Data-Driven Control for a Milli-Scale Spiral-Type Magnetic Swimmer using MPC

Haoran Zhao*, Yitong Lu, Aaron T. Becker, and Julien Leclerc

Abstract— **This paper presents four data-driven system models for a magnetically controlled swimmer. The models were derived directly from experimental data, and the accuracy of the models was experimentally demonstrated. Our previous study successfully implemented two non-model-based control algorithms for 3D path-following using PID and model reference adaptive controller (MRAC). This paper focuses on system identification using only experimental data and a model-based control strategy. Four system models were derived: (1) a physical estimation model, (2, 3) Sparse Identification of Nonlinear Dynamics (SINDY), linear system and nonlinear system, and (4) multilayer perceptron (MLP). All four system models were implemented as an estimator of a multi-step Kalman filter. The maximum required sensing interval was increased from $180\,\mathrm{ms}$ to $420\,\mathrm{ms}$ and the respective tracking error decreased from $9\,\mathrm{mm}$ to $4.6\,\mathrm{mm}$. Finally, a Model Predictive Controller (MPC) implementing the linear SINDY model was tested for 3D path-following and shown to be computationally efficient and offers performances comparable to other control methods.**

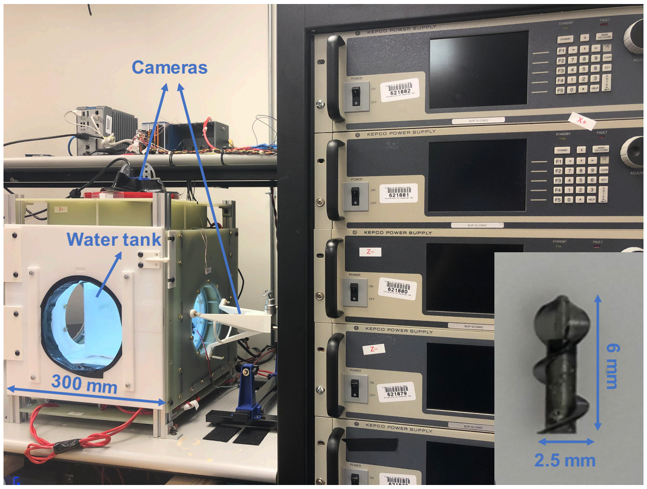


Fig. 1. Photo of magnetic manipulator. (Inset) A magnetic swimmer.

I. Introduction

Magnetic robots have attracted significant attention in recent years due to their size and functionality. The size of the robots vary from millimeter scale [1]–[4] to micrometer scale [5] and nanometer scale [6]. The fluidic environments vary from water [1]–[3], [7] to a low-Reynolds-number regime [8]. The driven mechanisms range from gradient force [9] to torque produced by magnetic field [1]–[5]. These magnetic machines show great potentials, especially for clinical and medical applications, such as diagnosis [10]–[12], drug delivery [10], [13], and minimally invasive surgery [14]–[19]. Because these magnetic robots can be navigated through bodily fluids to perform their tasks, precise position control is often essential. The magnetic robots are actuated by an external magnetic field, so the motion mode, speed, and orientation of the robots depend heavily on properties of the external magnetic field, such as flux density, frequency, and direction of the field. Precise positioning control can be achieved by tuning the parameters of the external field and by advanced controller design.

In 1976, Purcell presented a flagellar propulsion mechanism for low Reynolds number (Re) environments [20], then in 1977, Purcell proposed a general model for this mechanism [21]. Since then, there have been many magnetic machines designed with spherical or helical heads and helical tails [22]. Moreover, many researchers have advanced the modeling and analysis of such systems. The force, torque, translation, and rotation of helical robots are often linked by a propulsion matrix, which is a function of geometric parameters [23] and fluid viscosity [24]. Abbott et al. [25] explained that as microrobot size decreases or as the distance increases from the magnetic field sources, helical rotation becomes more efficient than oscillations for propulsion force. Modeling these was simplified when Wang et al. [26] experimentally and analytically demonstrated that the robot's forward and rotational velocity ratio is independent of their surface coatings if the swimmers are operated below their step-out frequency. However, interfacial slippage was observed during their experiments. This slippage is nontrivial, and should be considered into a helical swimmer model. Wang et al. [23] established a comprehensive dynamic model to analyze the swimming properties that influence velocity, step-out frequency, and maximum velocity. They provided guidance for helical microrobot design. Additional magnetic control system theory and system analyses were developed and presented in [27]–[29].

Our work focuses on spiral-type magnetic swimmers. A spiral type has a screw-like shape [30] rather than a magnetic head connected to a tail shaped like an compression spring. Our magnet is inserted into a central cavity along the central axis. Although the swimming properties of a spiral-type magnetic swimmer were theoretically analyzed using the 3D finite volume method in [31], [32], the design of our swimmer was refined by redesigning the tip shape and experimentally optimizing dimensions in our previous

This work was supported by the National Science Foundation under Grants No. [IIS-1553063] and [CNS-1932572].

Authors are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77204.
*Corresponding author.
Email: hzhao9.uh.edu

work [1]. Our system is a highly non-linear system that does not satisfy the superposition principle. The system is unstable without a closed-loop controller. Thus, conventional system identification algorithms such as the eigensystem realization algorithm (ERA) [33] cannot be employed because the swimmer's position is unstable under open loop control. However, in our previous work [1]–[4] we successfully navigated our swimmer in a 3D environment with a non-model-based controller, which provided a large amount of experimental data.

