Motion Primitives for Path Following with a Self-Assembled Robotic Swimmer

Carlos Orduño, Aaron Becker, and Timothy Bretl

***Abstract*— This paper presents a control strategy based on model learning for a self-assembled robotic "swimmer". The swimmer forms when a liquid suspension of ferro-magnetic micro-particles and a non-magnetic bead are exposed to an alternating magnetic field that is oriented perpendicular to the liquid surface. It can be steered by modulating the frequency of the alternating field. We model the swimmer as a unicycle and learn a mapping from frequency to forward speed and turning rate using locally-weighted projection regression. We apply iterative linear quadratic regulation with a receding horizon to track motion primitives that could be used for path following. Hardware experiments validate our approach.**

I. Introduction

Recent work has shown that a suspension of ferro-magnetic micro-particles (45μm) will form a large-scale structure when exposed to an alternating magnetic field that is oriented perpendicular to the liquid surface [1]–[3]. This structure becomes mobile and turns into a self-assembled robotic "swimmer" when symmetry is broken by adding a larger non-magnetic bead (1mm). The motion of this swimmer is unidirectional, generally toward the bead that forms the head. Changes in frequency of the alternating magnetic field (30-40Hz) cause changes in the arrangement and characteristic length of the swimmer, which in turn change the surrounding flow and affect subsequent motion.

Previous work has focused on understanding the physics that govern these phenomena. In this paper we focus on developing a control strategy that allows us to steer the resulting swimmer along primitives of canonical shape (e.g., circles of different radius) that could be used to follow arbitrary paths.

We model the self-assembled robotic swimmer as a non-holonomic unicycle, under the constraint that it only moves forward. We apply locally weighted projection regression (e.g., see [14]) to learn the mapping from applied frequency to forward speed and turning rate given data collected offline. We apply iterative linear quadratic regulation with a receding horizon (e.g., see [18]) to track motion primitives given visual feedback. We validate our approach in hardware experiments with the system shown in Figure 1.

Our work is motivated by applications that include targeted drug delivery and non-invasive surgery. The self-assembled robotic system that we describe here (and, in particular, the visual feedback that we currently require) is clearly not appropriate for these applications yet. We view this system as a platform for the development of new control strategies and for the exploration of new mechanisms for self-assembly, which we hope may build a foundation for future systems with more practical application.

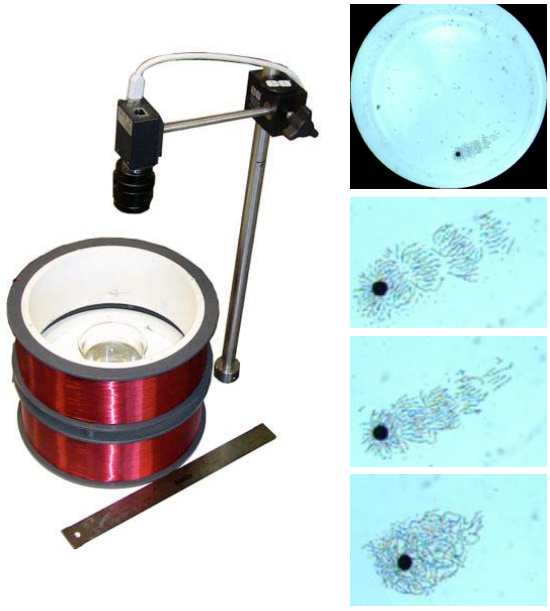

Fig. 1. Our hardware platform (left) and the resulting self-assembled robotic "swimmer" (right), which arises when an alternating magnetic field is applied perpendicular to an air-water interface that contains a suspension of ferro-magnetic micro-particles and a non-magnetic bead. The structure of the swimmer changes with the frequency of the alternating field—shown is the same swimmer at a frequency of 28, 40, and 77 Hz.

Our paper proceeds as follows. Section II gives a brief overview of related work, focusing on methods of self-assembly, of magnetic control of micro-robots. Section III describes both the self-assembled robotic swimmer and our hardware implementation in more detail. Section IV presents our approach to model learning and validation, which is based on the use of locally weighted projection regression. Section V presents our approach to control, which is based on the use of iterative linear quadratic regulation with a receding horizon. Section VI shows our experimental results. Section VII concludes with a brief discussion of opportunities for future work.

II. Related Work

The swimmers covered by this work exhibit self-assembly. Their size and actuation mechanisms place them in the

C. Orduño is with the Department of Mechanical Science and Engioneering, A. Becker is with the Department of Electrical and Computer Engineering and T. Bretl is with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA {orduno1,abecker5,tbretl}@illinois.edu

class of magnetically controlled micro robots. Finally, their dynamics are dependent on their structure, which is randomly generated when they are formed. This uncertainty requires model learning before we can control them.

