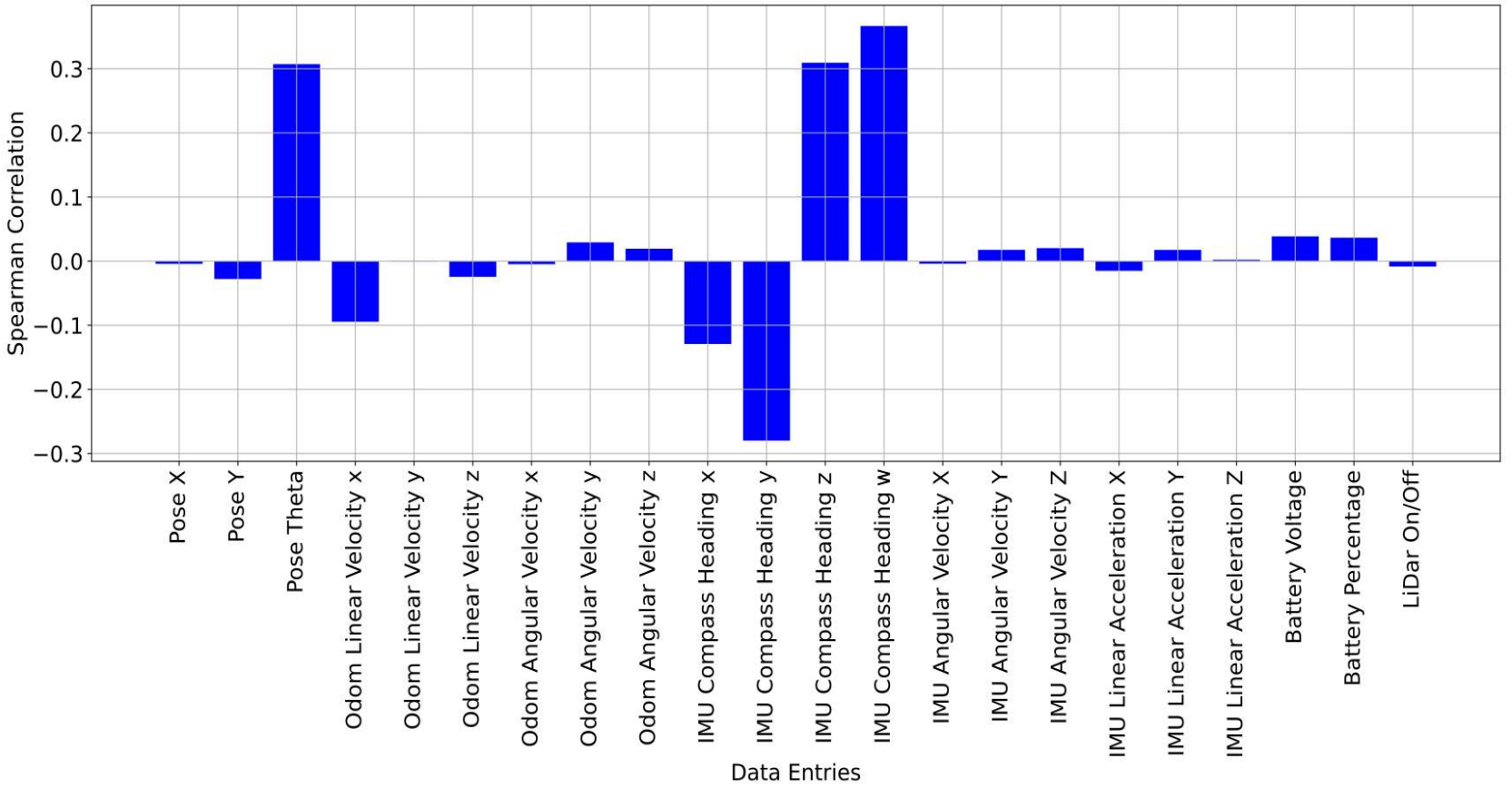


Figure 4. Spearman Correlation of Data

Neural Network Architecture

- **6-layer feed-forward** neural network
  - 20-neuron input layer
  - 4 512-neuron hidden layers
  - 1-neuron output layer
- Activation function was **ReLU**
  - 30% dropout to prevent overfitting.
- Output was the **magnetic disturbance**
  - Allows for a position invariant predicition