In this paper, instead of using conventional methods or finite element analysis software, we demonstrate the feasibility of using completely data-driven methods to estimate a system model for the swimmer state prediction. We also show methods to develop controllers using these models. The inverse magnetics calculations for controlling the magnetic swimmer with the external magnetic field are presented in Section II-A (using our method from [1]–[4], [34]). The pure data-driven algorithms used for estimating the system model are illustrated in Section II. Section III presents the experimental study and main results of this paper. Finally, the conclusion and discussion are in Section IV.

II. Methodology

This section illustrates the algorithms employed for the data-driven system identification and model-based controller, MPC.

A. Physical estimation model

A physical estimation model is developed to predict the swimmer's velocity and position as a function of the control input. The model first acquires the swimmer's initial position $\boldsymbol{P}_0$, and assumes the initial velocity $\boldsymbol{V}_0$ of point $\boldsymbol{P}_0$ is zero. The velocity $\boldsymbol{V}_i$ of $\boldsymbol{P}_i$ can be calculated as $\boldsymbol{V}_i = (\boldsymbol{P}_i - \boldsymbol{P}_{i-1})/\Delta t$. Figure 2 shows the free-body diagram for the model.

The propulsion force $\boldsymbol{F}_r$ produced by the rotating magnetic swimmer was assumed to be aligned with its rotational axis. It can be represented as $\boldsymbol{F}_r = k \cdot \boldsymbol{\omega}$, where k is the thrust coefficient to be tuned. The drag force $\boldsymbol{F}_d$ was computed using a simplified fluid mechanics model:

$$\boldsymbol{F}_d = -C_d \cdot \boldsymbol{V}_i \tag{1}$$

$$C_d = C_{d0} \cdot \sin(\alpha) + C_{d1}, \tag{2}$$

where C_d is the drag coefficient and C_{d0} and C_{d1} are parameters to be tuned. The angle of attack α is determined by the angle between the rotational vector $\boldsymbol{\omega}$ and the velocity vector $\boldsymbol{V}_i$. The lift force $\boldsymbol{F}_\ell$ can be calculated by

$$\boldsymbol{F}_\ell = C_\ell \cdot \|\boldsymbol{V}_i\| \cdot \hat{\boldsymbol{v}}_\perp \tag{3}$$

$$C_\ell = C_{\ell 0} \cdot \sin(2\alpha), \tag{4}$$

where C_ℓ is the lift coefficient and $C_{\ell 0}$ is the parameter to be tuned. $\hat{\boldsymbol{v}}_\perp$ is the unit vector perpendicular to the velocity vector $\boldsymbol{V}_i$ and in the plane of $\boldsymbol{\omega}$ and $\boldsymbol{V}_i$. The vertical static force $\boldsymbol{F}_{\text{vs}}$ represents the sum of the weight and buoyancy force of the swimmer, where $\boldsymbol{F}_{\text{vs}} = F_{\text{vs}} \cdot [0, 0, 1]$, and the magnitude of the force F_{vs} can be tuned. To calculate the centripetal force $\boldsymbol{F}_c$, we determine the direction of the force by

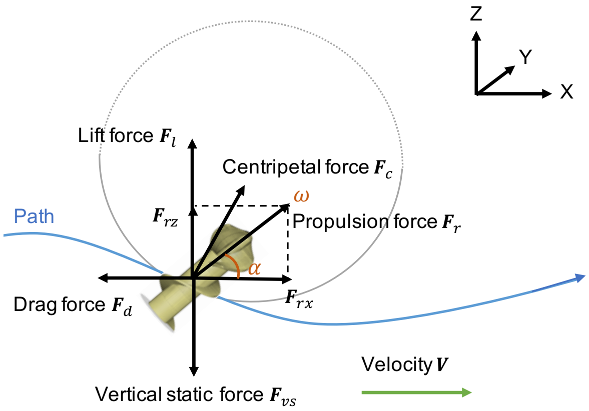

Z
Y
X
Lift force $\boldsymbol{F}_l$
Centripetal force $\boldsymbol{F}_c$
ω
Path
$\boldsymbol{F}_{rz}$
Propulsion force $\boldsymbol{F}_r$
α
Drag force $\boldsymbol{F}_d$
$\boldsymbol{F}_{rx}$
Velocity $\boldsymbol{V}$
Vertical static force $\boldsymbol{F}_{vs}$

Fig. 2. Free-body diagram for the *Physical* model. The grey circle indicates the XY plane.

$$\hat{\boldsymbol{c}} = -\frac{\boldsymbol{P}_i}{\|\boldsymbol{P}_i\|}, \tag{5}$$

where the position of the center of the manipulator is defined as the origin. The $\boldsymbol{F}_c$ is approximately equal to zero when the swimmer is at the center of the manipulator or near the workspace boundary, so the $\boldsymbol{F}_c$ is calculated as

$$\boldsymbol{F}_c = F_{\text{max}} \cdot \sin\left(\pi \frac{\|\boldsymbol{P}_i\|}{\|\boldsymbol{P}_b\|}\right) \cdot \hat{\boldsymbol{c}}, \tag{6}$$

where the maximum value F_{max} can be tuned, and $\boldsymbol{P}_b$ is the position vector when the swimmer is near the manipulator boundary. Because the length of the water tank we used in this study is 150 mm, the maximum range from the center of the manipulator to the boundary is 75 mm. We assume $\boldsymbol{P}_b$ = [70, 70, 70]. The acceleration $\boldsymbol{a}$ is computed as

