Summer Internship Report

Feedback Control of Purcell's Swimmer for Trajectory Tracking

Name: Pratik J
Dept. of Mechanical Engineering
National Institute of Technology, Tiruchirappalli

Guide: Dr.Ravi N Banavar
Systems and Control Dept.,
IIT Bombay (Summer, 2017)

4th July 2017

Abstract

The Purcell's swimmer is a really intriguing topic mainly because its dynamics is not very straight forward in terms of intuitive visualization, which is the primary reason for it being such a challenging and alluring research subject. This report deals with the modeling and simulation of basic motion primitives, which when executed in a particular sequence lead to a system displacement as per the requirement. The report also talks about the use of a vision based feedback system employed to achieve finer trajectory tracking. As a starting exercise a, solution to the parallel parking problem was simulated on MatLab, which employed a four step drive/steer sequence. Simulations were made for implementing motion primitives for trajectory tracking under open loop and feedback controlled trajectory tracking for a random natural drift in the system.

Objectives

The objectives for the summer internship were as follows :

- Simulation and analysis of the solution to the parallel parking problem.
- Simulating motion primitives for the three-link Purcell's swimmer :
    1. Symbolic Calculation of each of the Lie Bracket terms
    2. Using these Lie Bracket terms to generate a linear combination that results in motion in a desired direction
    3. Calculating co-efficients of each of the terms in the linear combination for pure X, pure Y and pureθ
- Generating parameter plots and a simulation for each of the motion primitives
- Implementing trajectory tracking under open loop.
- Implementing closed loop control, using a vision based feedback loop for accurate trajectory tracking.
- Simulating an implementation of feedback correction for a random drift added in an affine manner to the system, to simulate the effects of a natural drift in the system.

1 Literature Review

Swimming at low Reynold's number is very different from a general notion of swimming primarily because, inertial effects play almost no significant role against the effects of viscosity.This means that any body moving in such conditions will cause the local boundary layer to move along with it, or as stated by E.M.Purcell, "you can't shake off your environment" [3].

In his paper, Purcell proposes a gait to generate motion for the three link swimmer. This, drift-free three link swimmer is our subject of interest for this paper as shown in Fig. 1

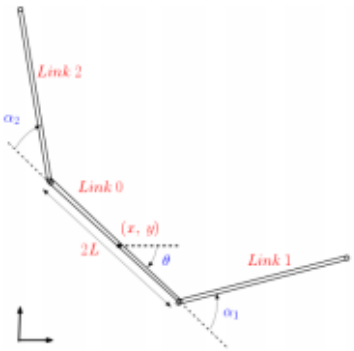


Figure 1: Purcell's Three Link Swimmer

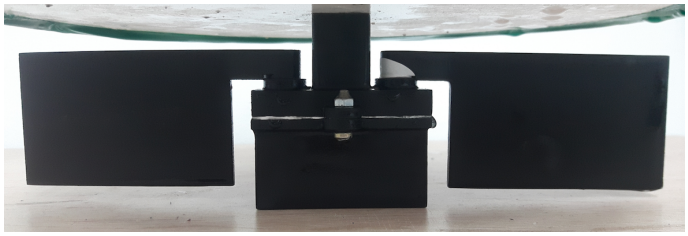


Figure 2: Purcell's Three Link Swimmer

, and its dynamics is explicitly defined [1] and due to very low Reynold's numbers, the momentum terms are dropped, reducing the equation to,

$$\zeta = -A(r)\dot{r}$$

,where A is the local connection form matrix, ζ is the body velocity vector, $\begin{bmatrix}\dot{x} & \dot{y} & \dot{\theta}\end{bmatrix}^\top$ and $r = (\alpha 1, \alpha 2)$, $\dot{r}$ being the shape velocity.

The drag forces and moments are thereby found to be functions of ζ. This enables us to write the net force $F = \begin{bmatrix}F_x & F_y & M\end{bmatrix} \top$ as,

$$F = \omega(\alpha) \begin{bmatrix}\zeta \\ \dot{\alpha}\end{bmatrix}$$

which on reducing for low Reynold's number conditions (net force on isolated system is zero) gives, $A = \omega_1^{-1}\omega_2$. We will use this explicit definition of the connection form in later sections of this report.