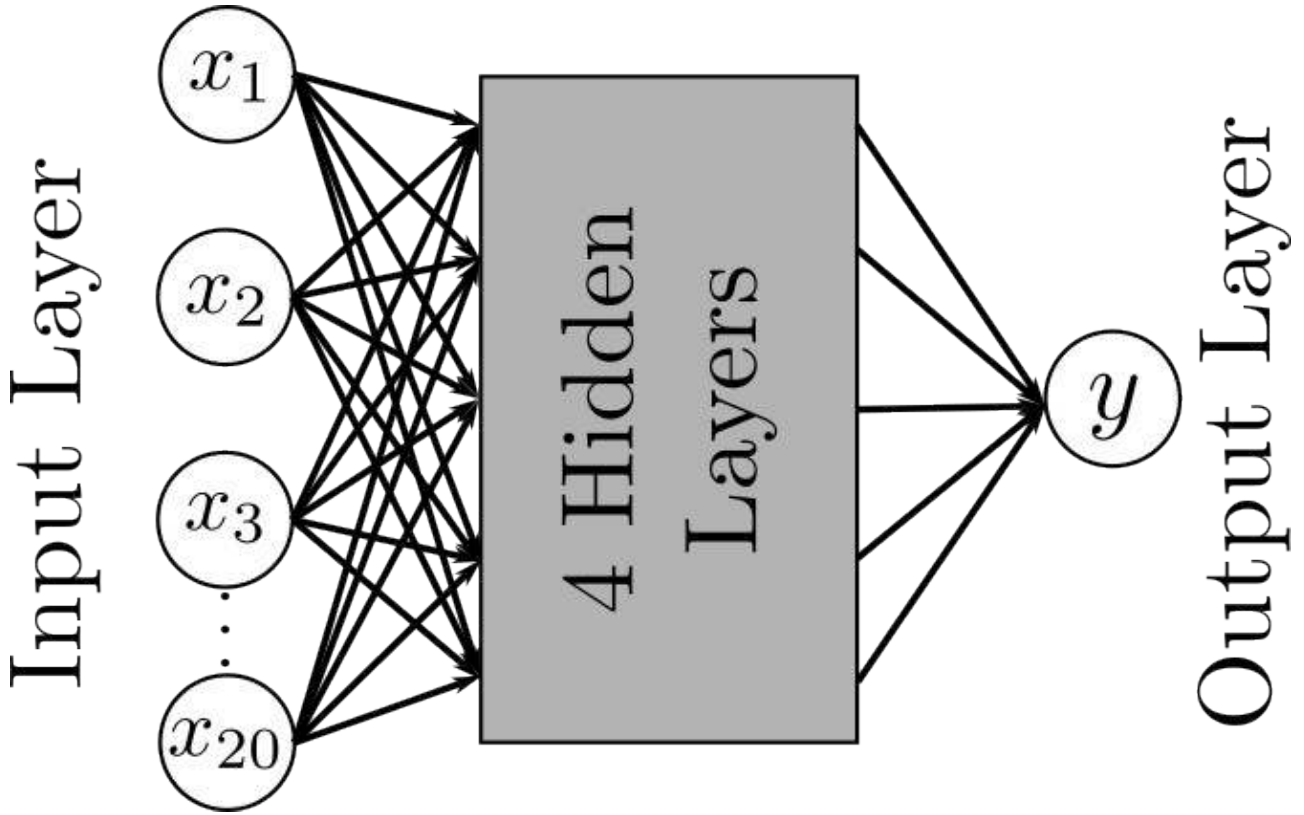


Figure 5. Neural Network Model

Data Collection For Training

- An **indoor UGV** with a TFM was used to collect data.