A. Self-assembly

Magnetic swimmers fit into a broader category of self-assembled systems, an overview of which is given in [5]. In that work by Whitesides and Grzybowski [3], self-assembly was defined as *a reversible process by which pre-existing discrete entities bind to each other without being directed externally*. The magnetic swimmers form spontaneously under appropriate particle loading and magnetic frequency. The process is reversible in that the swimmer components quickly separate out when the driving frequency is removed.

Self-assembly is of considerable interest to the robotics community in community, from applications such as self-arrangement in parts-handling [6] to task-oriented self-assembly of modular robots [7]. Our swimmers have no autonomy, so only their formation can be considered self assembly. With external control, these swimmers can be directed to follow paths, impact their environment, or grow into larger swimmers. The former tasks are related to parts-handling while the latter is similar to modular robotics.

B. Magnetically controlled of micro robots

Magnetically controlled micro robots are an active field of research. Because they involve actuation from a distance they are being pursued for applications in medical devices [8]–[10].

Sudo et al. presented a 5mm magnet with a flexible tail, propelled by an external magnetic field [8], capable of swimming in viscous fluid. This robot was designed to be navigable through the human heart and large arteries. Abbott et al. contrast the efficacy of pulling using a magnetic gradient verse micro robots that flex or use helical propeller to swim through a fluid, and find that swimming micro robots become more desirable as distance from the generating magnetic field increases and as the robot size decreases. The swimmers presented by [9], [10] exhibit variation in velocity, but are constrained to all have the same orientation, so they cannot be steered independently.

Parallels can also be found with the non-swimming micro manipulation work of Diller and Sitti et al. They controlled the 2D coordinates of multiple micro-scale permanent magnets by exploiting heterogeneity in the dimensions of the magnets [11], [12]. In this work a single control signal was applied to each magnet, but unlike the helical swimmers of [9], [10], independent control was possible due to heterogeneity. Similarly, the magnetic swimmers we present have unique dynamic models that are based on their structure. This heterogeneity leads to unique velocity and curvature as a function of magnetic frequency, and could enable independent control.

III. System Description

A. Self-Assembled Magnetic Swimmers

Magnetic particles suspended on the liquid-air interface, subject to an alternating magnetic field, will exhibit a self-assembly behavior [1]. The phenomenon is driven by the collective response to the magnetic field and the generation of surface waves by the oscillating particles.

The self-assembly process starts when in the presence of the alternating field (20 to 120 Hz) the magnetic moments from the particles are aligned along the chain direction driven by dipole-dipole interactions [1]. On a larger scale, the chains order to form segments, which in contrast to the chains, have an antiferromagnetic ordering.

The applied alternating field also causes the particles to oscillate in place dragging the adjacent water. Consequently the segment oscillations create fluid motion at both ends of the snake nearly equal for frequency values below 85 Hz [3]. By placing a glass bead (1-1.5mm in diameter) near one end, it will attach and the liquid displacement will be higher at the opposite tail producing a net forward displacement.

Belkin and Snezhko present in-depth experimental and theoretical studies of the water flows [2] which range from 0.4 cm/s to 2 cm/s.

Self-assembled swimmers present an impressive system for the manipulation of objects on the surface of water. By adjusting the driving signal frequency we are capable of modifying the segment arrangement and characteristic length, process which results on a change of the differential water flow from both sides. As the frequency value is increased the segment sections contract increasing the overall water flow and the difference in flows from both sides. This inhomogeneity opens the opportunity for doing limited swimmer steering by selectively choosing a frequency value to achieve a particular turning rate or swimming curvature.

The feasible curvatures followed by a swimmer depend on the symmetric property of structure. Given a perfectly symmetric swimmer, varying the control frequency will only result on a change of the forward speed with no modification of the turning rate.

Below critical frequency values [15] the swimmer is stable in the short-term, subject only to change when chains detach from their corresponding segment due to collision with the beaker wall, and interactions with the water flow and suspended particles along the path. These changes modify the response of the swimmer to a given frequency.

We select the 30 to 40 Hz frequency range for driving the swimmers since this offers two relevant advantages: we can assume the swimmer configuration is reversible and this range exhibits the largest changes in velocity and turning rate.

B. Hardware Platform

Our hardware platform consists of a cylindrical glass container (145 mm in diameter) inside a Helmholtz coil (220 mm in diameter, 2x 83 mm in height). The coil is energized by an alternating voltage generated by a dedicated

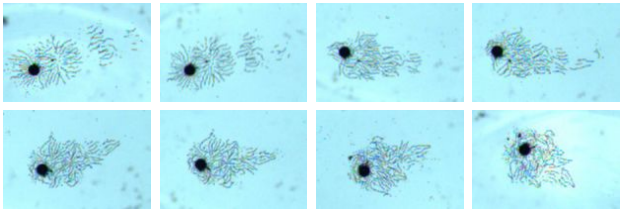

Fig. 2. A self-assembled swimmer shown at magnetic frequency 24, 30,40,50,60,70,80 and 90Hz. The swimmer begins with three well-defined links, but changes shape significantly.

