

---

# Fast and Long Range Magnet Tracking in Free Space

---

**Sumukh Marathe**

**Mohit Yogesh Padhye**

**Devika Deshpande**

**Aditya Deshmukh**

**Sadaf Inamdar**

{sumukhm, mohityp, devikasd, aditydes, sadafin}@umich.edu

## Abstract

Magnet tracking in 3D free space has found multiple applications in the field of augmented and virtual reality, and biomedical sensing. State of the art physics-based analytical systems for estimating the location and pose of the magnet typically use an array of magnetometers to measure magnetic fields of permanent magnets or electro-magnets. However, such analytical sensing systems are susceptible to the low signal to noise ratios at long distances, thus restricting their range to about 10-15 cm. On the other hand, machine learning and deep learning based approaches which try to extend this range face issues of high latency making it infeasible for real-time tracking. Hence, we believe that developing solutions which increase the range of tracking while maintaining low latency can lead to opening the door to multiple novel applications that cannot be served with current techniques.

To this end, we use a real-world system for magnet tracking. The system comprises of 112 magnetometers mounted underneath a desk of area 160cm \* 85cm. We then develop both, the physics based analytical model, and data-centric model for magnet tracking. These models are then evaluated with the real-world magnet tracking system. Finally, based on the insights from evaluation of the two methods, we propose a novel and hybrid approach which combines the low-latency and accuracy of physics based analytical models and the efficacy of data-centric approaches over longer ranges and areas. Our hybrid model achieves a mean tracking error of 1.81cm, 2.39cm and 2.36cm over  $X$ ,  $Y$ , and  $Z$  axes respectively.

## 1 Introduction

Object tracking in 3D free space is a topic that has been explored extensively in the past two decades. Most of the common state-of-the-art approaches use optical computer vision based approaches. Such approaches are either reliant on adding a source of light or a fiducial marker that is tracked by a camera [16] [17]. Both methods suffer from line-of-sight issues and hence are not applicable in many key areas such as biomedical sensing.

Magnetic fields permeate through most commonly occurring materials like wood, concrete, water and most importantly, human bodies. Thus systems which utilise magnetic fields to track objects overcome the significant issue of occlusion that is present in optical tracking systems. Magnetic tracking systems are wireless, low-power and also low-cost. Hence, they have found applications in multiple domains, some of which include, biomedical sensing [22] [21] and Augmented Reality/Virtual Reality [13],[15]. However, the greatest barrier in deploying such systems in real-world scenarios is the lack of solutions which can track magnets in real-time at increased ranges ( 50 cm). In this work, we first develop baseline approaches based on previously proposed magnet tracking systems. Particularly, we analyse physics based analytical approaches and data-centric approaches for magnet tracking at long distances. We then propose a novel hybrid approach that combines the strengths of the two baseline techniques and removes some critical faults. We believe that such an approach would enable magnet tracking at long distances while maintaining real-time tracking.

We believe that magnet tracking solution that can reliably, accurately and efficiently track a magnet at increased ranges would have multiple applications in the fields of biomedical sensing and AR/VR. A prime example of an application that our solution would enable is real-time tracking of muscle and joint movements across the entire human body using just one pair of tracking magnets and a magnetometer array. In comparison, the existing state-of-the-art-techniques would either perform offline and non-real-time tracking or require multiple magnetometer arrays to cover the area of a human body.

## 2 Related Work

### 2.1 Object Tracking Systems

There are many alternatives to a magnet based object tracking system. In this section we discuss the advantages and disadvantages of the same.

#### 2.1.1 Optical Sensing

These systems use computer vision, light sources and optical sensors to calculate the location and orientation of the tracked object. Typically the object is tracked either by adding a light source to the object itself, or by measuring the interference caused by the object to light emitted by an external light source [16] [17]. Some other techniques overcome the need to have light sources by using optical markers placed on the object which are then tracked with the use of cameras [25]. However, a minimum amount of ambient light level is needed for the marker based technique to work. Similarly, while the tracking accuracy of all of the optical systems is good, the key limitation is chance of visual occlusion.

#### 2.1.2 Inertial Sensing

Systems that use inertial sensing typically add Inertial Measurement Units (IMUs) to the tracked object [12]. Due to the low latency of IMUs, inertial sensing systems typically provide accurate relative tracking for fast motions. However, due to the significant presence of linear and non-linear drift in IMUs, such systems do not perform well over a long period of time and add significant error when tracking a slow moving object. Similarly, such systems required actively powered unit on the tracked object.

#### 2.1.3 Acoustic Sensing

Ultrasonic time of flight based sensors are also used for tracking an object [14]. While systems which use these sensors are good at predicting the location of a stationary object, due to multi-path reception, they are not good at tracking the pose and movement.