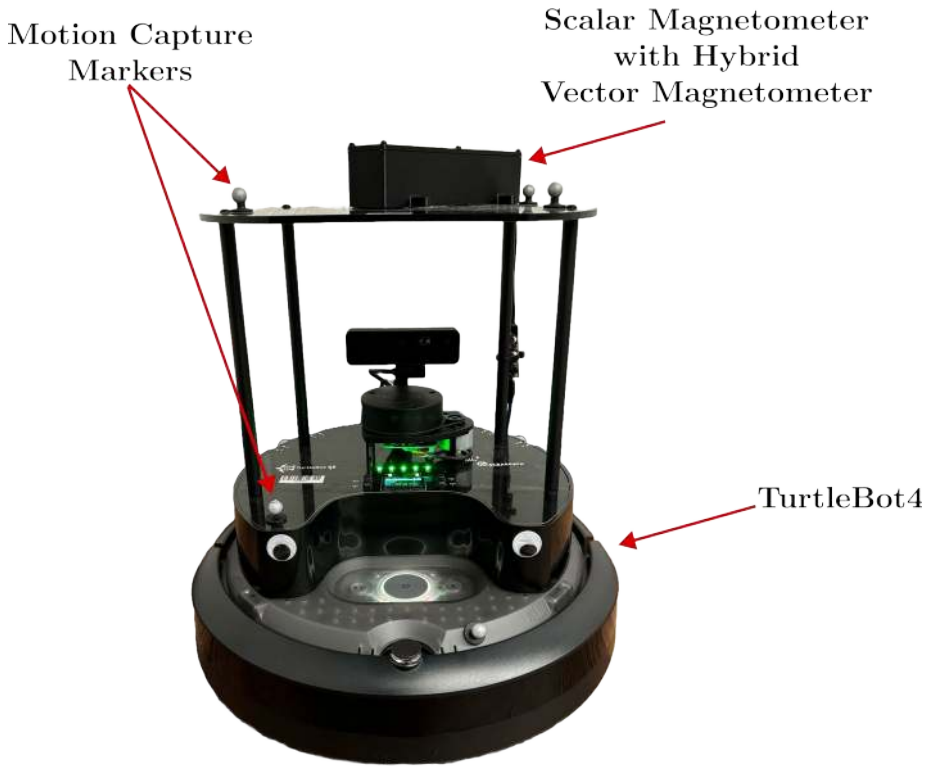


Figure 6. TurtleBot with Total Field Magnetometer

Data Collection For Training

- An **indoor UGV** with a TFM was used to collect data.
- Several paths were chosen to collect the data

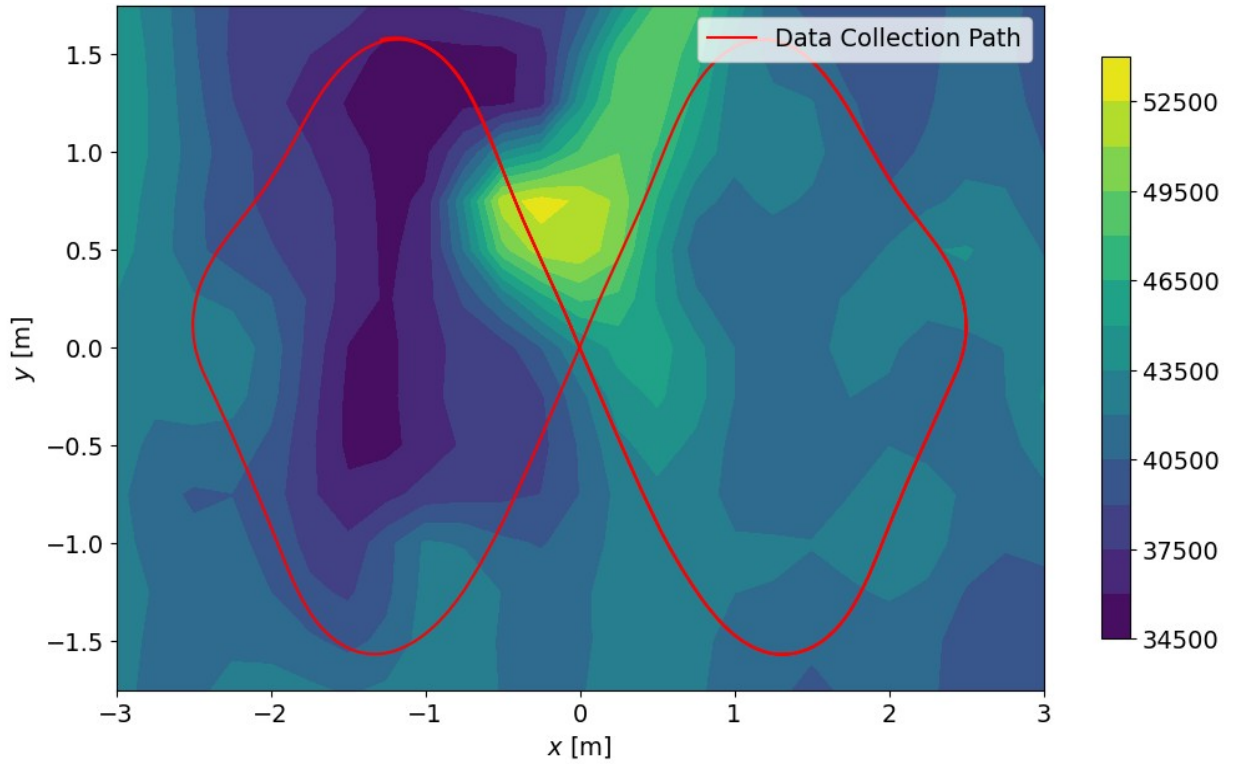


Figure 7. Sine Wave Path For Data Collection

Data Collection For Training

- An **indoor UGV** with a TFM was used to collect data.
- Several paths were chosen to collect the data
- **Unstructured paths** were included for a broader range of the input space

