2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

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 [4]) 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 [5]) to track motion primitives given visual feedback. We validate our approach in hardware experiments with the system shown in Fig. 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.

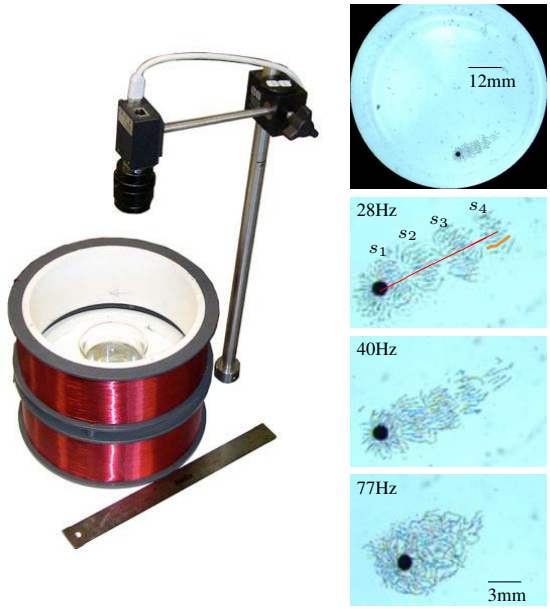

Fig. 1. Our hardware platform (left) and the resulting self-assembled robotic "swimmer" (right), which arises when an alternating magnetic field is applied perpendicular to an air-water interface that contains a suspension of ferro-magnetic micro-particles and a non-magnetic bead. The structure of the swimmer changes with the frequency of the alternating field—shown is the same swimmer at three frequencies. The four swimmer segments are indicated by s. One chain has been colored in orange and the centerline is shown in red. The multimedia attachment shows swimmer formation.

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 by Whitesides and Grzybowski [6]. They define self-assembly 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, from applications such as self-arrangement in parts-handling [7] to task-oriented self-assembly of modular robots [8]. 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 [9]–[11].

Sudo et al. presented a 5mm magnet with a flexible tail, propelled by an external magnetic field [9], 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 versus 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 [10], [11] 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 [12], [13]. In this work a single control signal was applied to each magnet, but unlike the helical swimmers of [10], [11], 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

The self-assembled swimmers consist of nickel microparticles suspended on the liquid-air interface. Driven by a collective behavior and due to an external alternating magnetic field, these particles congregate into a snake-like structure capable of generating water flows on the surface. We now describe the dynamics behind the self-assembly process, the steady-state behavior and their response to changes on the magnetic field. This work leads to the design of a modeling algorithm, described in Section IV. A brief description of the hardware platform used is also presented.

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 magnetic field (20 to 120Hz), the magnetic moments from the particles are aligned ferromagnetically along the chain direction driven by dipole-dipole interactions [1]. On a larger scale, these 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-like structure. This fluid motion is nearly equal for frequency values below 85Hz [3]. If a glass bead (1-1.5mm in diameter) is placed near one end, it will slow fluid flow at that end and the liquid displacement will be higher at the opposite end producing a net forward displacement.

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

By adjusting the magnetic field frequency we are capable of modifying the segment arrangement and characteristic length, resulting in a change of the differential water flow from both sides. As the frequency value is increased the segment sections contract, increasing both the overall water flow and the difference in flows from both sides. This inhomogeneity enables swimmer steering by selectively choosing a frequency value to achieve a particular turning rate or swimming curvature.

On a perfectly symmetric swimmer, varying the control frequency would only result in a change of the forward speed with no modification of the turning rate. In practice, all swimmers have some degree of asymmetry, resulting in a variety of possible curvatures. In our experiments, we often deliberately exaggerate asymmetries when we introduce a bead.