### 2.2 Tracking Permanent Magnets

Systems like [13] and [23] use a magnetometer array to track passive magnets. Both of these physics based analytical models use mathematical optimization techniques to precisely estimate the location and pose of a passive magnet. However, neither of these systems try to cover a long range of tracking ( 50 cms). Larger areas lead to key issues with analytical models, wherein they often fail to converge on to the global minima of the optimization. Systems like [23] use a hybrid approach where they use an analytical model to track objects which are in the vicinity of the magnetometers ( $< 20$  cm) and then use a neural network based approach to track magnets which are further away from the magnetometer ( $> 20$  cm). This is done as they observe that while analytical models perform better when the magnets are closer, data-centric approaches yield lesser relative errors when the distances increase. However, one key limiting factor in this work is the fact that as their system has a high latency they are not able to use it for real-time applications.

### 2.3 Tracking Electro-Magnets

Works like [15],[20] use electromagnets mounted on fingertips to track finger movements. While these systems showcase high accuracy in tracking of the fingertip, they are mainly restricted by the range of sensing which is not more than 15 cm and requirement for the tracker to be provided with a power source. A similar work [24] uses electromagnets to track human joint-angles in real-time. However, this system is also restricted by the range and the need to be powered continuously.

## 3 Hardware Setup

In this section, we present the hardware setup we use to collect data for and to test our magnet tracking systems.

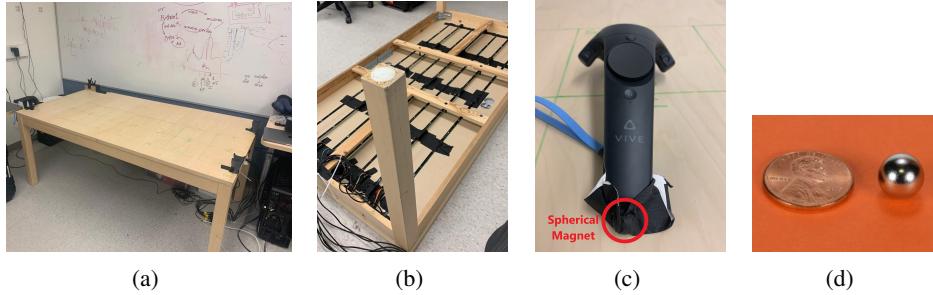


Figure 1: Hardware setup: (a) and (b) Table and Magnetometers mounted underneath the table, (c) Vive Tracker with spherical magnet attached, and (d) Spherical Magnet S6

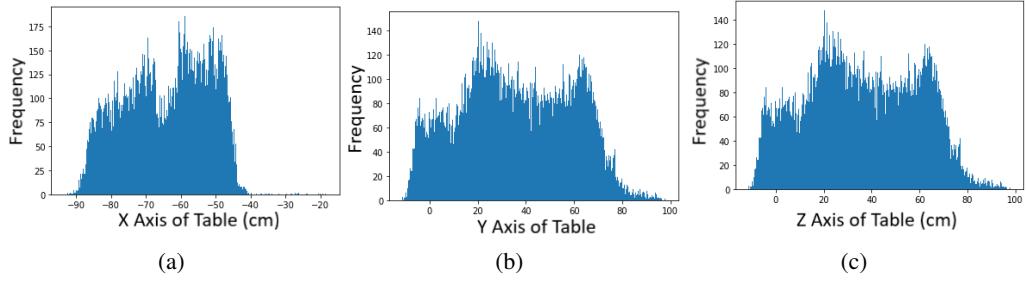


Figure 2: Distribution of data over (a) X Axis of table, (b) Y Axis of table, (c) Z Axis of table

### 3.1 Magnetometer Array

Figure 1 (a), (b) shows our setup which consists of a 2D magnetometer array of 112 MLX90393 magnetometers [5] mounted on a 160 cm \* 85 cm wooden table in a 7 rows and 16 columns configuration. The spacing between 2 magnetometers is 10cm. We use seven Adafruit Feather nRF52832 [1], one for each row, for interfacing with the magnetometers over I2C. Each feather collects data from 16 magnetometers and sends it over serial port to our computing unit (Intel i7 2019, 32GB RAM). The sampling frequency of our data collection pipeline is 60 Hz. We use the following settings for our magnetometers: *Gain: 2x, Oversampling: 0, Digital Filter: 2*.

### 3.2 Ground Truth

We use a HTC Vive [11] Tracker to provide us with ground truth information of location and pose of the permanent magnet which we want to track. Figure 1 (c), highlights the Vive light tower which tracks the Vive controller shown in Figure 1 (d).

### 3.3 Permanent Magnet

We use a Neodymium N42 spherical magnet of diameter 5/8" [8] shown in 1 (e) for all of our experiments. This magnet is mounted at the end of the Vive tracker as shown in 1 (d).