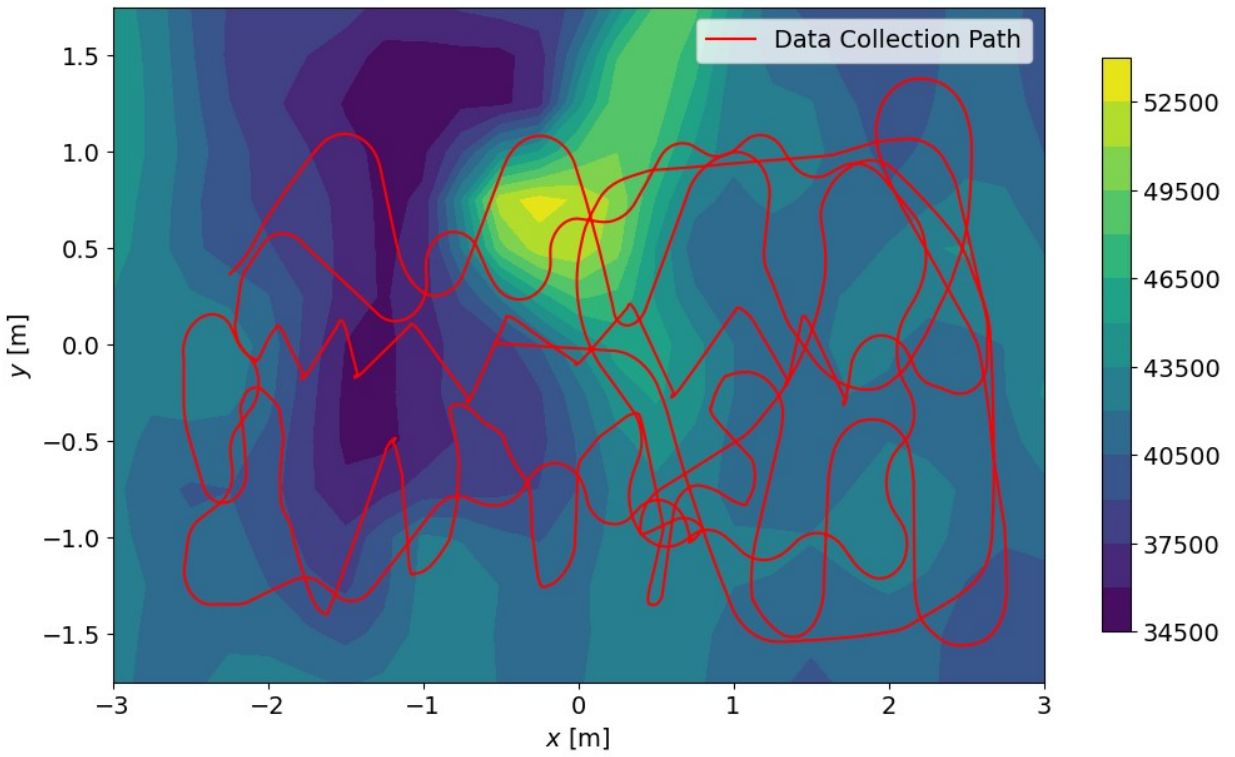


Figure 8. Unstructured Trajectories for Data Collection

Data Collection For Training

- An **indoor UGV** with a TFM was used to collect data.
- Several paths were chosen to collect the data
- **Unstructured paths** were included for a broader range of the input space
- The neural network was trained on several iterations of these paths for **50 epochs**