Below critical frequency values [14] the swimmer structure is stable, subject to changes only when chains detach from their corresponding segment due to collision with the beaker wall or 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 40Hz frequency range for driving the swimmers because it offers two relevant advantages: within

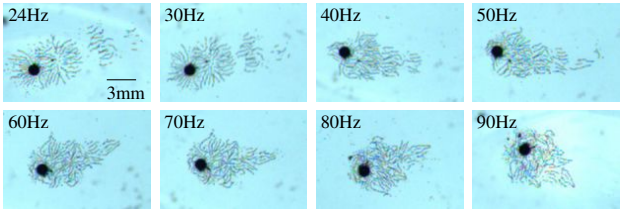

Fig. 2. One self-assembled swimmer shown at increasing magnetic frequencies. The swimmer begins with four well-defined segments, but looses structure when the frequency increases above 40Hz. If the magnetic frequency returns to 40Hz, the swimmer will have a new shape and new dynamics.

this range the swimmer configuration is mostly reversible and this range exhibits the largest changes in velocity and turning rate. As Fig. 2 depicts, above 40Hz the segments of a swimmer tend to loose structure.

B. Hardware Platform

Our hardware platform consists of a cylindrical glass container (145mm in diameter) inside a Helmholtz coil (220mm in diameter, 83mm in height). The coil is energized by an alternating voltage generated by a dedicated 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 (camera: Basler A601f / lens: Edmund Industrial Optics 35mm double gauss 54689). A backlight is added using a flat illumination screen obtained from a laptop computer.

IV. Model Learning

The swimmer structural configuration defines its dynamics. The swimmer formation process is stochastic, so we cannot create a reliable model until the swimmer forms. We could attempt to obtain a model whose parameters depend on the observed features of the swimmer, but this approach would require both complex image processing and an expensive, high-resolution vision system. Instead we choose to use simpler computer vision algorithms to locate the swimmer head, record a history of these positions under a range of excitation frequencies, and use this data to learn an online forward dynamic model, where future swimmer states can be predicted based on past observations. Several regression methods (e.g [4], [15], [16]) have been applied for the control of robotic systems (e.g. [4], [17]–[19]). We want a method to learn a forward model with the following characteristics:

1) requires no previous knowledge of the swimmer dynamics
2) is fast to compute for both learning and evaluation
3) is capable of on-line learning

These criteria imply we are looking for an incremental regression method that is structurally adaptive and is based on local models. We choose Locally Weighted Projection Regression (LWPR). This is a regression method that builds local linear regressions of a non-linear function and can operate on-line. Fig. 3 shows a block diagram of this process.

A. Locally Weighted Projection Regression

Locally Weighted Projection Regression (LWPR) (see [4]) is an extension of Receptive Field Weighted Regression in [20]. LWPR approximates a nonlinear function by piecewise linear models. The learning process consists of choosing K, the number of local models, and obtaining the characteristic parameters $\mathbf{b}_k$ of the hyperplanes that describe each local model. A crucial step is determining the region of validity, also called the Receptive Field (RF), in which each linear model can be trusted.

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

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

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 N local models are chosen, the prediction is calculated by a weighted average

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

and the regression function can be written as:

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

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 become 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. [21].

B. Swimmer Modeling

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

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k), \quad k = 0, \ldots, K-1 \qquad (3)$$

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 (3) as:

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

On the magnetic swimmer the state and control input are defined as

$$\mathbf{x}_k = \left[r_k,\ \beta_k,\ \phi_k\right]^\top, \quad \mathbf{u_k} = u_k\ . \qquad (5)$$