## 4 Data Collection

We collected 102,231 datapoints over a sub-region on the table of dimensions 30cm \* 40cm \* 50cm. Each sample in this dataset is as follows:

$$< \text{timestamp}, x, y, z, \phi, \theta, B_{x_1}, B_{y_1}, B_{z_1}, \dots, B_{x_{112}}, B_{y_{112}}, B_{z_{112}} > \quad (1)$$

Figure 2 (a), (b), (c) highlight the distribution of data over  $x$ ,  $y$  and  $z$  axes respectively. We use this collected data to evaluate the analytical and data centric models discussed in Sections 5 and 6.

### 5 Baseline Approach: Analytical Model

In this work, we develop and evaluate the physics-based analytical model as a baseline to allow us to better understand its capabilities and help us with developing a hybrid approach which uses both the physics-based analytical model that is presented here and the data-centric model that we introduce in Section 6.

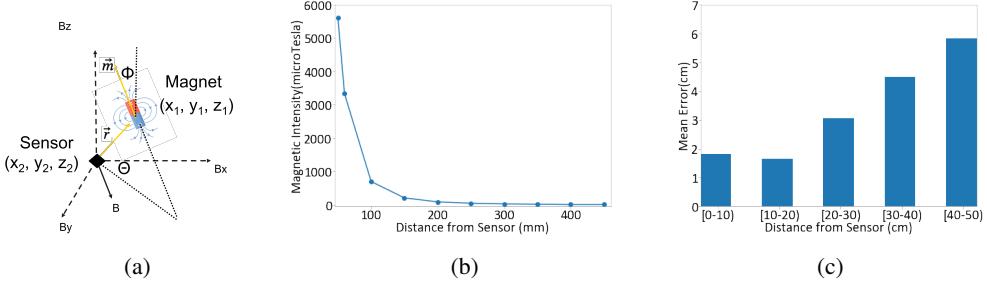


Figure 3: (a) Magnet and Magnetometer in Free Space, (b) Magnetic Field Intensity vs Distance from N42 Magnet, (c) Mean Error of location prediction at different distances

### 5.1 System Modelling

The analytical model relies on first principle physics based modelling to compute the location and pose of the magnet by utilising readings from magnetometers which are in the vicinity of the magnet. Figure 3 (a) shows a magnet and a magnetometer in free space, where the location and pose of the magnet relative to the magnetometer is characterised by the parameters  $x, y, z, \theta, \phi$ . These location and pose parameters can be used to estimate the magnetic field at the magnetometer with the help of the dipole model. We note that the dipole model is directly applicable to spherical magnets which we use, as the magnetic field of the magnet outside of the body of a finite radius spherical magnet is the same as that of a ideal point dipole. Thus with this, the analytical model converts the problem of finding the location and pose of the magnet into an optimisation task where the parameters  $x, y, z, \theta, \phi$  are optimised by using the dipole model and the real-world magnetometer readings.

Optimisation techniques for solving this particular problem have previously been presented by works like [23]. The techniques model the cost function as a difference in between the predicted magnetic field at each magnetometer, given a particular pose and orientation of the magnet, and the real-world magnetometer readings. More specifically, given a 2-D array of magnetometers, the magnetic field prediction error ( $E_i$ ) at the  $i^{th}$  sensor, is the difference between the measured magnetic field ( $\tilde{B}_i$ ) and the predicted magnetic field ( $B_i$ ). Thus,

$$E_i = B_i - \tilde{B}_i \quad (2)$$

Now if we let the estimated location of the  $j^{th}$  magnet be  $(x_j, y_j, z_j)$  and the position of the  $i^{th}$  sensor be  $(s_{ix}, s_{iy}, s_{iz})$ , a vector from the  $j^{th}$  magnet to the  $i^{th}$  sensor is given as follows :

$$\mathbf{r}_{ij} = (s_{ix} - x_j) \hat{\mathbf{x}} + (s_{iy} - y_j) \hat{\mathbf{y}} + (s_{iz} - z_j) \hat{\mathbf{z}} \quad (3)$$

Using  $z$  axis to be a reference,

$$\mathbf{m}_j = m_j (\sin \theta_j \cos \phi_j \hat{\mathbf{x}} + \sin \theta_j \sin \phi_j \hat{\mathbf{y}} + \cos \theta_j \hat{\mathbf{z}}) \quad (4)$$

where  $\theta_j$  and  $\phi_j$  are the magnet's estimated orientation from vertical and around vertical, respectively, and  $m_j$  is the magnet strength.

$$m_j = \frac{B_{rj}}{\mu_0} \frac{4}{3} \pi R_j^3 \quad (5)$$

Using the equation for the magnetic field of a dipole, the magnetic field prediction,  $B_i = (B_{ix}, B_{iy}, B_{iz})$ , at the  $i^{th}$  sensor can then be expressed as :

$$\mathbf{B}_i = \mathbf{G} + \sum_{j'=1}^{j'=M} \left( \frac{3\mathbf{r}_{ij'} (\overline{\mathbf{m}}_{j'}^\top \cdot \mathbf{r}_{ij'})}{r_{ij'}^5} - \frac{\overline{\mathbf{m}}_{j'}}{r_{ij'}^3} \right) \quad (6)$$

where,

$$\overline{\mathbf{m}}_j = \frac{\mu_0}{4\pi} \mathbf{m}_j \quad (7)$$

$\overline{\mathbf{m}}_j$  is the magnetic strength.