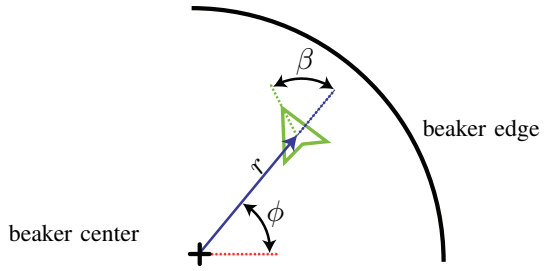

Fig. 4. Coordinate frame (r, β, ϕ) used for magnetic swimmers. The swimmer is shown in green.

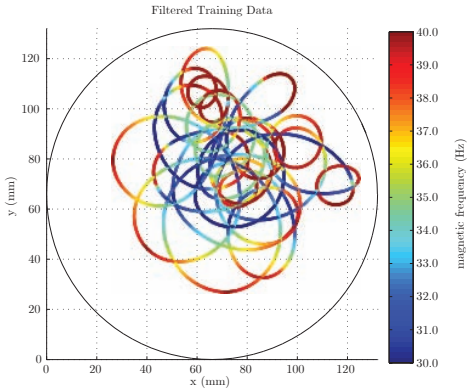

Fig. 5. Typical training data used for learning a model.

frequency modulator (FM) circuit connected to an amplifier. The control system runs on a workstation (Intel Xeon 2.4 GHz). The desired frequency is commanded to the FM circuit via an external DAC box (UEI Power DNA-A0-308-350). The control loop is closed using a digital camera (Basler with Edmund 35mm lens). A backlight is added using a flat illumination screen.

IV. Model Learning

A. Locally Weighted Projection Regression

The structural configuration obtained by swimmer formation is unpredictable. This uncertainty limits our ability to model the swimmer reliably beforehand. We could attempt to obtain a parametric model and set the parameters based on the observed swimmer structure, but this approach would require both complex image processing tasks to obtain important features of the swimmer structure and an expensive high-resolution vision system. Instead we choose to learn an online forward model, where future states can be predicted based on past state observations.

Locally Weighted Projection Regression (LWPR) (see [14]) is an extension of Receptive Field Weighted Regression in [16]. LWPR approximates a nonlinear function by piecewise linear models. The learning process consists of obtaining the number linear models K and characteristic parameters $\mathbf{b}_k$ of the hyperplanes that describe each local model. A crucial step is determining the region of validity, also called Receptive Field (RF), in which the local model can be trusted.

The receptive field of each linear model can be computed from a Gaussian kernel:

$$w_k = exp\left(-\frac{1}{2}\left(\boldsymbol{x}-\mathbf{c}_k\right)^{\top}\mathbf{D}_k\left(\boldsymbol{x}-\mathbf{c}_k\right)\right) \tag{1}$$

where w_k is a weight, $\boldsymbol{x}$ is the query point , $\mathbf{c}_k$ is the center of the k-th linear model, and $\mathbf{D}_k \geq 0$ is a distance metric that determines the size and shape of region of validity of the linear model. Other kernel functions are possible.

In LWPR local models are built continuously in the entire support area of the input data at selected points in input space. The prediction for a query point $\hat{y}$ is then formed as the weighted average of the predictions of the local models whose receptive fields are selected. If K local models are chosen, the prediction is calculated by a weighted average and the regression function can be written as:

$$\hat{y} = \hat{f}\left(\boldsymbol{x}\right) = \frac{\sum_{k=1}^{K} w_k \hat{y}_k}{\sum_{k=1}^{K} w_k} \tag{2}$$

$$\hat{y}_k = b_k^0 + \mathbf{b}_k^{\top}\left(\boldsymbol{x}-\mathbf{c}_k\right) \tag{3}$$

where b_k^0 and $\mathbf{b}_k^{\top}$ denote the offset and slope of the k-th local linear model.

LWPR can adapt to changes of the system dynamics in real-time as new data becomes available. This is done by setting a forgetting factor λ, which is selected to balance between preserving the learned model and adapting to new data.

We use the LWPR algorithm implementation provided by Klanke et al. [17].