2 Parallel Parking & Non-Holonomy

2.1 Problem Overview

The parallel parking problem is quite an fundamental exercise to understand the most basic square gait, a four step drive-steer control sequence and mainly the significance of non-holonomic constraints in day to day life. The MatLab code we wrote makes our two wheeled car start with the axle, centered at the origin, parallel to the X-axis.

2.2 The Solution

1. The car first drives straight up to the first plot in Fig.3.
2. It then performs a steering motion with the current axle center being the center of rotation to the position shown in the top right plot of Fig. 3.
3. The car then drives backward maintaining the angular orientation to the position show in the bottom left plot of Fig. 3.
4. Finally performing a steering maneuver as in Step 2 to orient itself parallel to the X-axis arriving at the position shown in the last plot of Fig. 3.

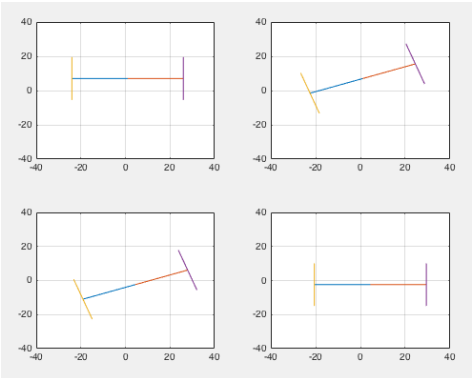


Figure 3: Plots showing the square-gait, drive-steer control sequence

2.3 Analysis

Evidently performing these motions sequentially enables us to have a net translation in a direction that seems impossible to achieve with the given set of available actuations. There is a negligible deviation in the Y direction X direction, which can be minimized by decreasing the time-step for which each of the drives and steers are performed.

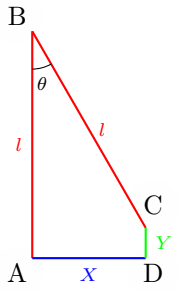


Figure 4: Diagram to calculate the final position of the car

Fig.4 shows the trajectory of the car (red lines), starting at the bottom left. Taking the length of each red line to be l and the steer angle to be θ, we get the net displacements along the axes as,

$$X = \bar{AD} = \sin\theta \tag{1}$$

$$Y = \bar{CD} = 2l\sin^2\frac{\theta}{2} \tag{2}$$

3 Drift-Free Purcell's Swimmer Simulation

3.1 The Algorithm for Generating Motion Primitives

The user first inputs the desired final position. This is the state to be achieved after performing a sequential set of shape changes. The dynamic model described above links the body velocity directly to the shape velocity through the local connection form matrix. So, the first step is to calculate the connection form matrix, which is achieved using the explicit definition in Sec.1

Having found the connection form matrix, we can define the state space equation of our system as [2],

$$\begin{aligned}\begin{bmatrix}\dot{\alpha_1}\\ \dot{\alpha_1}\\ \zeta\end{bmatrix} &= \begin{bmatrix}1\\ 0\\ A(\alpha_1,\alpha_2)\end{bmatrix}\dot{\alpha_1} + \begin{bmatrix}0\\ 1\\ A(\alpha_1,\alpha_2)\end{bmatrix}\dot{\alpha_2}\\ &= g_1(\alpha_1,\alpha_2)\dot{\alpha_1} + g_2(\alpha_1,\alpha_2)\dot{\alpha_2}\end{aligned} \tag{3}$$

Here g_1, g_2 are the control vector fields. Having found these vector fields we proceed to find the Lie brackets, $[g_1, g_2]$, $[g_1, [g_1, g_2]]$, $[g_2, [g_1, g_2]]$. A linear combination of these 5 vector fields will span the entire Reachable set of any point on our swimmer along the space manifold M.

We stop at the second order Lie bracket and don't consider higher order Lie bracket because for controllability at all points in its configuration space, according to the Chow's theorem we have, [2]

$$span(g_1, g_2, [g_1, g_2], [g_1, [g_1, g_2]], [g_2, [g_1, g_2]]) = span(T_qQ)$$

Since the rank of T_qQ is 5, we consider only the first 5 terms of the Lie bracket expansion and thus Lie brackets of higher order aren't taken into consideration.