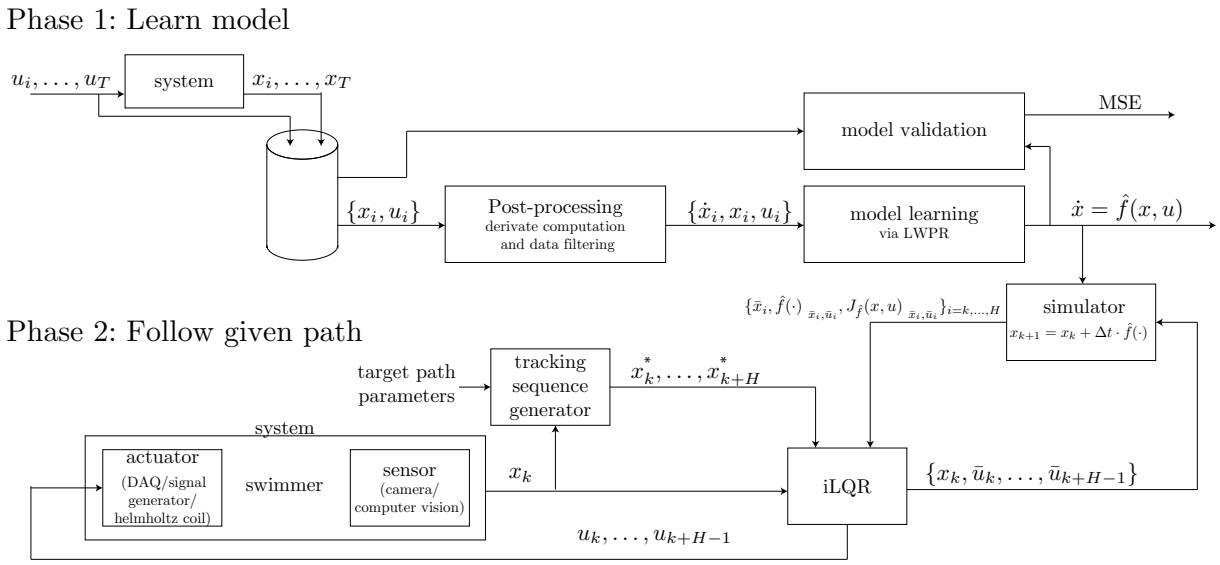

Fig. 3. Block diagram of model learning and our control strategy, explained in Sections IV and V.

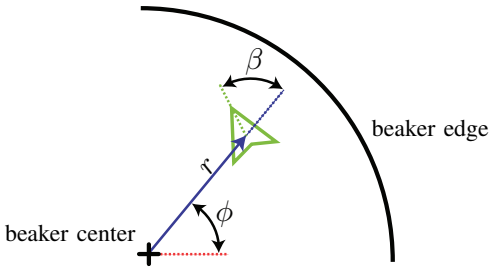

Fig. 4. Coordinate frame (r, β, ϕ) used for magnetic swimmers. The swimmer is shown in green.

The state components r_k and ϕ_k are the magnitude and angle of the vector pointing from the world center to the swimmer's head, and β denotes the angle of the body with respect to this vector, as shown in Fig. 4. 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. Experimentally, A_k is kept fixed and $u_k = \psi_k$.

Assuming symmetric dynamics with respect to ϕ, we approximate the general form in (4) by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta t \cdot \hat{\mathbf{f}}(\pi(\mathbf{x}_k), \mathbf{u}_k) \qquad \pi \mapsto r, \beta. \tag{6}$$

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:

$$\begin{bmatrix} (r_{k+1} - r_k)/\Delta t \\ (\beta_{k+1} - \beta_k)/\Delta t \\ (\phi_{k+1} - \phi_k)/\Delta t \end{bmatrix} = \begin{bmatrix} M_{\hat{r}}(r, \beta, u) \\ M_{\hat{\beta}}(r, \beta, u) \\ M_{\hat{\phi}}(r, \beta, u) \end{bmatrix}. \tag{7}$$

We start the learning process by collecting real-time data. The system is driven for 240s by a repeating triangular input from 30Hz to 40Hz with a 10-second period, and the resulting state measurements are collected. Fig. 5 shows representative data. The time history of states is then smoothed

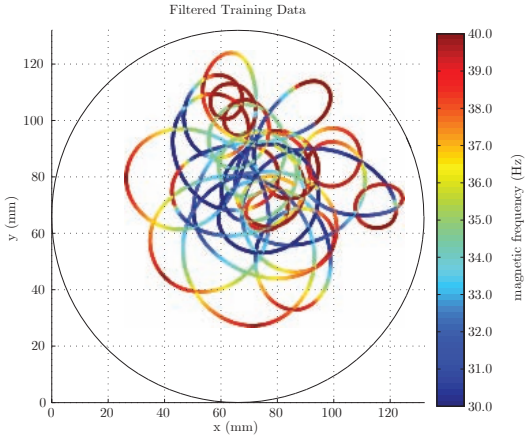

Fig. 5. Typical training data used for learning a model.