B. Swimmer Modeling

The nonlinear dynamical behavior of the swimmer can be described by the difference equation:

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \ldots, K-1 \tag{4}$$

Assuming the system can be approximated to an autonomous model, valid over some time horizon $K-1$, i.e. the duration of the motion primitive, we can rewrite (4) as:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \ldots, K-1 \tag{5}$$

Where, for the magnetic swimmer the state and control input are defined as

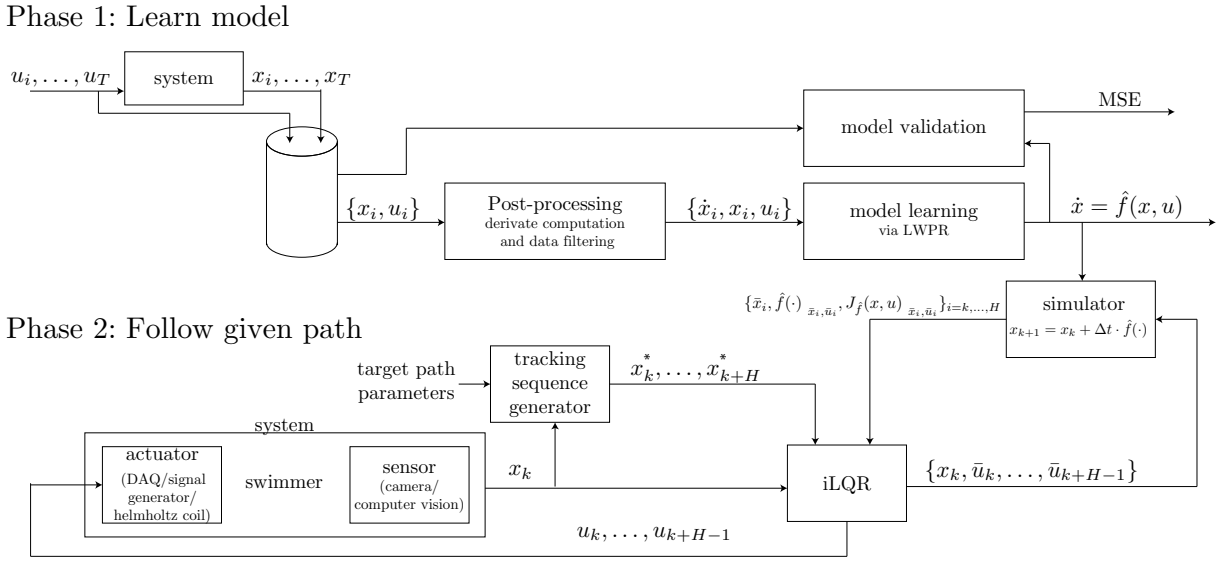

Fig. 3. Block diagram, explained in Sections IV and V

$$\mathbf{x}_k = [r_k,\ \theta_k,\ \phi_k]^\top \quad \mathbf{u_k} = u_k \tag{6}$$

The state components r_k and ϕ_k are the magnitude and angle of the vector pointing from the world center to the swimmer; and θ denotes the angle with respect to this vector. The Helmholtz coil is energized with a sinusoidal signal $A_k \sin(2\pi\psi_k t)$, where A_k is the amplitude of the signal and ψ_k the frequency.

Assuming symmetric dynamics with respect to ϕ, we approximate the general form in (5) to:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \hat{\mathbf{f}}(\pi(\mathbf{x}_k), \mathbf{u}_k) \quad \pi \mapsto r, \beta \tag{7}$$

The function $\hat{\mathbf{f}}$ is learned using LWPR. The general form of $\hat{\mathbf{f}}$ is split into three individual models corresponding to each output variable, where the input-output pairs are computed as:

$$\begin{bmatrix} (r_{k+1} - r_k)/\Delta t \\ (\beta_{k+1} - \beta_k)/\Delta t \\ (\phi_{k+1} - \phi_k)/\Delta t \end{bmatrix} = \begin{bmatrix} M_{\hat{r}}(r, \beta, u) \\ M_{\hat{\beta}}(r, \beta, u) \\ M_{\hat{\phi}}(r, \beta, u) \end{bmatrix} \tag{8}$$

We start the learning process by collecting real-time data. The system is driven for a finite time by a repeating ramp input from $30Hz$ to $40Hz$ for $240s$, resulting state measurements are collected. The time history of states is then smoothed using a low-pass Butterworth filter taking advantage of the time dependance.

As described previously, our objective is to learn a model that, given the state $\mathbf{x}_k$ and the control input $\mathbf{u}_k$ at time $\Delta t \cdot k$, would give a good estimate of the resulting state $\mathbf{x}_{k+1}$

The result is not an analytical model of the swimmer, but rather an algorithm that answers a query i.e. a state, by weighting a series of linear models learned beforehand and returning an approximated state derivative.