$G_i = (G_{ix}, G_{iy}, G_{iz})$  is an estimate of the spatially uniform disturbance field.

The three components of the magnetic field are as follows :

$$\begin{aligned}
 B_{ix} &= G_x + \sum_{j'=1}^{j'=M} \bar{m}_{j'} \left( \frac{3\bar{x}_{ij'}(\sin \theta_{j'} \cos \phi_{j'} \bar{x}_{ij'} + \sin \theta_{j'} \sin \phi_{j'} \bar{y}_{ij'} + \cos \theta_{j'} \bar{z}_{ij'} )}{(\bar{x}_{ij'}^2 + \bar{y}_{ij'}^2 + \bar{z}_{ij'}^2)^{5/2}} - \frac{\sin \theta_{j'} \cos \phi_{j'}}{(\bar{x}_{ij'}^2 + \bar{y}_{ij'}^2 + \bar{z}_{ij'}^2)^{3/2}} \right) \\
 B_{iy} &= G_y + \sum_{j'=1}^{j'=M} \bar{m}_{j'} \left( \frac{3\bar{y}_{ij'}(\sin \theta_{j'} \cos \phi_{j'} \bar{x}_{ij'} + \sin \theta_{j'} \sin \phi_{j'} \bar{y}_{ij'} + \cos \theta_{j'} \bar{z}_{ij'} )}{(\bar{x}_{ij'}^2 + \bar{y}_{ij'}^2 + \bar{z}_{ij'}^2)^{5/2}} - \frac{\sin \theta_{j'} \sin \phi_{j'}}{(\bar{x}_{ij'}^2 + \bar{y}_{ij'}^2 + \bar{z}_{ij'}^2)^{3/2}} \right) \\
 B_{iz} &= G_z + \sum_{j'=1}^{j'=M} \bar{m}_{j'} \left( \frac{3\bar{z}_{ij'}(\sin \theta_{j'} \cos \phi_{j'} \bar{x}_{ij'} + \sin \theta_{j'} \sin \phi_{j'} \bar{y}_{ij'} + \cos \theta_{j'} \bar{z}_{ij'} )}{(\bar{x}_{ij'}^2 + \bar{y}_{ij'}^2 + \bar{z}_{ij'}^2)^{5/2}} - \frac{\cos \theta_{j'}}{(\bar{x}_{ij'}^2 + \bar{y}_{ij'}^2 + \bar{z}_{ij'}^2)^{3/2}} \right)
 \end{aligned} \tag{8}$$

## 5.2 Implementation

We solve equations 8 using Levenberg-Marquardt (LM) algorithm [19] which minimizes the cost function given by equation 2. To achieve this, the LM algorithm computes the local minimum iteratively using an initial guess and the analytical gradient for each of the following parameters  $x_{ij}, y_{ij}, z_{ij}, \theta_j, \phi_j, G_x, G_y, G_z, m_j$ . We use the LM optimisation implemented by Ceres-Solver [2] and run it till convergence. Additionally, we put a Kalman Smoother [7] on the output of the LM optimisation with a lag of N=4 to remove the noise in predictions.

## 5.3 Results and Drawbacks of the Analytical Model

The cost function 2 is non-convex. Hence, the solution to the optimisation problem is heavily dependent on the initial guess. An initial guess that severely differs from the actual location and pose would lead to the LM optimiser getting stuck in a local minima. This issue is evident as the magnet starts going farther away from the sensing array. Figure 3 (b) shows the decreasing magnetic intensity with increasing sensing distance. This causes the non-convexity of the cost function 2 to increase, further increasing the dependence on an optimal initial guess. By running on our collected dataset with the last temporal datapoint as the initial guess, we see in Figure 3 (c), mean error increases significantly when sensing distance increases. Thus, it is imperative to have an initial guess as close as possible to the actual location to increase chance of convergence to a global minimum.

## 6 Baseline Approach: Data Centric Model

While the analytical model takes a physics-based first principle approach to solving the magnet tracking problem, the data-centric method relies on learning location and pose of the magnet as a function of magnetometer readings by training a regressor on labelled data.

### 6.1 Feature Engineering

Our setup consists of 112 magnetometers, each giving 3 data-streams in the form of magnetic field over  $X, Y$  and  $Z$  axes. There is a good chance of over-fitting if this raw magnetometer data from 112 magnetometers, each having data from 3 axes, is used as it is to train a regressor. Hence, we develop a feature vector that can capture the key information from the magnetometer data.