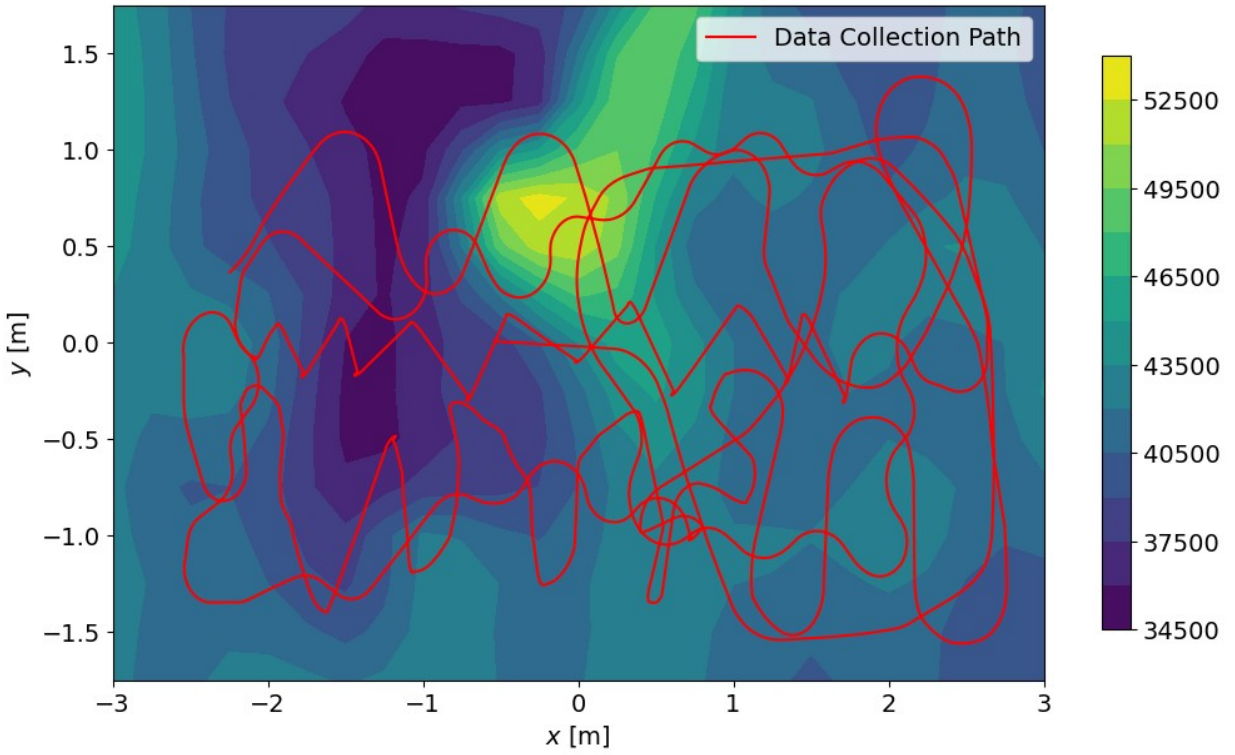


Figure 8. Unstructured Trajectories for Data Collection

Isaac Sim Setup

- **NVIDIA's Isaac Sim** was used for the digital twin implementation
- The physical system was duplicated into the virtual world for the **Neural Magnetometer**
- System was connected to **ROS 2** to enable easy integration in physical and digital worlds
- Isaac Sim allows for the expansion into reinforcement learning applications through **Isaac Gym**