$$\boldsymbol{a} = (1/m) \cdot (\boldsymbol{F}_r + \boldsymbol{F}_d + \boldsymbol{F}_\ell + \boldsymbol{F}_{\text{vs}} + \boldsymbol{F}_c), \tag{7}$$

where the mass of the swimmer m used in this study is 12.4 mg. The estimated velocity $\hat{\boldsymbol{V}}_i$ and estimated position $\hat{\boldsymbol{P}}_i$ can be computed by

$$\hat{\boldsymbol{V}}_i = \boldsymbol{V}_i + \boldsymbol{a} \cdot \Delta t \tag{8}$$

$$\hat{\boldsymbol{P}}_i = \boldsymbol{P}_i + \boldsymbol{V}_i \cdot \Delta t + \frac{1}{2} \cdot \boldsymbol{a} \cdot \Delta t^2. \tag{9}$$

B. Sparse Identification of Nonlinear Dynamics (SINDY)

The sparse identification of nonlinear dynamics (SINDY) was first introduced in [35], and demonstrated with the chaotic Lorenz system. Then, the extension of the SINDY algorithm, including external inputs and control, was proposed in [36] and illustrated the relationship of SINDY with and without control. The SINDY algorithm is a fully nonlinear dynamic system from measured data. It exploits the fact that many dynamic systems have relatively few terms that dominate the system characteristics in the right-hand side of the governing equations.

$$\frac{d}{dt}X = f(X) \tag{10}$$

By defining a library of candidate nonlinear functions,

$$\theta(X) = \begin{bmatrix} 1 & X & X^2 & X^3 & \ldots & \sin(X) & \sin(2X) & \ldots \end{bmatrix}, \quad (11)$$

where X is the states of the system, and the nonlinear candidates are the combinations of the states from order 0 to user defined highest order. Then the dynamic system can be written as:

$$\dot{X} = \xi \theta^T(X), \quad (12)$$

where T is the transpose. The coefficient ξ in the above library are sparse for most cases. Therefore, a sparse regression was implemented to identify the corresponding sparse ξ with the fewest nonlinearities that can still estimate system. Then the SINDY method is extended with inputs and control u, and (10) is rewritten as:

$$\frac{d}{dt} X = f(X, u) \quad (13)$$

This extension is readily generalized to include control, which requires building a larger library $\theta(X, u)$. Moreover, these functions can include nonlinear cross-terms of X and u. Then we can solve for the sparse coefficients ξ with the following:

$$\dot{X} = \xi \theta^T(X, u). \quad (14)$$

The SINDY algorithm was programmed in MATLAB. The nonlinear SINDY model was trained with the highest order of 2 without sinusoidal functions, and the linear SINDY model was trained with the highest order of 1 without sinusoidal functions.

C. Multi-layer perceptron (MLP)

Multi-layer perceptron (MLP) is a class of feedforward supervised artificial neural networks. Chen et, al. [37] demonstrated that feedforward multi-layered neural networks offer a competitive alternative for modeling complex nonlinear systems. Other multi-layer perceptron system identification studies are presented in [38]–[40]. An MLP consists of at least one input layer, one hidden layer, and one output layer, and each layer consists of many perceptrons. Except for the input layer, each layer can be followed by a nonlinear activation function, and the MLP utilizes a supervised learning technique for training which is called back propagation [41], [42]. In this paper, the MLP neural network structure was built using Keras [43] and programmed in Jupyter. The structure of the MLP is shown in Table I. The input features are current position P_x, P_y, P_z, current velocity V_x, V_y, V_z, and next recorded control input $\omega_x, \omega_y, \omega_z$, a total of nine input features.

TABLE I

MLP SUMMARY

Layer	dimension	Activation	Param #
1	10	ReLU	100
2	10	ReLU	110
3	6	Linear	66
Total parameters: 276			

D. Model predictive control (MPC)

Model predictive control is a widely used control algorithm in robotics that incorporates state and control constraints [44]. The MPC solves an optimal control problem over a horizon time subject to constraints, and the first control action will be implemented as the following control input. For a linear system, the optimization can be solved efficiently with convex optimization. However, for nonlinear systems, often parameters require manual tuning. Although the MPC has achieved the desired performance in simulation, it can cause suboptimal performance and even instability if the system model is inaccurate. In this study, the optimal solution subjected to system constraints is solved by the MATLAB function `fmincon`. The optimization objective function is defined as:

$$\begin{aligned} \min J(x_t) = \min \sum_{k=1}^{m_p} [(\hat{X}_{t+k} - X_r) Q (\hat{X}_{t+k} - X_r)^T + \\ (u_{t+k} - u_{t+k-1}) R (u_{t+k} - u_{t+k-1})^T], \end{aligned} \quad (15)$$

where m_p is the receding window size, $\hat{X}_{t+k}$ is the predicted state at time step $t + k$, and X_r is the corresponding reference. This objective function is defined with positive definite matrices Q to minimize the tracking error and R to minimize control effort.

III. EXPERIMENTAL SETUP AND RESULTS

The experimental study presented in this paper include three parts: (1) data-driven control model training, (2) model accuracy comparison, and (3) model predictive control implementation.