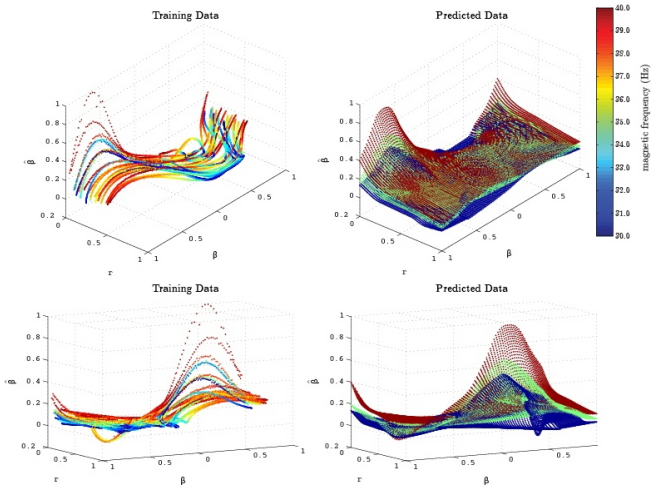

Fig. 6. Training data and learned model $M_{\hat{\beta}}(r, \beta, u) \mapsto \hat{\beta}$. Similar models are created for $M_{\hat{r}}(r, \beta, u)$, and $M_{\hat{\phi}}(r, \beta, u)$.

using a low-pass Butterworth filter that takes advantage of the time dependance. Example data and the learned model are shown in Fig. 6.

Our objective is to learn a model that, given the state $\mathbf{x}_k$ and the control input $\mathbf{u}_k$, predicts the resulting state $\mathbf{x}_{k+1}$

The result is a nonlinear approximation of (6) which we use in Section V to simulate our system dynamics and design a LQR controller. Fig. 3 provides an overview of this process.

V. Control Strategy

Our objective is to obtain a control law $u_k^* = \pi_k(\mathbf{x}_k)$ for following circular paths. We use the nonlinear dynamical model learned in Section IV. We define the following optimal control problem that penalizes deviation from the desired state $\mathbf{x}_k^*$ and control effort $\mathbf{u}_k^*$.

$$\begin{aligned} \underset{u_k}{\text{minimize}} \quad & \frac{1}{2}\sum_{k=0}^{K}(\delta\mathbf{x}_k^\top \mathbf{Q}\delta\mathbf{x}_k + \delta\mathbf{u}_k^\top \mathbf{R}\delta\mathbf{u}_k) \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \end{aligned} \tag{8}$$

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) [5], also known as Gauss-Newton LQR and Sequential LQR, is a method for solving optimal control problems with a nonlinear cost function and dynamical model:

$$\begin{aligned} \underset{\mathbf{u}_k}{\text{minimize}} \quad & J = \sum_{k=0}^{K}\mathbf{l}_k(\mathbf{x}_k, \mathbf{u}_k) \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k). \end{aligned} \tag{9}$$

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 (9), and continues by computing a linear approximation of the dynamics and a quadratic approximation for the cost around the nominal trajectory $\bar{\mathbf{X}}_k^{(i)}$. Essentially iLQR transforms a nonlinear optimal control problem to a linear time-varying optimal control problem:

$$\begin{aligned} \underset{\mathbf{u}_k}{\text{minimize}} \quad & J = \sum_{k=0}^{K}(\mathbf{x}_k^\top \mathbf{Q}\mathbf{x}_k + \mathbf{u}_k^\top \mathbf{R}\mathbf{u}_k) \\ \text{subject to} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k. \end{aligned} \tag{10}$$

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 see [22]). 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 (9), we apply the method described in Section V-A with the following extensions and modifications:

- To improve convergence to a local optimum, we use the dynamic model

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

  and decrease α each iteration, to switch from the desired trajectory when $\alpha = 1$ to the system model when $\alpha = 0$. In practice, computational speed restricts the number of iterations we can do during real-time control, but experimentally, three iterations with $\alpha = \{0.7, 0.1, 0.05\}$ suffice 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 are 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:

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

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

We can compute a feedback law for the complete motion primitive (from 7s to 30s), but our simulator performance quickly degrades after 5s. Our solution is similar to a receding horizon controller. We use a computed feedback law for 1s and then recompute the law, using our current position as the new initial position.