The first step towards designing robust features is to ensure it has less susceptibility to noise. For this, we collect N=10000 samples from all 112 magnetometers when there is no permanent magnet in the vicinity of the magnetometers. We then use this data to compute a mean and standard deviation for each of the  $112 * 3 = 336$  sensor values. Essentially, we record a background noise information in the form of a  $112 * 3 * 2 = 772$  valued tuple:

$$< \mu_{1x}, \sigma_{1x}, \mu_{1y}, \sigma_{1y}, \mu_{1z}, \sigma_{1z}, \dots, \mu_{112z}, \sigma_{112z} > \tag{9}$$

We then use this tuple to compute a z-score [6] of each sensor value recorded when tracking a permanent magnet. A z-score is given by,

$$z\_score = (x - \mu) / \sigma \tag{10}$$

Thus, we end up with an intermediate feature vector for one sample of the magnet tracking system which looks like,

$$< z\_score_{1x}, z\_score_{1y}, z\_score_{1z}, \dots, z\_score_{112x}, z\_score_{112y}, z\_score_{112z} > \tag{11}$$

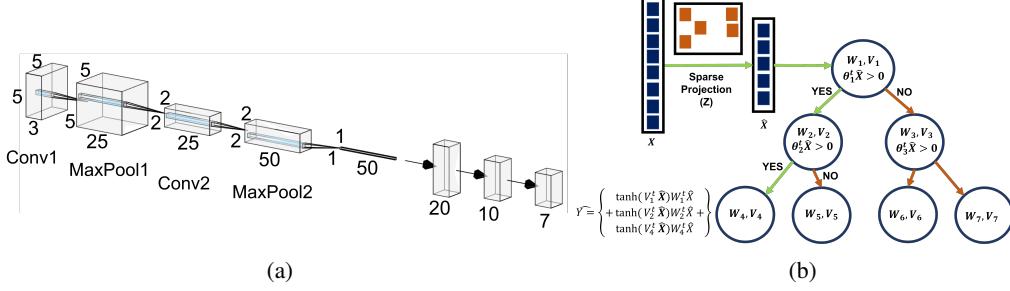


Figure 4: (a) Deep Neural Net Architecture, (b) Bonsai Architecture

Therefore, by computing the z-scores we believe that effect of both, hard and soft iron drifts [9] and sensor noise are negated to a good degree. We then reshape this intermediate feature vector into an array of shape (7,16,3) to physically and spatially represent our setup. While we discuss our hardware setup in Section 3, the reshaped array can be interpreted by knowing that our sensors are arranged into 7 rows, 16 columns and each sensor has 3 data streams, one each for  $X$ ,  $Y$ ,  $Z$  axes. Thus, the reshaped intermediate feature vector would look like,

$$\begin{bmatrix} z\_scores_{(1,1)} & z\_scores_{(1,1)} & \dots & z\_scores_{(1,16)} \\ z\_scores_{(2,1)} & z\_scores_{(2,1)} & \dots & z\_scores_{(2,16)} \\ \vdots & \vdots & \ddots & \vdots \\ z\_scores_{(7,1)} & z\_scores_{(7,1)} & \dots & z\_scores_{(7,16)} \end{bmatrix} \quad (12)$$

where each

$$z\_scores_{(i,j)} = [z\_score_{(i,j)_x}, z\_score_{(i,j)_y}, z\_score_{(i,j)_z}] \quad (13)$$

Then, from this intermediate and reshaped feature vector 12, we compute a 2D subarray of size  $M * M$  which has the highest sum of  $z\_scores$  across all possible sub-arrays. Thus, with this we are able to zoom in upon the area of interest and remove features which would contribute less towards inference. This step would hence help in avoiding over-fitting on data. Now, we reshape this feature vector of the sub-array back to a 1-D row and we then add the  $x$  and  $y$  physical coordinates of the centre of the sub-array. This step adds information about the physical space to our feature vector. With this, our final feature vector of length  $M * M + 2$ , is represented by,

$$< z\_score_{subarray(1,1)_x}, z\_score_{subarray(1,1)_y}, z\_score_{subarray(1,1)_z}, \dots, z\_score_{subarray(M,M)_x}, z\_score_{subarray(M,M)_y}, z\_score_{subarray(M,M)_z}, x_{subarray\_centre}, y_{subarray\_centre} > \quad (14)$$

## 6.2 Learning Algorithms

We propose and evaluate three different learning algorithms for location and pose prediction as a baseline for our data-centric approach.

### 6.2.1 Random Forest Regressor

We train a Random Forest Regressor to predict the location and pose given by  $[x, y, z, \theta, \phi]$ . Each sample from training and testing set contains the ground truth  $[x, y, z, \theta, \phi]$ ,  $112 * 3$  magnetometer readings which are processed and converted to a feature vector represented in equation 14 with  $M = 3$ . We use scikit-learn's [3] implementation of *ExtraTreesRegressor* [4] and set the hyper-parameters as ( $n\_jobs = -1$ ,  $n\_estimators = 200$ ,  $random\_state = 42$ ). We also implement K-fold cross validation for training and testing with  $K=10$ .