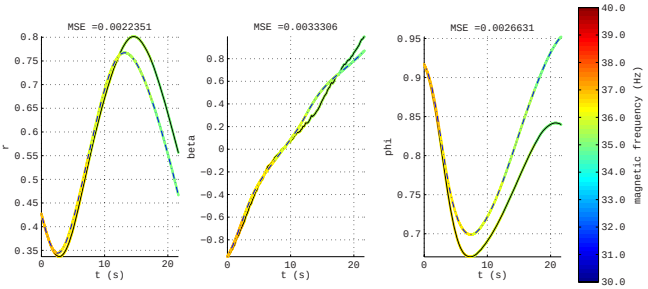

Fig. 6. Predicted (dash) vs. actual (line) paths as a function of time. The simulator prediction has low error up to 5s. This value is used for calculating the controller's time horizon.

The result is a nonlinear estimation for (7) which serves as a forward simulator to our system dynamics, useful for the design of the LQR controller as we will see in Section V.

V. Control Strategy

Our objective is to obtain a control law $u^*_k = \pi_k(\mathbf{x}_k)$ for the task of following circular paths with support of the non-linear dynamical model learned in Section IV. By defining our problem on an optimal control setting, we may penalize deviation from the desired state $\mathbf{x}^*_k$ and control effort $\mathbf{u}^*_k$.

$$\begin{aligned} \underset{u_k}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=0}^{K}(\delta\mathbf{x}_k^\top \mathbf{Q}\delta\mathbf{x}_k + \delta\mathbf{u}_k^\top \mathbf{R}\delta\mathbf{u}_k) \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \end{aligned} \tag{9}$$

where $\delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}^*_k$, $\delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}^*_k$ and $\mathbf{Q} \geq 0, \mathbf{R} > 0$ are the state cost and input cost matrices.

A. Iterative LQR

Iterative LQR (iLQR) [18], also known as Gauss-Newton LQR and Sequential LQR, is a method for solving optimal

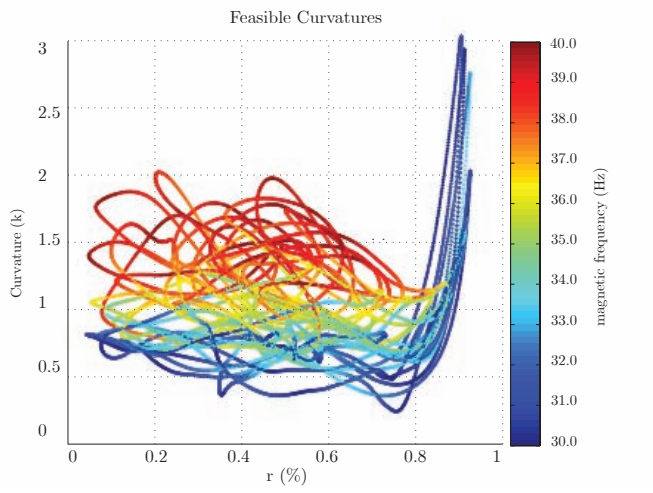

Fig. 7. Curvature as a function of r and u. Each swimmer has a unique response to the input u, and have maximum and minimum curvatures which constrain feasible paths.

control problems with non-linear cost functions and dynamical model:

$$\begin{aligned} \underset{\mathbf{u}_k}{\text{minimize}} \quad & J_k = \sum_{k=0}^{K} \mathbf{l}_k(\mathbf{x}_k, \mathbf{u}_k) \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \end{aligned} \tag{10}$$

The method begins with an initial control guess $\bar{\mathbf{U}}_k^{(0)} = \{\bar{\mathbf{u}}_0^{(0)}, \bar{\mathbf{u}}_1^{(0)}, \ldots, \bar{\mathbf{u}}_K^{(0)}\}$, and the corresponding nominal state sequence $\bar{\mathbf{X}}_k^{(0)} = \{\bar{\mathbf{x}}_0^{(0)}, \bar{\mathbf{x}}_1^{(0)}, \ldots, \bar{\mathbf{x}}_K^{(0)}\}$ obtained from (10), and continues by computing a linear approximation of the dynamics and a quadratic one for the cost around the nominal trajectory $\bar{\mathbf{X}}_k^{(i)}$. Essentially iLQR transforms our non-linear optimal control problem to a linear time-varying one.

$$\begin{aligned} \underset{\mathbf{u}_k}{\text{minimize}} \quad & J = \sum_{k=0}^{K} (\mathbf{x}_k^\top \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R} \mathbf{u}_k) \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \end{aligned} \tag{11}$$

Solving the LQ formulation will provide an improved control sequence $\bar{\mathbf{U}}_k^{(i)}$ and corresponding state trajectory $\bar{\mathbf{X}}_k^{(i)}$ (for more details on linear quadratic methods the authors suggest [19]) We iterate over the last state trajectory or exit if $J^{(i)} < tol \cdot J^{(i-1)}$.