The swimmer is moving forward whenever an alternating magnetic field is applied. Removing the field causes the structure to rearrange, changing the dynamic behavior. This is relevant since in our implementation, the time K^* required to compute the optimal control sequence is approximately 10% of the sequence duration. During this time we cannot apply feedback control. This issue is solved by propagating in simulation the current state K^* steps forward using a nominal control input and feeding the resulting state as the initial condition for our computation.

We use this methodology to track motion primitives that are each circular arcs. The user specifies the radius r^* and center location c_x, c_y for the circle, and the algorithm follows this path by projecting the current point to the target path (i.e. the closest point on 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 experimental trial, the magnetic field is turned off and $<$0.1g of 45μm nickel spheres are deposited over the

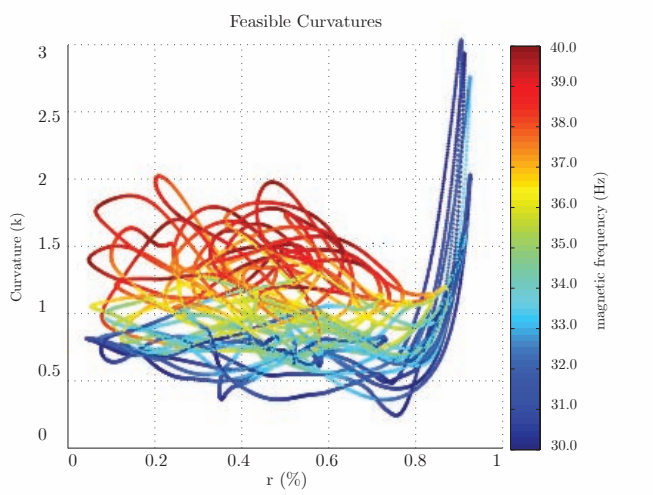

Fig. 7. Curvature as a function of r and u calculated using 400s of data from a single swimmer. When the swimmer reaches the border, i.e. $r = 1$, meniscus effects bounce it back, resulting in high curvature. Each swimmer has a unique response to the input u, and has maximum and minimum curvatures which constrain feasible paths.

water-filled container (27.5mm water height) using a sieve (200-mesh). We then generate a sinusoidal magnetic field at 30Hz and allow the magnetic moments of the micro-particles to orient and form chains and anti-ferromagnetically oriented segments, a process described in Section III-A. Once a stable swimmer has been formed we linearly increase and decrease the frequency from 30Hz to 40Hz every 10s, an action that stabilizes the segments. At this point the swimmer is nearly symmetric and the water flow at each end is balanced, resulting in a static structure.

To induce an asymmetric structure we manually place a glass bead (1.5mm diameter Ni-Pd-Ni), and one end and position it slightly offset from the centerline, creating the swimmer's head. Offsetting the head to the right produces a counterclockwise swimmer, while placing it to the left produces a clockwise swimmer. Further manipulation with a permanent magnet may be required to make the swimmer more compact.

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. Next, this 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. Fig. 7 shows a plot of curvature and u vs r.

At this point the user may specify a target circle on the user interface and start the control process. The data generated 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, each following a circular

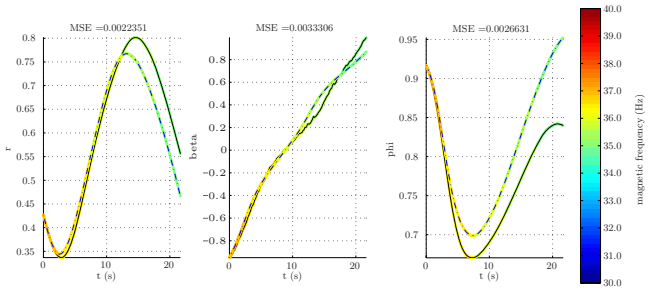

Fig. 8. Predicted (dashed line) vs. actual (solid line) paths as a function of time. The simulator prediction has low error up to 5s. This value is used for calculating the controller's time horizon.