### 6.2.2 Deep Neural Network

We train a deep neural network to predict the location and pose of the magnet. The location is given by three co-ordinates  $[x, y, z]$ . The pose co-ordinates given by  $[\theta, \phi]$  are converted to quaternions [10]  $[qw, qx, qy, qz]$  which provide better interpolation and can be estimated by our neural network better. The  $112 * 3$  magnetometer readings are converted to a feature vector represented in equation 14 with  $M = 5$ . Figure 4 shows the architecture of the neural network. The network makes use of 2 convolutional and max-pooling layer sequences to preserve the spatial information in the input feature matrix. These are followed by 3 fully-connected layers, of which the final layer is the output

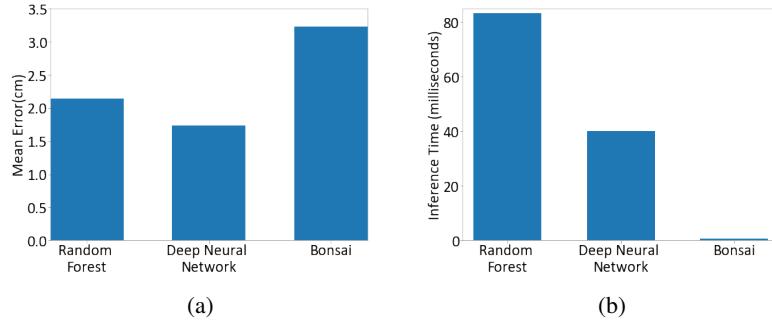


Figure 5: (a) Location Prediction Error, (b) Location Prediction Inference Time

later having 7 nodes. 3 of these nodes predict the location  $[x, y, z]$  and the remaining 4 predict the quaternion co-ordinates  $[qw, qx, qy, qz]$ . We also implement K-fold cross validation for training and testing with  $K = 5$ .

#### 6.2.3 Bonsai Model

Bonsai is a shallow tree-based machine learning algorithm developed for efficient prediction on IoT devices [18]. We use a Bonsai architecture with depth = 3 and projection = 28, as shown in figure 4. Bonsai being a shallow regressor with powerful nodes can reduce inference time and still model non-linear data with the help of a non-linear  $\tanh$  function.

### 6.3 Results and Drawbacks of Baseline Data Centric Approaches

Data centric models are able to accurately predict the location even at higher distances from the sensors. Figure 5 (a) shows the mean error achieved by each of the three data-centric approaches. Similarly, Figure 5 (b) shows the inference time of each. We can see that in terms of prediction accuracy, the deep neural net performs the best, however, Bonsai based approach gives the  $50x$  faster inference while adding an error of 1.43cm. Hence, while one approach minimizes error in prediction, the other minimizes inference time. Thus, it is necessary to develop a new approach which can achieve both.

## 7 Novel Approach: Hybrid Model

We propose a novel hybrid approach which combines the long-range capabilities of data centric algorithms and accuracy of the analytical model to give fast and long-range tracking. The analytical model has a non-convex function. With an initial guess that is far away from the actual location and pose, the analytical model typically converges at local minima. This problem gets amplified at higher distances from the sensor. To resolve this, we propose using a moderately accurate and a low latency data-centric model like Bonsai to provide an initial guess that is very close to the actual location. Such "almost-there" initial guess would significantly increase the chance of convergence to a global minima. Hence, such an approach would tackle the deficiencies of the analytical model at higher distances while still maintaining real-time and accurate operation.

### 7.1 Implementation

Figure 6 (a) shows the block diagram of the our novel hybrid approach. First the raw magnetometer data is fed to a trained Bonsai model which would make an intermediate prediction of location. This predicted location would then be used as the seed point for the LM optimisation. Hence, the first step comprises of using the data-centric approach to give an approximate location and the second step would use physics-informed analytical approach to accurately predict the location.

## 8 Results

We conduct an evaluation of the proposed hybrid model discussed in Section 7 on 102,231 datapoints collected with our real-world hardware setup. Over the entire data, the hybrid model gives a mean error of 1.81cm, 2.39cm and 2.36cm over X, Y, and Z axes respectively. Figure 6 (b) shows comparison of ground truth data and the prediction made by the hybrid model for X, Y and Z axis separately.