A. Hardware description

The hardware main design and control of the magnetic manipulator in this study are described in detail in our previous study [1]–[4]. However, the power supply mode was switched from current mode to voltage mode by adding a series of capacitors to improve the power factor and resonating frequency for blood clot removal as demonstrated in [45]. The magnetic manipulator has six electromagnet coils arranged on the faces of a cube with a 300 mm edge length. Each EM coil has 795 turns, an internal radius of 180 mm and an external radius of 215 mm. Each EM is powered by a set of two **Kepco BOP 20-50MG** power supplies connected in series. Each power supply can provide up to 2000 W with a maximum of 100 V. They can either work in voltage or current mode. An industrial controller **National Instrument IC3173** was used to control the power supplies with an external analog signal. Two Basler acA800 cameras set on the right and top sides of the manipulator are used to measure the swimmer's position during 3D path-following. The industrial controller performs computations for image processing and real-time control. The diameter of the swimmer we used in this study is 2.5 mm, and the length is 6 mm, which was experimentally optimized in [1]. The controllers used for 3D path-following were implemented in LabVIEW.

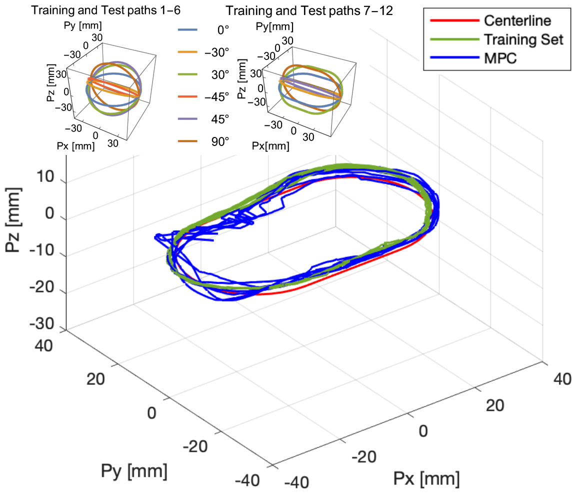


Fig. 3. Plot of experimental result of following a racetrack path using camera feedback at 20 ms (without using an estimator). The racetrack path has 30 mm straightaways and 40 mm radius curves. The average tracking error for the training set was 2.68±1.98 mm and for the MPC was 2.7±2.2 mm.

B. Model training

TABLE II
TRAINING TIME AND NUMBER OF PARAMETERS

Model Name	Physical	$\text{SINDY}_{\text{Non}}$	$\text{SINDY}_{\text{Linear}}$	MLP
Time[s]	1696	20.64	2.10	2760
Number of Params.	6	330	60	276

To train the model, we recorded experimental results while following a racetrack path and a circular path that is rotated about the y-axis at six angles (0°, -30°, 30°, -45°, 45°, 90°) as the training set, which are shown as the small plots in the Fig.3. As the validation set a circular path that rotated continuously at 0.5 Hz, which is shown as the schematic on the right side of Table III. The radius of all circular path is 35 mm. The recorded experimental data included the time of measurements $t[ms]$, the rotational vector $\boldsymbol{\omega}[rad/s]$ that defines the swimmer's rotational speed and orientation, and the swimmer's position $\boldsymbol{P}[m]$ measured from the camera. For the training data, the unit of time is converted from $[ms]$ to $[s]$, the unit of rotation vector is converted from $[rad/s]$ to $[Hz]$, and the unit of position is converted from $[m]$ to $[mm]$. In addition, we set the period to 20 ms for both camera and controller and recorded 14,500 experimental data points for each case.

1) Physical estimation model: We used the `fminsearch` function from MATLAB's optimization toolbox to find parameters (k, C_{d0}, C_{d1}, $C_{\ell 0}$, F_{vs}, and F_{max}). The objective function Eq. (16) is defined as the average error between estimated position and the measured position.

$$J_{\text{Physical}} = 1/N \sum_{i=1}^{N} (\tilde{P}_i - \bar{P}_i), \tag{16}$$

where $\tilde{P}_i$ is estimated position, $\bar{P}_i$ is measured position, and N is the total number of data points. Finally the `fminsearch` function returns the optimized parameters that minimizes average error of the function. The parameters found are k = 0.00209 N·s/rad, C_{d0} = 0.00021 N·s/mm, C_{d1} = 0.00127 N·s/mm, $C_{\ell 0}$ = -0.00017 N·s/mm, F_{vs} = -0.01519 N, and F_{max} = -0.00013 N.

2) SINDY: The state X is a vector containing position and velocity. According to to [35], [36], the recorded experimental data can be rearranged as:

$$X = [X_1,\ X_2,\ X_3,\ \ldots,\ X_{m-1}], \tag{17}$$

and

$$X' = [X_2,\ X_3,\ X_4,\ \ldots,\ X_m]. \tag{18}$$

Then the dynamic system can be expressed with (14), where $\dot{X}$ can be replaced by X'.

3) MLP: The MLP neural network is programmed with Keras API. The loss of the neural network is defined as the mean squared error (MSE), and the optimizer is *adam* [46]. The training dataset was split with 80% as the training set and 20% as the test set, and the dataset was normalized using *Min Max Scaler*. An external Python script communicates with the LabVIEW controller using TCP/IP to implement the MLP model. Because the Keras predict function requires more time to check parameters than 20 ms, the weights of the MLP model were attracted from the trained model and the activation functions were implemented in a customized neural network that performs the computation between each layer.