			Model S_1	Model S_2
Data	$\hat{r}$	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	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	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

NORMALIZED MEAN SQUARE ERROR (NMSE) BETWEEN ACTUAL DATA AND VALUES PREDICTED USING LWPR FOR TWO SWIMMERS, S_1 AND S_2. DATA IS SEPARATED INTO A TRAINING SET (80%) AND A TESTING SET (20%). $26,490$ POINTS WERE USED FOR $S1$ AND $20,791$ FOR S_2.

motion primitive. We describe the results obtained for the model learning and motion primitive tracking phases.

1) *Model Learning:* Table I summarizes the performance of the models learned for swimmers $S1$ and $S2$. As expected, the nMSEs obtained from the training data are lower than those from the test data. Additionally, the cross-validation errors ($S1$ model vs. $S2$ data, $S2$ model vs. $S1$ data) are up to 3x greater than those from the test data. This emphasizes the importance of learning a new model for each swimmer. As seen in Fig. 8, the simulator accurately predicts the swimmer's trajectory up to a horizon of 5s.
2) *Motion Primitive Tracking:* Our controller was able to reliably follow circular paths at multiple origin and radius values. Fig. 9 shows two representative experiments for different swimmers and receding horizons.

These results are preliminary. Currently it is difficult to produce a solution of nickel particles that will generate a stable swimmer. Less than 10% of particle solutions resulted in swimmer formation. Once a satisfactory solution of nickel particles was found, slightly more than 50% of trials were successful (success defined as gathering training data, learning a model, and tracking one revolution of a circle.)

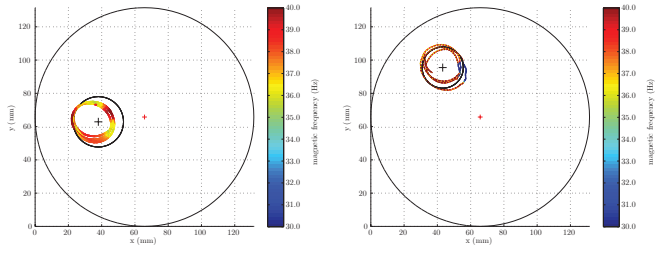

Fig. 9. Paths followed by two different swimmers while tracking a circle. Left: a CCW swimmer with a receding horizon of 1s. Right: a CW swimmer with a receding horizon of 0.2s

VII. Conclusion and Future Work

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

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

These results are preliminary. The swimmer's structure varies over time, and the learned model should continue to adapt. Future work could extend these results to obstacle/collision avoidance and point-to-point manipulation. The multimedia attachment shows a human controlling the magnetic field frequency for both of these tasks. There are many methods to automate this. 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 [23, 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. The video attachment illustrates some of the challenges involved with multiple swimmers. Collisions are more likely, and fluid flow from one swimmer can disrupt another.

VIII. Acknowledgements

The authors thank Hugo Leon for constructing the experimental apparatus. This work was supported by the National Science Foundation under CPS-0931871 and CMMI-0956362.

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] 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

[5] 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

[6] 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

[7] 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.

[8] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proceedings of the IEEE*, vol. 96, no. 9, pp. 1490–1508, Sep. 2008.

[9] S. Segawa 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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

[15] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, 2004.

[16] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[17] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 167–172, Oct. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354701

[18] P. Abbeel, C. A, and A. Y. Ng, "Autonomous helicopter aerobatics through apprenticeship learning," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, Jun. 2010. [Online]. Available: http://ijr.sagepub.com/cgi/doi/10.1177/0278364910371999

[19] J. Morimoto and G. Zeglin, "Minimax differential dynamic programming: Application to a biped walking robot," *Robots and Systems, 2003*, no. i, pp. 1–6, 2003. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1248926

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

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

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

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