7 (a) shows the mean error of the hybrid model with a Bonsai predictor compared with pure data centric models. The figure shows that the hybrid model outperforms all data centric approaches in X axis location prediction. In Y axis and Z axis prediction, only the deep neural network predictor outperforms the hybrid model. On all three axes, the hybrid model with a Bonsai predictor

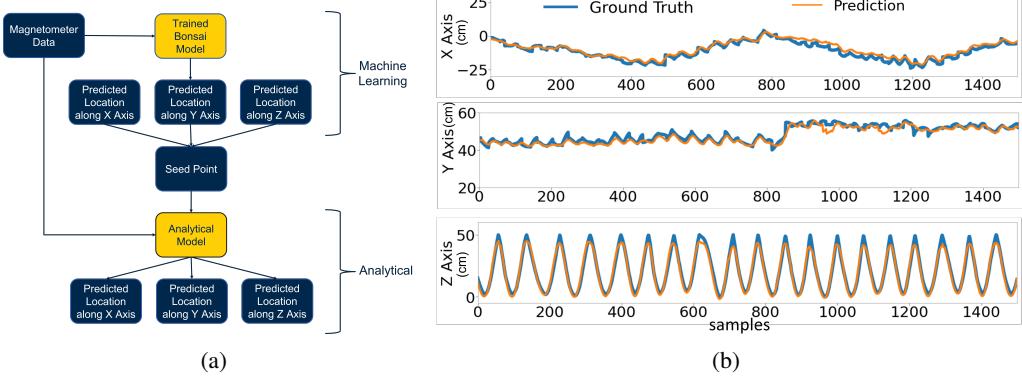


Figure 6: (a) Hybrid Approach Architecture, (b) Ground Truth and Prediction by Hybrid approach on test data

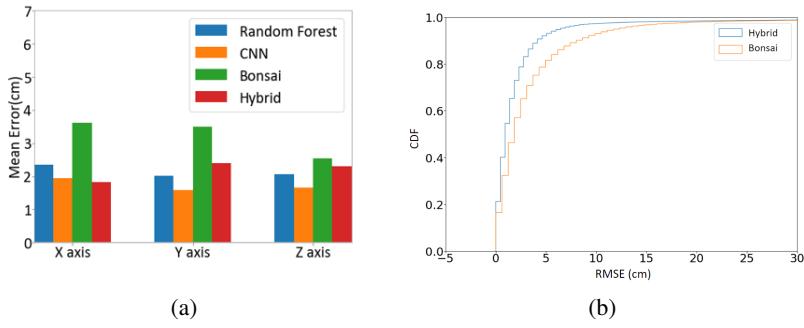


Figure 7: Ground Truth and Prediction by Hybrid approach on test data

outperforms a pure Bonsai prediction model, showing that the hybrid model is more accurate than its corresponding data centric model. Similarly 7 (b) shows the CDF of errors of only the data-centric Bonsai model and the hybrid model. Thus the hybrid model extends tracking range of analytical model from 10-15 cm to 40-45 cm and at the same time maintains high accuracy.

## 9 Future Work

Our developed hybrid approach gives high tracking accuracy on data that we collected. However, our collected dataset, while almost uniformly covering location variations ( $[x, y, z]$ ) does not cover a high amount of pitch and yaw variations ( $[\theta, \phi]$ ). Hence, as of now, the robustness of the proposed hybrid approach to variations in orientation ( $[\theta, \phi]$ ) of the magnet has not been tested extensively. We plan to collect more data with high amount of variations in orientation and then re-evaluate the efficacy of our hybrid magnet tracking approach.

## 10 Conclusion

In this work, we present a real-world system for magnet tracking. Our physics-based analytical model performs optimally when magnets are closer in addition to providing low latency and accuracy, whereas the data-centric approach is found to increase effectiveness over longer ranges. The results show that our proposed hybrid model is the most optimal solution for fast and long-range tracking with low prediction error.

## 11 Author Contributions

All authors contributed equally to this project.

## 12 Appendix

Here are the links to Demo Video and Code:

[Demo Video Link](#)

[Project Code Link](#)