The next task in our algorithm is to find the co-efficients of each of the Lie bracket terms corresponding to each of the motion primitive direction vectors.

$$\begin{bmatrix} g1 & g2 & [g_1,g_2] & [g_1,[g_1,g_2]] & [g_2,[g_1,g_2]] \end{bmatrix} \begin{bmatrix} C1 \\ C2 \\ C3 \\ C4 \\ C5 \end{bmatrix} = \begin{bmatrix} B1 \\ B2 \\ B3 \\ B4 \\ B5 \end{bmatrix} \tag{4}$$

where, $C_i (1 \leq i \leq 5)$ is the matrix of coefficients and $B_i (1 \leq i \leq 5)$ is the desired final orientation-position matrix.

It is found that for all the three motion primitives, *i.e.* pure longitudinal displacement, pure lateral displacement and pure orientation change, the co-efficients of g_1, g_2 are 0 and the other three are say, α, β, γ. Once these are calculated, they are used in the net flow composition equation [2] given below to generate the desired motion,

$$\begin{aligned}
\phi_t^{\alpha[g_1,g_2]\beta[g_1,[g_1,g_2]]\gamma[g_2,[g_1,g_2]]} &= \lim_{n\to\infty}[(\phi_{t/n}^{\alpha[g_1,g_2]} \circ \phi_{t/n}^{\beta[g_1,[g_1,g_2]]} \circ \phi_{t/n}^{\gamma[g_2,[g_1,g_2]]})^n] \\
&= \lim_{n\to\infty}[(\phi_{\sqrt{t}/n}^{-g_2} \circ \phi_{\sqrt{t}/n}^{-\alpha g_1} \circ \phi_{\sqrt{t}/n}^{g_2} \circ \phi_{\sqrt{t}/n}^{\alpha g_1})^n \circ \\
&(((\phi_{\sqrt{(\sqrt{t})}/n}^{g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{g_1} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{-g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{-g_1})^n) \circ \phi_{\sqrt{t}/n}^{-\beta g_1} \circ \\
&((\phi_{\sqrt{(\sqrt{t})}/n}^{-g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{-g_1} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{g_1})^n) \circ \phi_{\sqrt{t}/n}^{\beta g_1})^n] \\
&(((\phi_{\sqrt{(\sqrt{t})}/n}^{g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{g_1} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{-g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{-g_1})^n) \circ \phi_{\sqrt{t}/n}^{-\gamma g_2} \circ \\
&((\phi_{\sqrt{(\sqrt{t})}/n}^{-g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{-g_1} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{g_2} \circ \phi_{\sqrt{(\sqrt{t})}/n}^{g_1})^n) \circ \phi_{\sqrt{t}/n}^{\gamma g_2})^n]^n
\end{aligned} \tag{5}$$

The intuitive meaning of the above expanded equation is that each of the flow terms is executed sequentially with the specified actuation for the specified time-step.

As discussed in Sec.2.3, there is a slight deviation in directions other than the desired direction, but this deviation approaches zero as n tends to ∞.

The next step in the algorithm is to implement Eq.(5). In my code I have created three separate functions, Flow1,Flow2,Flow3. Each of these functions takes the current displacement and orientation as inputs and calculates the limb angles and their angular velocities at each time-step for the α, β and γ controlled

flow brackets using the calculated values of α, β and γ respectively.

These functions are then put together to implement Eq.(5) and thus generating a simulation for each of the required motion primitives.

In each of the Flow functions, the angular actuation input$(\dot{\alpha_1}, \dot{\alpha_2})$ is defined for each time step in a loop. For each of these time-steps, the limb angles(α_1, α_2) are calculated and are substituted into the connection form matrix. The body frame velocity is then calculated as per Eq.3.

This body-frame velocity matrix is then converted into its Lie Algebra form and each element of this matrix is multiplied with the time-step. The matrix exponential of the resultant matrix gives the net change in the displacement vector per for that time-step. This exponential matrix is then multiplied with the previous displacement vector to give the position and orientation of the swimmer at the end of said time-step, in the form of the current displacement vector. This loop is then run n-times to give the net desired motion.