It is relevant to highlight that $\mathbf{x}_k^*$, $\mathbf{u}_k^*$ need not be feasible for this method to work.

B. Tracking Motion Primitives

To solve (10), we apply the method described in Section V-A with the following extensions and modifications:

- To improve convergence to local optimum we gradually shift the dynamics between iterations from being close to the desired trajectory to the actual system dynamics:

$$\mathbf{x}_{k+1} = \beta \mathbf{x}_k^* + (1-\beta)\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$$

  In practice, computational speed restricts the number of iterations we can do during realtime control, but experimentally, three iterations with $\beta = 0.7, 0.1, 0.05$ suffices for convergence.
- The LQ approximation of the dynamics and cost are expressed in terms of $\delta\mathbf{x}_k$ driving the error to zero, and extended to penalize an immediate change in control input:

$$\begin{aligned} \underset{v_k}{\text{minimize}} \quad & J = \frac{1}{2}\sum_{k=0}^{K} \mathbf{z}_k^\top \mathbf{Q} \mathbf{z}_k + \mathbf{v}_k^\top \mathbf{R} \mathbf{v}_k \\ \text{subject to} \quad & \mathbf{z}_{k+1} = \bar{\mathbf{A}}_k \mathbf{z}_k + \bar{\mathbf{B}}_k \mathbf{v}_k \end{aligned}$$

  where:

$$\mathbf{z}_k = [\delta\mathbf{x}_k,\ \delta\mathbf{u}_k]^\top \qquad \mathbf{v}_k = \delta\mathbf{u}_k - \delta\mathbf{u}_{k-1}$$
$$\bar{\mathbf{A}}_k = \begin{bmatrix} \mathbf{A}_k & \mathbf{B}_k \\ 0 & I \end{bmatrix} \qquad \bar{\mathbf{B}}_k = \begin{bmatrix} \mathbf{B}_k \\ I \end{bmatrix}$$

- The initial guess $\mathbf{u}_0$ and target control inputs $\mathbf{u}^*$ are approximated using curvature information from the model learning data.

Computing a feedback law for the duration of the complete motion primitive (from 7s to 30s) has an important limitation, since our simulator performance quickly starts degrading after 5 s. We solve this problem by computing multiple times along the trajectory, similar to using a receding horizon, running the controller for a 1 s long trajectory and recomputing at the end.

The swimmer is moving forward while a magnetic field is applied. Removing the field will cause the structure to rearrange and the dynamical behavior to change from the previous one. This is relevant since on our implementation, the time K^* it takes to compute the optimal control sequence is approximately 10% of its duration, during which we are not capable of doing feedback control, resulting on the initial conditions being off from the ones used on the beginning of the computation. The issue is solved by propagating the current state K^* steps on the simulator using a nominal control input and feeding the resulting state as initial condition.

By using this methodology, we are interested in tracking motion primitives in the form of circular paths. Starting with a particular circle, the user specifies the radius r^* and center location c_x, c_y, and the algorithm accomplishes the task by projecting the current point to the target path (closest point to the circle) and obtaining K target states $\mathbf{x}_k^*$. The iLQR controller is then computed as presented.

VI. Hardware Experiments

A. Experimental Procedure

For an experiment trial, <0.1g of 45μm nickel spheres are deposited over the water filled container (27.5mm water height) using a sieve (200-mesh) while no magnetic field is applied. We generate a sinusoidal magnetic field at $30Hz$ and allow the magnetic moment of the micro-particles to orient for the formation of chains followed by anti-ferromagnetically oriented segments, process which has been described in more detail on III-A. Once a stable swimmer has

been formed we linearly increase and decrease the frequency from $30Hz$ to $40Hz$ on $10s$ to allow further settlement of the segments. At this point the swimmer is symmetric and the water flow on both tails is balanced, resulting on a static structure. To induce an asymmetric structure we manually place a glass bead (1.5mm diameter Ni-Pd-Ni), the swimmer's head, close to one of the tails and position it slightly offset from the centerline. Offsetting the head to the right will produce a counterclockwise swimmer, while placing it to the left will produce a clockwise one. Further manipulation with a permanent magnet may be required to increase the segment particle density. Once a reliable swimmer has been formed we proceed to the model learning phase. We apply a triangular wave ranging from $\min$ to $\max$ frequencies, and record a minimum of $20,000$ swimmer states at $60Hz$. After completion, the time data is fed to our model learning algorithm which outputs the number of receptive fields K, curvature vs r, u and β plots and the MSE obtained with respect to the training data to ensure a similar performance to the values obtained during the tuning phase of LWPR. At this point the user may specify a target circle on the user interface and start the control process. The data stream during this process is recorded for later analysis.