All models are trained using the same amount of data which is a total of 174,000 data points from 12 cases (6 circle trajectories and 6 racetrack trajectories). The time consumed for training is shown in Table II. The SINDY linear model only took about 2.1 s because The SINDY solver converged faster than `fminsearch` or MLP backpropagation. The time required by the SINDY solver is a function of the number of parameters, so the linear model converged the fastest. The MLP model spent the most of time which is 2.760 s and performed 300 training epochs with a batch size of 10. The difference between experimental data and predictions gained from models are presented in Table III. The error norm of training and test results from highest to lowest is SINDY Linear, Physical, MLP, and SINDY Nonlinear. Due to the simplicity of the path, we further verified the model accuracy using a rotating circular path. Compared to the rotating circle and racetrack paths, the swimmer must deal with a continuously changing reference frame. The result is shown in the **Validation Results** in Table III.

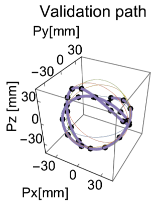
C. Model accuracy comparison

Next, the real-time prediction studies were performed. There were two studies conducted to verify the accuracy of the models. The first is to predict the position of the swimmer with different feedback rates. Because the model is trained with experimental data recorded every 20 ms, the model iteratively performed prediction of the swimmer states until there was a position measurement captured from

TABLE III

TRAINING SUMMARY (MINIMUM ERROR VALUE IN PINK)

Model Name	Training Results [mm]				Test Results [mm]				Validation Results [mm]			
	X	Y	Z	Norm	X	Y	Z	Norm	X	Y	Z	Norm
Physical	0.30	0.30	0.17	0.54	0.40	0.40	0.17	0.68	0.21	0.23	0.19	0.45
SINDY$_{\text{Non}}$	0.20	0.25	0.18	0.45	0.20	0.25	0.18	0.45	0.34	0.35	0.33	0.70
SINDY$_{\text{Linear}}$	0.46	0.46	0.29	0.83	0.47	0.48	0.29	0.86	0.57	0.65	0.34	1.03
MLP	0.32	0.36	0.19	0.52	0.35	0.39	0.19	0.56	0.34	0.46	0.25	0.63



the camera. We recorded the data with measurement rates ranging from $20\,\mathrm{ms}$ to $100\,\mathrm{ms}$ at a $20\,\mathrm{ms}$ interval. Figure **??** shows the experimental result of the swimmer following a 0° racetrack path without models. The results shown in Fig. 4 are the error between prediction and measured position at the time of update, and the physical model has the lowest error. The second study was to predict the position of the swimmer tracking a continuously rotating circle rotating about the y-axis. The results are shown in Table IV, and the tracking trajectories of the physical model is plotted in Fig. 5. The plots of the other models are similar, so they are not presented. As the result shows, all four models can predict the motion of the swimmer. The green curve in the right bottom plot is the result after removing outliers. Outliers are caused by image processing errors due to the reflection of the water and acrylic tank or lost detections, which are the spikes in the blue plot. For the rotating circle the error without outliers of physical, SINDY nonlinear, SINDY linear, and MLP models are 0.5 ± 0.65 mm, 0.88 ± 0.56 mm, 0.89 ± 0.53 mm, and 1.02 ± 0.7 mm.

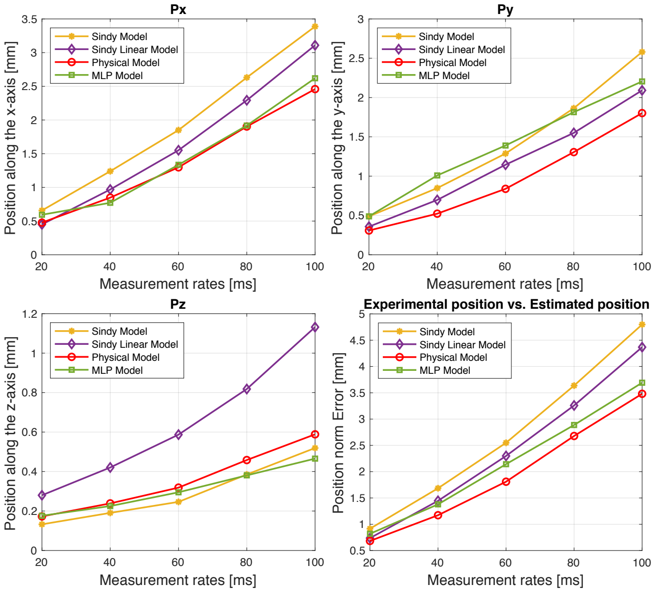


Fig. 4. 0° racetrack path tracking results with 4 models.

D. *Kalman filter with different feedback rates*

As demonstrated in the previous section, the models can predict the swimmer's position even with gaps between sensor measurements. This section demonstrates that the model can be implemented using a multi-step Kalman filter, which can significantly improve the control performance. The procedure is the same as in the previous section. Suppose there is a new measurement from the camera. In that case, we use the new measured swimmer's position and current control input as the input to the model, then predict the estimated velocity and position for the following position. If no measurements is received, the state of the swimmer is iteratively computed until there is an update. As shown in Fig. 6, using the same PI controller, the swimmer can be controlled with a maximum of $180\,\mathrm{ms}$ between measurements, and the tracking error is about $9\,\mathrm{mm}$. However, after the multi-step Kalman filter was employed, the maximum time between measurements was increased to $420\,\mathrm{ms}$, and the maximum tracking error is below $4.6\,\mathrm{mm}$ for all four models.