The flow of the algorithm is described in the pseudo-code in Fig.5 below.

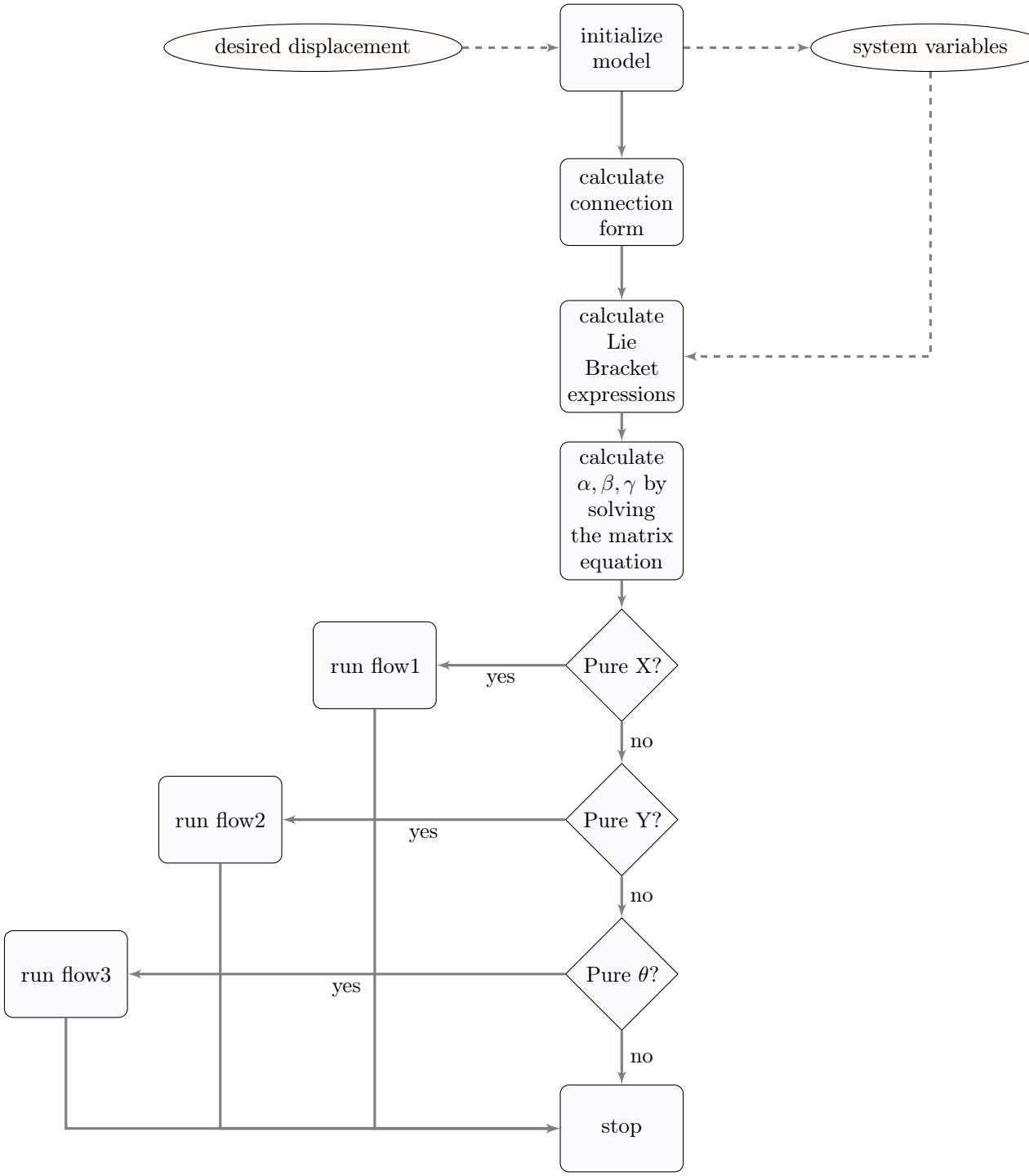


Figure 5: Flow of the Main Code

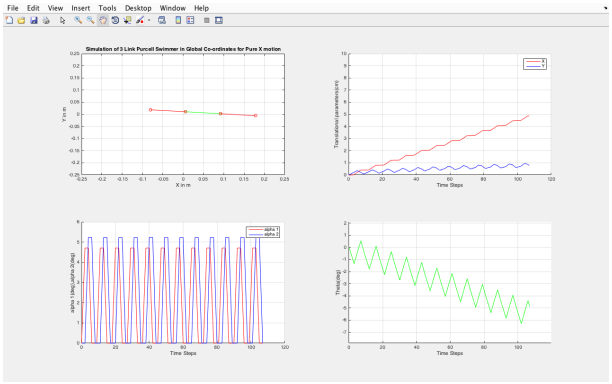


Figure 6: Parameter plots for Pure-X motion

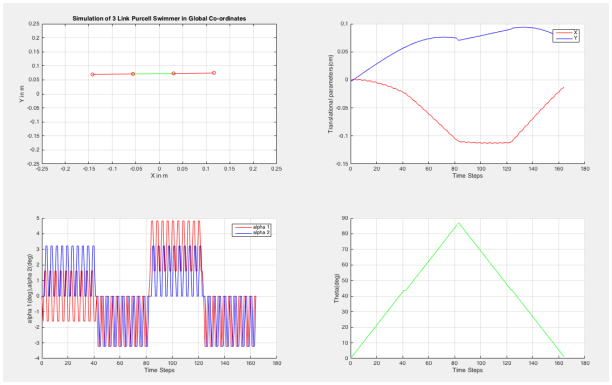


Figure 7: Parameter plots for Pure-Y motion

3.2 Analysis of the Code

The calculations show that for pure-X motion, α is non-zero and β, γ are reduced to zero, for pure-Y motion $\beta$is negative, γ is positive and both have same magnitudes, and for pure-Theta motion, β, γ take equal, positive values, while α remains zero for both pure-Y and pure-Theta cases.

These results imply that for pure-X we do not need to call Flow2 and Flow3 which implement the β and γ terms respectively, since they are zero and their internal terms nullify each others actuation effect, thus making these functions redundant for pure-X.

Similarly, for pure-Y and pure-Theta Flow1 becomes redundant and is thus not called.

The parameter plots for each of the motion primitives is shown in Fig.6,Fig.7,Fig.8 respectively.

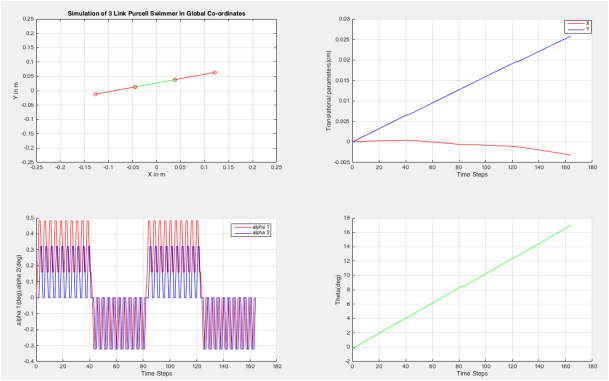


Figure 8: Parameter plots for Pure-T motion

For simulation purposes n was taken to be 100, because of which the parameter plots shown,Fig.6,Fig.7,Fig.8, have small discrepancies. Theoretically when n tends to ∞ these discrepancies die out. However, the hardware lays a constrain on the minimum time-step of execution, so having a sufficiently large n will cause the deviations to die out considering a drift-less system.

The code is not based on the longitudinal to lateral viscous force ratio mentioned in [1]. The value of viscous forces per unit length is calculated using an AnSys simulation of the model as described in [2]. The results in table 1 give the new values for the co-efficients of the linear combination corresponding to each motion primitive.

motion	α	β	γ
pure X	0.8999	0	0
pure Y	0	-0.8871	0.8871
pure θ	0	0.8871	0.8871

Table 1: Calculated values for co-efficients

The current code for pure-Y implements β terms first then $-\gamma$, instead of the composition described in Eq.5. When n tends to ∞ both these compositions should converge, but for the n value considered in the code, the said approach is marginally more accurate.