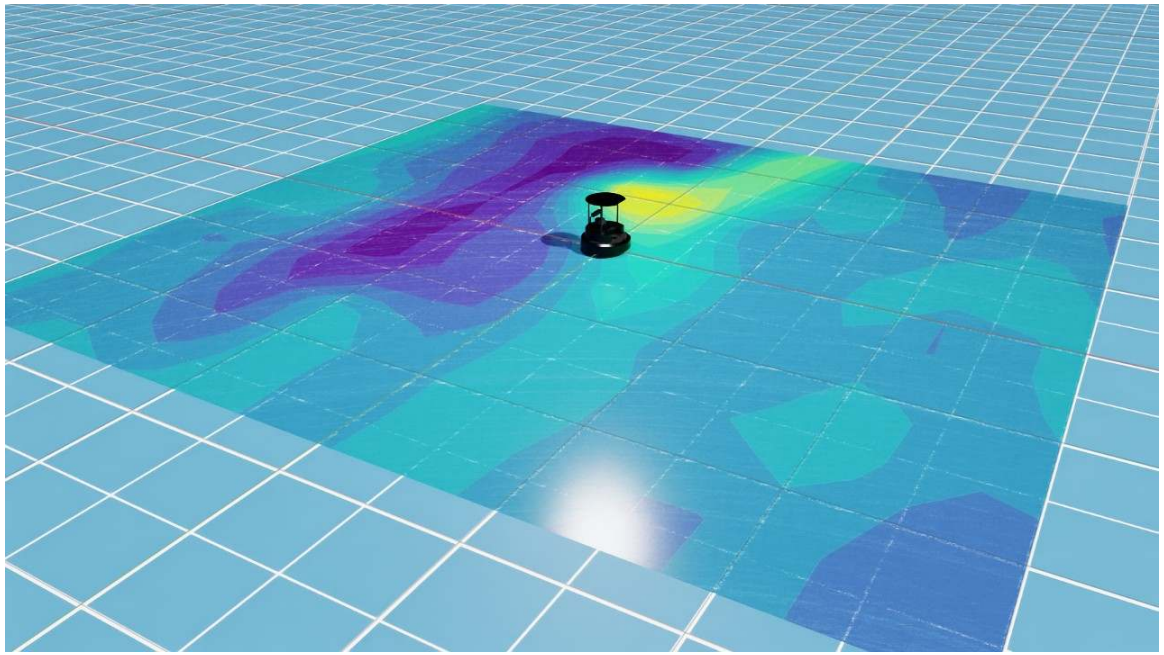


Figure 9. Digital Twin Environment in Isaac Sim

Results

- Model was evaluated on several test scenarios

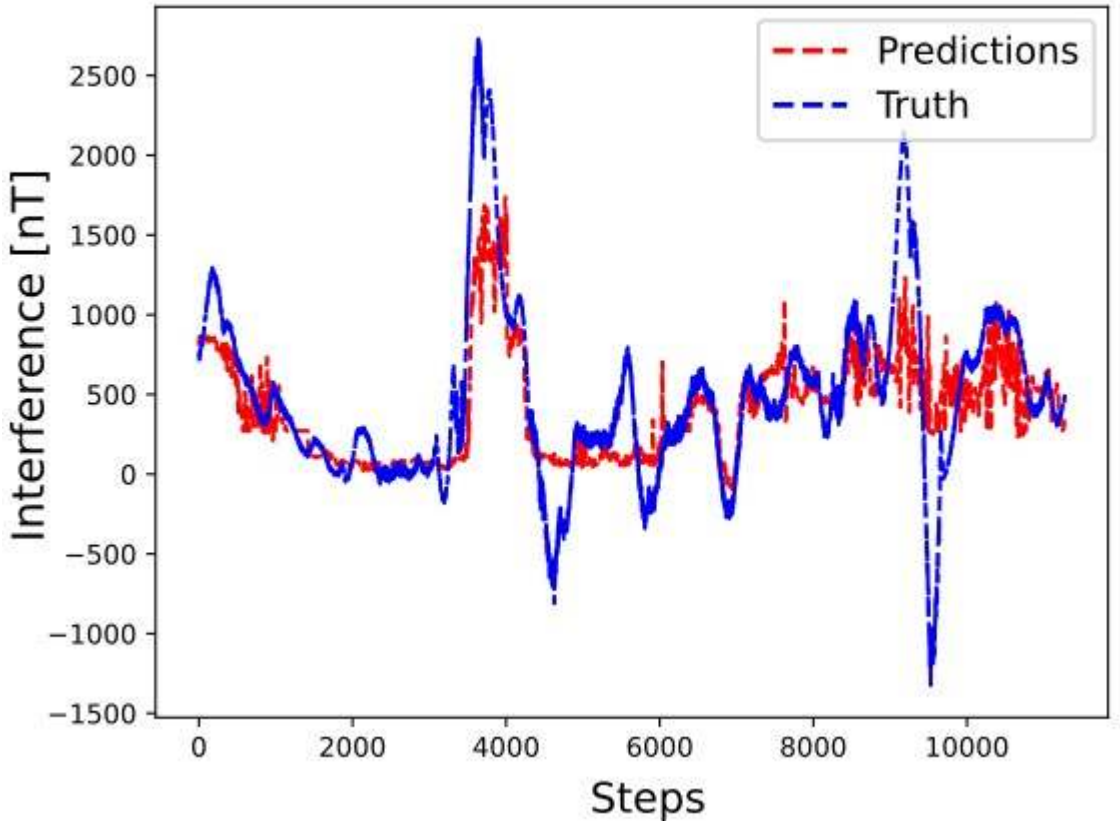


Figure 10. Model Evaluation

Results

- Model was evaluated on several test scenarios
- **MSE** compared against a **naïve mean** of the training data
  - Loss shows better performance

Test Case	Path Types	Loss	Naive Loss
A	High Frequency Sine Wave	217.6165	225.1818
B	Curvilinear Run #1	307.8796	407.1321
C	Curvilinear Run #2	267.0007	381.4163
D	Lawmower with a Y-axis Split	284.6316	333.0688
E	Narrow Figure 8	323.6039	409.3078