TABLE IV

REAL-TIME PREDICTION OF ROTATING CIRCLE AT 20 ms FEEDBACK RATE. ALL ERRORS ARE IN MM.

Model Name	Error$_X$	Error$_y$	Error$_z$	RMSE
Physical	0.24	0.34	0.31	0.62
SINDY$_N$	0.42	0.46	0.52	0.94
SINDY$_L$	0.48	0.54	0.40	0.95
MLP	0.69	0.55	0.42	1.02

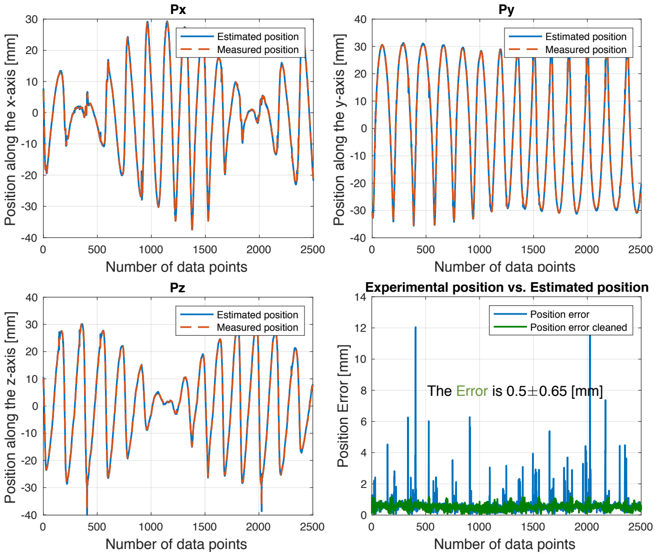


Fig. 5. Rotating circular path tracking results using the *Physical* model.

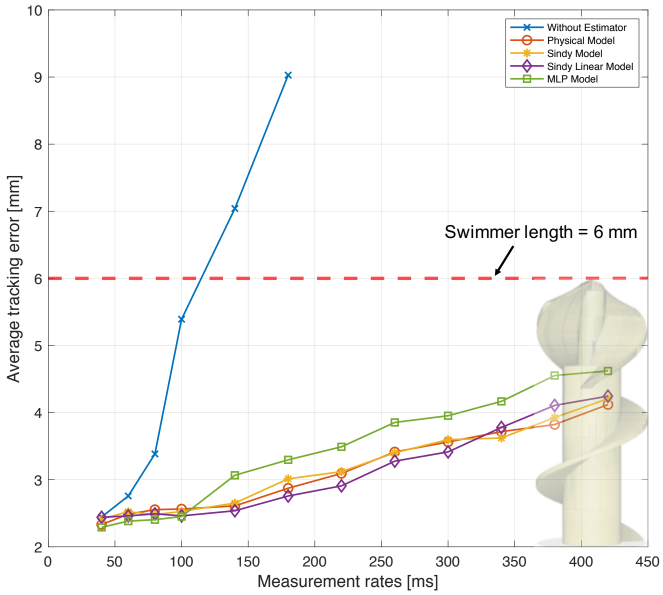


Fig. 6. Comparison of multi-step Kalman filter with latency feedback from $40\,\mathrm{ms}$ to $420\,\mathrm{ms}$.

TABLE V

MPC PARAMETERS

Variable	Definition	Value
N	Time window horizon	5
Q	State weights	[3 3 3]
R	Δ Input weight	[1 1 1]
Ru	Input weight	[1 1 1]
LB	Low bound of control	[-1 -1 -1]
UB	Upper bound of control	[1 1 1]
LBdu	Low bound of Δ control	[-1 -1 -1]
UBdu	Upper bound of Δ control	[1 1 1]
k	Look forward velocity gain	0.001
Lfd	Look forward distance	0.009

E. Model predictive control results

In this study, we implemented the estimated model discussed in the previous section into the MPC controller. The MPC was programmed in MATLAB and communicated with LabVIEW using TCP/IP. The objective function was defined as (15), and the parameters of the MPC controller are listed in Table V.

Due to the $20\,\mathrm{ms}$ control rate, the SINDY linear model was used for MPC computation, and the time window prediction horizon was limited to 5 steps. The `fmincon` optimizer function was set as Sequential Quadratic Programming (SQP), and all other settings to expedite the computation process are shown in Table VI. We validated the estimated model and MPC performance on a fixed racetrack path. The pure-pursuit [47] algorithm was also employed to smooth the tracking performance. The tracking results are plotted as Fig. 3, and the average tracking error is 2.7 ± 2.2 mm. The path-following results on a helix and racetrack combined path are shown as Fig. 7, where the average tracking error was 3.3 ± 1.9 mm. Each MPC computation took 10.7 ± 5.3 ms.

TABLE VI

FMINCON SETTING

Algorithm	SQP
MaxIterations	10
StepTolerance	1.0e-015
MaxFunctionEvaluations	30

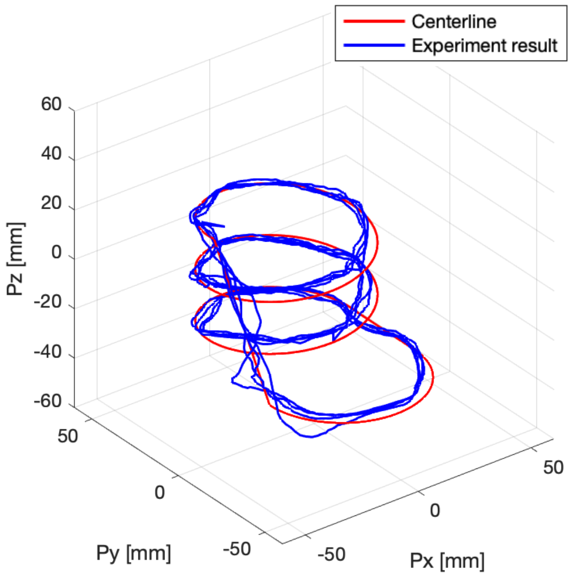


Fig. 7. Experimental result of using MPC following a helix and racetrack combined path with camera feedback at $20\,\mathrm{Hz}$. The average tracking error is 3.3 ± 1.9 mm. Four cycles are shown.