## References

- [1] Adafruit feather nrf52832. [https://cdn-learn.adafruit.com/downloads/pdf/bluefruit-nrf52-feather-learning-guide.pdf/](https://cdn-learn.adafruit.com/downloads/pdf/bluefruit-nrf52-feather-learning-guide.pdf).
- [2] Ceres solver. <http://ceres-solver.org>.
- [3] scikit-learn. [https://scikit-learn.org/stable//](https://scikit-learn.org/stable/).
- [4] sklearn.ensemble.Extratreesregressor. [https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html/](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html).
- [5] MLx90393 datasheet by melexis technologies nv, 2017. [https://www.digikey.com/htmldatasheets/production/1686159/0/0/1/mlx90393-datasheet.html?utm\\_adgroup=xGeneral&utm\\_source=bing&utm\\_medium=cpc&utm\\_campaign=Dynamic%20Search\\_EN\\_Product&utm\\_term=digikey&utm\\_content=xGeneral&utm\\_id=bi\\_cmp-420474294\\_adg-1309518711610365\\_ad-81844994178165\\_dat-2333644710616871:loc-190\\_dev-c\\_ext-\\_prd-&msclkid=5e6f9f4dc4e71933c187fd676134f41b](https://www.digikey.com/htmldatasheets/production/1686159/0/0/1/mlx90393-datasheet.html?utm_adgroup=xGeneral&utm_source=bing&utm_medium=cpc&utm_campaign=Dynamic%20Search_EN_Product&utm_term=digikey&utm_content=xGeneral&utm_id=bi_cmp-420474294_adg-1309518711610365_ad-81844994178165_dat-2333644710616871:loc-190_dev-c_ext-_prd-&msclkid=5e6f9f4dc4e71933c187fd676134f41b).
- [6] What is a z-score?, 2020. <https://datascience.eu/mathematics-statistics/what-is-a-z-score/>.
- [7] Kalman filter, 2022. [https://en.wikipedia.org/wiki/Kalman\\_filter](https://en.wikipedia.org/wiki/Kalman_filter).
- [8] Kj magnetics, 2022. <https://www.kjmagnetics.com/proddetail.asp?prod=S6>.
- [9] Magnetometer hard and soft iron calibration, 2022. <https://www.vectornav.com/resources/inertial-navigation-primer/specifications--and--error-budgets/specs-hsicalibration#:~:text=Hard%20iron%20distortions%20will%20only,warp%20the%20existing%20magnetic%20fields>.
- [10] Quaternion, 2022. <https://en.wikipedia.org/wiki/Quaternion>.
- [11] Vive tracker, 2022. <https://www.vive.com/us/accessory/vive-tracker/>.
- [12] Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. Itracku: Tracking a pen-like instrument via uwb-imu fusion. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys ’21, page 453–466, New York, NY, USA, 2021. Association for Computing Machinery.
- [13] Dongyao Chen, Mingke Wang, Chenxi He, Qing Luo, Yasha Iravantchi, Alanson Sample, Kang G. Shin, and Xinbing Wang. Magx: Wearable, untethered hands tracking with passive magnets. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, MobiCom ’21, page 269–282, New York, NY, USA, 2021. Association for Computing Machinery.
- [14] Jian Chen, Fan Yu, Jiaxin Yu, and Lin Lin. A three-dimensional ultrasonic pen-type input device with millimeter-level accuracy for human-computer interaction. *IEEE Access*, 8:143837–143847, 2020.
- [15] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI ’16, page 1504–1514, New York, NY, USA, 2016. Association for Computing Machinery.
- [16] Jaehyun Han, Seongkook Heo, Hyong-Euk Lee, and Geehyuk Lee. The irpen: A 6-dof pen for interaction with tablet computers. *IEEE Computer Graphics and Applications*, 34(3):22–29, 2014.
- [17] Seongkook Heo, Jaehyun Han, Sangwon Choi, Seunghwan Lee, Geehyuk Lee, Hyong-Euk Lee, SangHyun Kim, Won-Chul Bang, DoKyoon Kim, and ChangYeong Kim. Ircube tracker: An optical 6-dof tracker based on led directivity. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST ’11, page 577–586, New York, NY, USA, 2011. Association for Computing Machinery.

- [18] Ashish Kumar, Saurabh Goyal, and Manik Varma. Resource-efficient machine learning in 2 KB RAM for the internet of things. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1935–1944. PMLR, 06–11 Aug 2017.
- [19] Jorge J More. The levenberg-marquadt algorithm: Implementation & theory. 1978.
- [20] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. Auraring: Precise electromagnetic finger tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), dec 2019.
- [21] S. Tarantino, F. Clemente, D. Barone, M. Controzzi, and C. Cipriani. The myokinetic control interface: tracking implanted magnets as a means for prosthetic control. *Scientific Reports*, 7(1):17149, Dec 2017.
- [22] C. R. Taylor, S. S. Srinivasan, S. H. Yeon, M. K. O’Donnell, T. J. Roberts, and H. M. Herr. Magnetomicrometry. *Science Robotics*, 6(57):eabg0656, 2021.
- [23] Cameron R. Taylor, Haley G. Abramson, and Hugh M. Herr. Low-latency tracking of multiple permanent magnets. *IEEE Sensors Journal*, 19(23):11458–11468, 2019.
- [24] Amanda Watson, Andrew Lyubovsky, Kenneth Koltermann, and Gang Zhou. Magneto: Joint angle analysis using an electromagnet-based sensing method. In *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (Co-Located with CPS-IoT Week 2021)*, IPSN ’21, page 1–14, New York, NY, USA, 2021. Association for Computing Machinery.
- [25] Po-Chen Wu, Robert Wang, Kenrick Kin, Christopher Twigg, Shangchen Han, Ming-Hsuan Yang, and Shao-Yi Chien. Dodecapen: Accurate 6dof tracking of a passive stylus. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST ’17, page 365–374, New York, NY, USA, 2017. Association for Computing Machinery.