B. Results

We conducted a series of experiments to assess the performance of our model learning algorithm and controller, and present two example swimmers following a circular motion primitive. We describe the results obtained for the model learning and motion primitive tracking phases.

1) *Model Learning:* The table I summarizes the performance of the models learned for both swimmers $S1$ and $S2$. As expected the nMSE obtained from the training data are lower that those for the test data, further more, the cross validations ($S1$ model vs $S2$ data, $S2$ model vs $S1$ data) present significant errors, up to 3x greater, than those from the test data. This shows the importance of learning a new model for each swimmer. As seen on Fig.6 the simulator is capable of predicting the spatial trajectory of the swimmer with low error up to a horizon of 5 s, giving us an important parameter for the design of the controller.
2) *Motion Primitive Tracking:* Our controller was able to reliably follow circle paths over multiple translations over a fixed range of curvatures. Figure 9 shows two example experiments for different swimmers and receding horizons.

VII. Conclusion and Future Work

In this paper we examined the self-assembled magnetic structures introduced by Snezho and Belkin [1]–[3]. These structures are of interest to the robotics community because they can be replicated with minimal hardware, are under-actuated yet controllable, and require model learning.

We presented a strategy using locally-weighted projection regression to learn dynamic models for self-assembled "swimmers" and a control policy based on iterative LQR

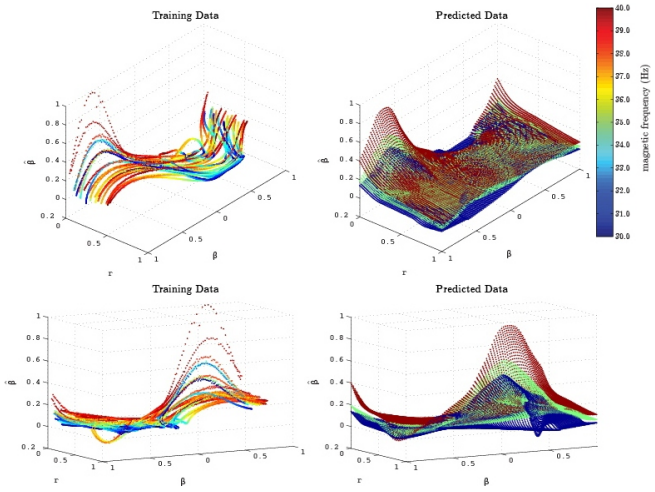

Fig. 8. Training data and learned model $M_{\hat{\beta}}(r, \beta, u) \mapsto \hat{\beta}$. Similar models are created for $M_{\hat{r}}(r, \beta, u)$, and $M_{\hat{\phi}}(r, \beta, u)$.

			Model	
	$\hat{r}$		S_1	S_2
Data	S_1	train	4.52e-03	4.95e-02
		test	1.62e-02	3.90e-02
	S_2	train	1.04e-01	1.18e-02
		test	1.26e-01	3.98e-02
	$\hat{\beta}$		S_1	S_2
	S_1	train	4.75e-02	1.35e+00
		test	2.94e-01	1.15e+00
	S_2	train	4.68e-01	4.83e-02
		test	4.02e-01	8.51e-01
	$\hat{\phi}$		S_1	S_2
	S_1	train	3.31e-02	6.14e-01
		test	2.96e-01	5.16e-01
	S_2	train	3.77e-01	2.19e-02
		test	1.38e-01	1.15e-01

TABLE I

The table shows the normalized Mean Square Error (nMSE) between actual data and predicted values from LWPR for two swimmers, S_1 and S_2. Data is separated into a training set (80%) and a testing set (20%) a total of $26,490$ points where used for $S1$ and $20,791$ for S_2. The nMSE is computed for the training and test sets for each swimmer. Crossed cases are also considered.

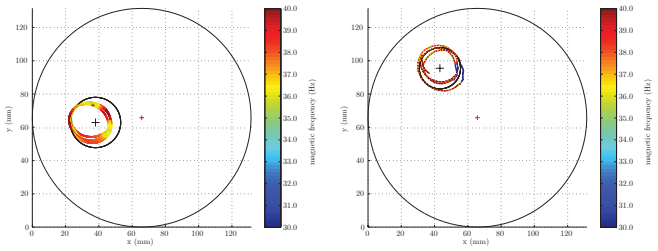

Fig. 9. The figures showing the path followed by two swimmers while tracking a circle. To the left a CCW swimmer with a receding horizon of 1s, to the right a CW swimmer with a receding horizon of 0.2 s

to follow motion primitives. Through hardware experiments, we validated our dynamic models and implemented them to follow circular paths.