Figure 11. Comparing Neural Network to Naïve mean

Results & Conclusion

- Model was evaluated on several test scenarios
- **MSE** compared against a **naïve mean** of the training data
  - Loss shows better performance
- The digital and physical experiments show that the network performs similarly

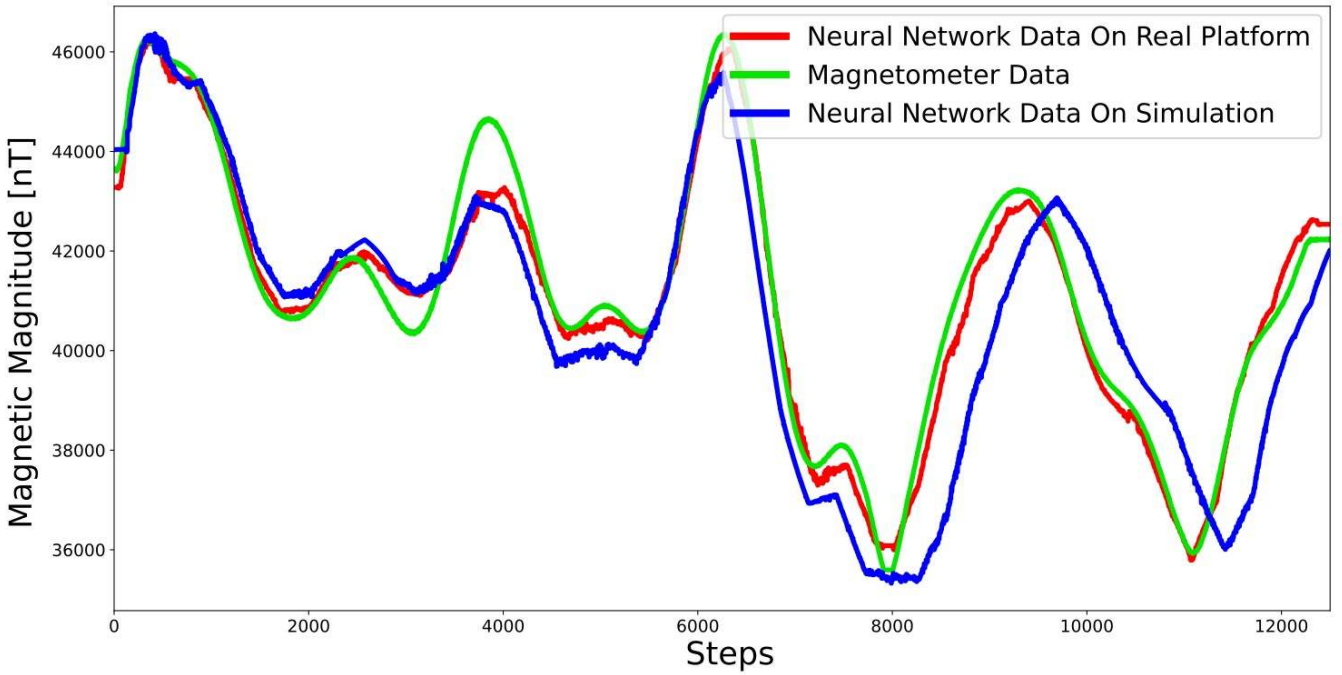
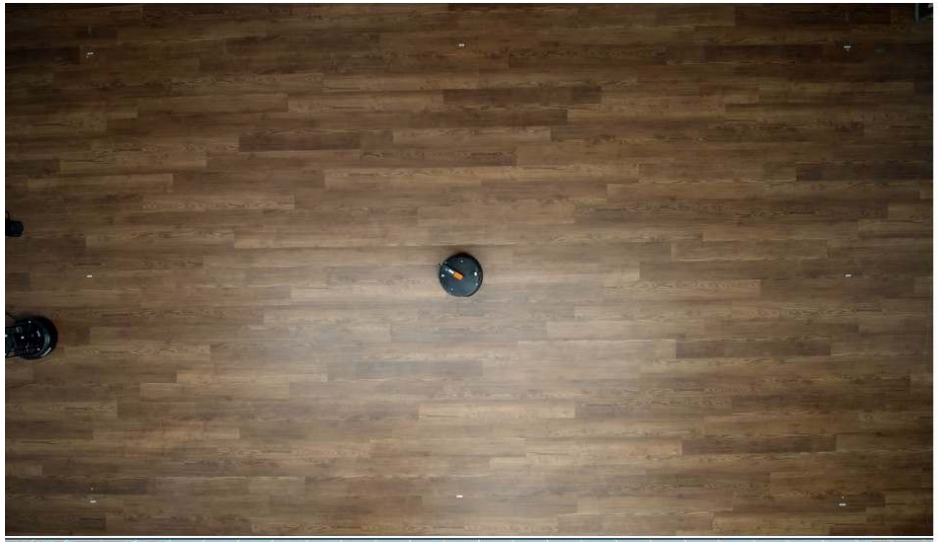
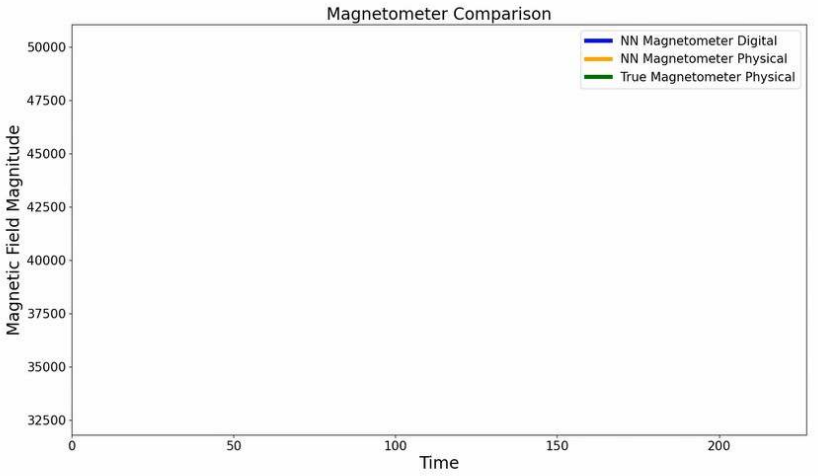