IV. Conclusion

This paper presents data-driven models and control of a milli-scale spiral-type magnetic swimmer using MPC. Four models were developed using only experimental data. The accuracy of the models was experimentally demonstrated. Furthermore, the control performance with large time between sensor measurements was significantly improved by using the model for state estimation and control using a multi-step Kalman filter. With this control the system maintained a tracking error smaller than the swimmer's length. Although the performance of the four models is similar to each other in multi-step Kalman filter studies, this also demonstrated that the physical estimation, SINDY, and MLP could efficiently estimate the system purely using the experimental data. The time consumed by the training process is mainly affected by the algorithm and the amount of data processed, both of which the user and applications can determine. Finally, the path following experiments were conducted with the estimated model and MPC controller, and the tracking error of racetrack is 2.7 ± 2.2 mm, which is about the same as the training data. Together, this paper demonstrates approaches for model building and for deploying controllers that do not rely on hand-tuned parameters.

References

[1] H. Zhao, J. Leclerc, M. Feucht, O. Bailey, and A. T. Becker, "3d path-following using mrac on a millimeter-scale spiral-type magnetic robot," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1564–1571, 2020.

[2] J. Leclerc, H. Zhao, and A. T. Becker, "3d control of rotating millimeter-scale swimmers through obstacles," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8890–8896.

[3] J. Leclerc, H. Zhao, D. Bao, and A. T. Becker, "In vitro design investigation of a rotating helical magnetic swimmer for combined 3-d navigation and blood clot removal," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 975–982, 2020.

[4] J. Leclerc, H. Zhao, D. Z. Bao, A. T. Becker, M. Ghosn, and D. J. Shah, "Agile 3d-navigation of a helical magnetic swimmer," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 7638–7644.

[5] L. Zhang, J. J. Abbott, L. Dong, B. E. Kratochvil, D. Bell, and B. J. Nelson, "Artificial bacterial flagella: Fabrication and magnetic control," *Applied Physics Letters*, vol. 94, no. 6, p. 064107, 2009.

[6] A. Ghosh and P. Fischer, "Controlled propulsion of artificial magnetic nanostructured propellers," *Nano letters*, vol. 9, no. 6, pp. 2243–2245, 2009.

[7] J. Grover, J. Zimmer, T. Dear, M. Travers, H. Choset, and S. D. Kelly, "Geometric motion planning for a three-link swimmer in a three-dimensional low reynolds-number regime," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 6067–6074.

[8] R. Dreyfus, J. Baudry, M. L. Roper, M. Fermigier, H. A. Stone, and J. Bibette, "Microscopic artificial swimmers," *Nature*, vol. 437, no. 7060, pp. 862–865, 2005.

[9] K. B. Yesin, K. Vollmers, and B. J. Nelson, "Modeling and control of untethered biomicrorobots in a fluidic environment using electromagnetic fields," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 527–536, 2006.

[10] K. Kim, J. Guo, Z. Liang, and D. Fan, "Artificial micro/nanomachines for bioapplications: biochemical delivery and diagnostic sensing," *Advanced Functional Materials*, vol. 28, no. 25, p. 1705867, 2018.

[11] O. Ergeneman, G. Dogangil, M. P. Kummer, J. J. Abbott, M. K. Nazeeruddin, and B. J. Nelson, "A magnetically controlled wireless optical oxygen sensor for intraocular measurements," *IEEE Sensors Journal*, vol. 8, no. 1, pp. 29–37, 2008.

[12] A. Chałupniak, E. Morales-Narváez, and A. Merkoçi, "Micro and nanomotors in diagnostics," *Advanced drug delivery reviews*, vol. 95, pp. 104–116, 2015.

[13] F. Qiu, S. Fujita, R. Mhanna, L. Zhang, B. R. Simona, and B. J. Nelson, "Magnetic helical microswimmers functionalized with lipoplexes for targeted gene delivery," *Advanced Functional Materials*, vol. 25, no. 11, pp. 1666–1671, 2015.

[14] C. S. Troisi, M. Knaflitz, E. S. Olivetti, L. Martino, and G. Durin, "Fabrication of new magnetic micro-machines for minimally invasive surgery," *IEEE Transactions on Magnetics*, vol. 44, no. 11, pp. 4488–4491, 2008.

[15] B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," *Annual review of biomedical engineering*, vol. 12, pp. 55–85, 2010.

[16] M. Sendoh, K. Ishiyama, K. I. Arai, M. Jojo, F. Sato, and H. Matsuki, "Fabrication of magnetic micromachine for local hyperthermia," *IEEE transactions on magnetics*, vol. 38, no. 5, pp. 3359–3361, 2002.