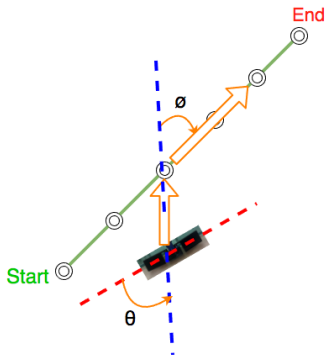


Figure 9: Correction algorithm

4 Closed Loop Control

The above sections of this paper talks about a drift-less system all along. However, as is the case with most systems, there is a natural, undefined drift that cannot be precisely model and varies indefinitely in time. This drift may be a result of non-uniform density of the fluid used, mixed impurities, or even the presence of small disturbances in the fluid medium due to varying boundary conditions.

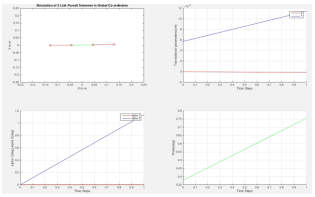
Whatever be the cause, for an accurate trajectory tracking, from a control perspective, this drift has to be accounted for and corrected.

4.1 Closed Loop Algorithm

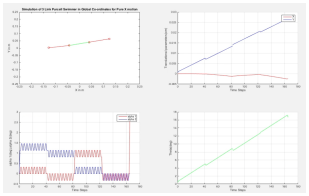
The trajectory to be tracked is first mapped with a set of checkpoints or markers. The vision based feedback system gives the position and orientation of the swimmer in real time. The markers enable us to calculate how much the swimmer has deviated from the path since it passed the previous marker.

Having calculated that, we calculate the duration of pure-Theta primitive to be executed in order to align it along the straight line joining its current position and the next marker.

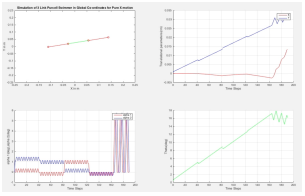
Then, we calculate the distance to be traversed along the longitudinal direction, and the pure-X primitive is implemented for the appropriate duration, following which a pure-Theta orients the swimmer back along the required trajectory. This whole process is represented in Fig.9



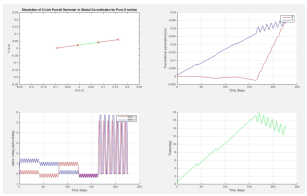
(a) Starting offset position



(b) End of theta maneuver



(c) Translation to next marker



(d) End Position

Figure 10: Plots at various stages of the closed loop implementation

4.2 Closed Loop Control Simulation

For the simulation, an artificial drift is forced onto the body velocity by altering Eq.3 in an affine manner. We add a drift term to this equation with a random number generator within $0\% - 5\%$ of the actual velocity at that time-step.

The current position is acquired and compared to the position vector of the nearest marker. It then subtracts the current position vector from that of the next marker and actuates accordingly so as to be consistent with the control logic discussed in 4.1.

Fig.10 show implementation of one such correction between two markers.

The desired trajectory is the line $y = x$ and the swimmer is at an initial offset position to simulate a deviation since the previous marker. The swimmer then corrects its trajectory according to the algorithm above.

The existence of a slight deviation even after the correction is implemented is possibly a consequence of a much smaller n

Acknowledgements

I would like to thank Dr.Ravi Banavar for providing me an opportunity to undertake this project and for guiding and motivating me throughout. I would like to thank Sudin Kadam and Pulkit Katdare for their mentorship, Aseem Borkar for his extended support in setting up the vision system and Mr. Kumar for his insights on hardware selection.

References

[1] Ross L Hatton and Howie Choset. Geometric swimming at low and high reynolds numbers. *IEEE Transactions on Robotics*, 29(3):615–624, 2013.

[2] Kedar Joshi, Naman Gupta, Pulkit Katdare, Sudin Kadam, and Ravi Banavar. Trajectory tracking using motion primitives for the purcell's swimmer. *arXiv preprint arXiv:1703.06731*, 2017.

[3] Edward M Purcell. Life at low reynolds number. *American journal of physics*, 45(1):3–11, 1977.