These results are preliminary. Future work should extend these results to obstacle/collision avoidance and point-to-point manipulation. For instance, to construct a feasible path from a start to goal location, we could use our motion primitives as a set of inputs for a Rapidly-Exploring Random Tree [20, Chap. 14]. Similar methods could be employed for obstacle avoidance. Finally, while we have demonstrated that self-assembled swimmers have unique dynamic models, work remains before we can simultaneously control multiple swimmers.

REFERENCES

[1] A. Snezhko, I. S. Aranson, and W.-K. Kwok, "Surface wave assisted self-assembly of multidomain magnetic structures." *Phys Rev Lett*, vol. 96, no. 7, p. 078701, 2006. [Online]. Available: http://www.biomedsearch.com/nih/Surface-wave-assisted-self-assembly/16606148.html

[2] M. Belkin, A. Snezhko, I. S. Aranson, and W.-K. Kwok, "Driven magnetic particles on a fluid surface: pattern assisted surface flows." *Phys Rev Lett*, vol. 99, no. 15, p. 158301, 2007. [Online]. Available: http://www.biomedsearch.com/nih/Driven-magnetic-particles-fluid-surface/17995219.html

[3] A. Snezhko, M. Belkin, I. S. Aranson, and W.-K. Kwok, "Self-assembled magnetic surface swimmers." *Phys Rev Lett*, vol. 102, no. 11, p. 118103, 2009. [Online]. Available: http://www.biomedsearch.com/nih/Self-assembled-magnetic-surface-swimmers/19392241.html

[4] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of Mathematics*, vol. 79, no. 3, pp. 497–516, 07 1957. [Online]. Available: http://www.jstor.org/stable/2372560

[5] G. M. Whitesides and B. Grzybowski, "Self-assembly at all scales," *Science*, vol. 295, no. 5564, pp. 2418–2421, 2002. [Online]. Available: http://www.sciencemag.org/content/295/5564/2418.abstract

[6] T. D. Murphey, J. Bernheisel, D. Choi, and K. M. Lynch, "An example of parts handling and self-assembly using stable limit sets," in *Int. Conf. Int. Rob. Sys.*, Aug. 2005, pp. 1624–1629.

[7] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1115–1130, Dec. 2006.

[8] S. S. Sudo Sudo and T. Honda, "Magnetic swimming mechanism in a viscous liquid," *Journal of Intelligent Material Systems and Structures*, vol. 17, no. 8-9, pp. 729–736, 2006.

[9] J. J. Abbott, K. E. Peyer, M. C. Lagomarsino, L. Zhang, L. X. Dong, I. K. Kaliakatsos, and B. J. Nelson, "How should microrobots swim?" *Int. J. Rob. Res.*, July 2009, doi:10.1177/0278364909341658.

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

[11] S. Floyd, E. Diller, C. Pawashe, and M. Sitti, "Control methodologies for a heterogeneous group of untethered magnetic micro-robots," *I. J. Robotic Res.*, vol. 30, no. 13, pp. 1553–1565, Nov. 2011.

[12] E. Diller, S. Floyd, C. Pawashe, and M. Sitti, "Control of multiple heterogeneous magnetic microrobots in two dimensions on nonspecialized surfaces," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 172–182, Feb. 2012.

[13] P. Abbeel, A. Coates, and a. Y. Ng, "Autonomous Helicopter Aerobatics through Apprenticeship Learning," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1608–1639, Jun. 2010. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364910371999

[14] S. Vijayakumar, A. D'Souza, and S. Schaal, "Incremental online learning in high dimensions." *Neural computation*, vol. 17, no. 12, pp. 2602–34, Dec. 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16212764

[15] A. Snezhko, I. Aranson, and W.-K. Kwok, "Dynamic self-assembly of magnetic particles on the fluid interface: Surface-wave-mediated effective magnetic exchange," *Physical Review E*, vol. 73, no. 4, p. 041306, Apr. 2006. [Online]. Available: http://pre.aps.org/abstract/PRE/v73/i4/e041306

[16] S. Schaal, "Receptive Field Weighted Regression," *Learning*, 1997.

[17] S. Klanke, "A Library for Locally Weighted Projection Regression," *Journal of Machine Learning Research*, vol. 9, pp. 623–626, 2008.

[18] W. Li and E. Todorov, "Iterative linear-quadratic regulator design for nonlinear biological movement systems," in *Proceedings of the First International Conference on Informatics in Control, Automation, and Robotics*. Citeseer, 2004, pp. 222–229. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.85.5196\&rep=rep1\&type=pdf

[19] B. D. O. Anderson and J. B. Moore, *Optimal control: linear quadratic methods*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1990.

[20] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.