[17] F. Sato, M. Jojo, H. Matsuki, T. Sato, M. Sendoh, K. Ishiyama, and K. I. Arai, "The operation of a magnetic micromachine for hyperthermia and its exothermic characteristic," *IEEE transactions on magnetics*, vol. 38, no. 5, pp. 3362–3364, 2002.

[18] P. M. Devlin, *Brachytherapy: applications and techniques*. Springer Publishing Company, 2015.

[19] I. S. Khalil, A. F. Tabak, K. Sadek, D. Mahdy, N. Hamdi, and M. Sitti, "Rubbing against blood clots using helical robots: modeling and in vitro experimental validation," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 927–934, 2017.

[20] E. M. Purcell, "Life at low reynolds number," *American journal of physics*, vol. 45, no. 1, pp. 3–11, 1977.

[21] ——, "The efficiency of propulsion by a rotating flagellum," *Proceedings of the National Academy of Sciences*, vol. 94, no. 21, pp. 11 307–11 311, 1997.

[22] K. E. Peyer, L. Zhang, and B. J. Nelson, "Bio-inspired magnetic swimming microrobots for biomedical applications," *Nanoscale*, vol. 5, no. 4, pp. 1259–1272, 2013.

[23] X. Wang, C. Hu, S. Pane, and B. J. Nelson, "Dynamic modeling of magnetic helical microrobots," *IEEE Robotics and Automation Letters*, 2021.

[24] K. E. Peyer, F. Qiu, L. Zhang, and B. J. Nelson, "Movement of artificial bacterial flagella in heterogeneous viscous environments at the microscale," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2553–2558.

[25] J. J. Abbott, K. E. Peyer, M. C. Lagomarsino, L. Zhang, L. Dong, I. K. Kaliakatsos, and B. J. Nelson, "How should microrobots swim?" *The international journal of Robotics Research*, vol. 28, no. 11-12, pp. 1434–1447, 2009.

[26] X. Wang, C. Hu, L. Schurz, C. De Marco, X. Chen, S. Pané, and B. J. Nelson, "Surface-chemistry-mediated control of individual magnetic helical microswimmers in a swarm," *ACS nano*, vol. 12, no. 6, pp. 6210–6217, 2018.

[27] H. Li, J. Tan, and M. Zhang, "Dynamics modeling and analysis of a swimming microrobot for controlled drug delivery," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 2, pp. 220–227, 2008.

[28] A. Acemoglu and S. Yesilyurt, "Effects of geometric parameters on swimming of micro organisms with single helical flagellum in circular channels," *Biophysical journal*, vol. 106, no. 7, pp. 1537–1547, 2014.

[29] C. Ye, J. Liu, X. Wu, B. Wang, L. Zhang, Y. Zheng, and T. Xu, "Hydrophobicity influence on swimming performance of magnetically driven miniature helical swimmers," *Micromachines*, vol. 10, no. 3, p. 175, 2019.

[30] K. Ishiyama, M. Sendoh, A. Yamazaki, and K. Arai, "Swimming micro-machine driven by magnetic torque," *Sensors and Actuators A: Physical*, vol. 91, no. 1-2, pp. 141–144, 2001.

[31] M. Sendoh, N. Ajiro, K. Ishiyama, M. Inoue, K. Arai, T. Hayase, and J. Akedo, "Effect of machine shape on swimming properties of the spiral-type magnetic micro-machine," *IEEE Transactions on Magnetics*, vol. 35, no. 5, pp. 3688–3690, 1999.

[32] A. Yamazaki, M. Sendoh, K. Ishiyama, T. Hayase, and K. Arai, "Three-dimensional analysis of swimming properties of a spiral-type magnetic micro-machine," *Sensors and Actuators A: Physical*, vol. 105, no. 1, pp. 103–108, 2003.

[33] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of guidance, control, and dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[34] J. Leclerc, B. Isichei, and A. T. Becker, "A magnetic manipulator cooled with liquid nitrogen," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4367–4374, 2018.

[35] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[36] ——, "Sparse identification of nonlinear dynamics with control (sindyc)," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 710–715, 2016.

[37] S. Chen, S. A. Billings, and P. Grant, "Non-linear system identification using neural networks," *International journal of control*, vol. 51, no. 6, pp. 1191–1214, 1990.

[38] X. Li and W. Yu, "Dynamic system identification via recurrent multilayer perceptrons," *Information sciences*, vol. 147, no. 1-4, pp. 45–63, 2002.

[39] J. H. Pérez-Cruz, A. Y. Alanis, J. d. J. Rubio, and J. Pacheco, "System identification using multilayer differential neural networks: a new result," *Journal of Applied Mathematics*, vol. 2012, 2012.

[40] A. Parlos, A. Atiya, K. Chong, W. Tsai, and B. Fernandez, "Recurrent multilayer perceptron for nonlinear system identification," in *IJCNN-91-Seattle International Joint Conference on Neural Networks*, vol. 2. IEEE, 1991, pp. 537–540.

[41] F. Rosenblatt, "Principles of neurodynamics. perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.

[42] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[43] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[44] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[45] J. Leclerc, Y. Lu, A. T. Becker, M. Ghosn, and D. J. Shah, "Resonating magnetic manipulation for 3d path-following and blood clot removal using a rotating swimmer," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 3083–3090.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[47] R. C. Coulter, “Implementation of the pure pursuit path tracking algorithm,” Carnegie-Mellon UNIV Pittsburgh PA Robotics INST, Tech. Rep., 1992.