Figure 12. Digital and Physical Experiments

Demo




Magnetometer Comparison
NN Magnetometer Digital
NN Magnetometer Physical
True Magnetometer Physical
Magnetic Field Magnitude
50000
47500
45000
42500
40000
37500
35000
32500
0
50
100
150
200
Time

Future Work

- Model can be improved by simulating a **vector magnetometer**
  - Based on highly correlated information from the true heading
  - This is inline with the Tolles-Lawson Calibration
- Ability to add **external interference** sources in the simulation
  - Allows for the changing of the map
  - Gives a complete implementation

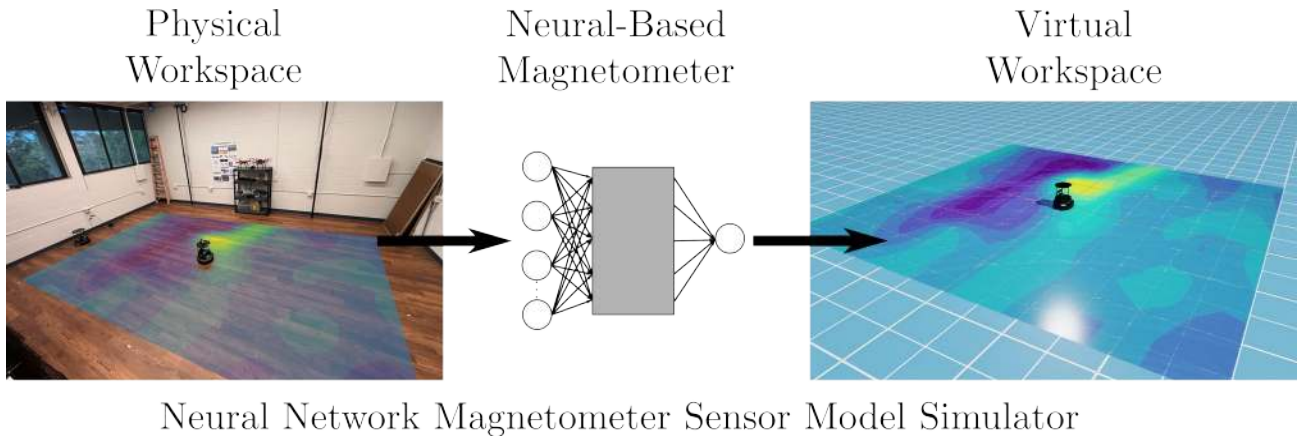


Figure 3. Neural Magnetometer Pipeline

Thank You!

Questions?