Swimming and steering of artificial microswimmer in viscous fluid

Aine Zhang

Shanghai High School International Division
Shanghai, China

Under the directions of

Mr. Hui Liao

Shanghai High School International Division
Shanghai, China

Professor Mian Li

Shanghai Jiao Tong University, Shanghai, China

Swimming and steering of artificial microswimmer in viscous fluid

Aine Zhang
Shanghai High School International Division

Nature is alive with all kinds of swimming microorganisms. The research on the locomotion of these microswimmers helps us not only to understand the micro world but also to engineer a variety of useful artificial devices. This report investigates swimming behaviors and steering of a microswimmer in a viscous fluid. We begin with background and then introduce a mathematical model of the microswimmer under study. The analytic expressions of dynamics are explicitly derived, which enables us to study basic properties of time evolutions and to examine symmetries in the dynamics as well as their ramifications on the swimming trajectory. Using Green's formula, we reveal the relation between gait symmetry and net rotation. Then we perform comprehensive investigations on the gaits with symmetric patterns and their resulting swimming trajectories. Finally, we study the steering of microswimmer by formulating it into an optimization problem and solving it by gradient algorithm. We apply sophisticated Automatic Differentiation tools to calculate the gradient efficiently. Various optimization results are discussed, especially those of steering by symmetric gaits.

Keywords: Dynamics, Symmetry, Optimization, Gradient algorithm

Contents

I. INTRODUCTION

Nature is alive with all kinds of swimming microorganisms, *e.g.*, protozoa, bacteria, algae, and sperm cells. These microorganisms move around by continuously changing their body shapes. Locomotion is one of their fundamental survival skills, as they need to move for nutrition, to evade predators or toxins, and to get fertilized, among others.

The governing physical laws in this micro world are completely different from those at macro scale. This is a world of *low Reynolds number* as Nobel laureate E. M. Purcell pointed out in his seminal talk [12]. The locomotion herein presents a different set of challenges, and these microorganisms have developed effective swimming strategies after evolving millions of years.

Researchers have strong interests in understanding these strategies [10]. This not only brings in new knowledge to biology, mechanics, mathematical control, and robotics, but also helps to engineer micro scale devices that can propel themselves to fulfill biomedical tasks such as medicine delivery and artery unblocking.

In this research we aim to study a prototype artificial microswimmer, *i.e.*, *three-link swimmer*, proposed by Purcell [12]. With a simple geometric configuration, this device contains all the necessary ingredients to study swimming and steering in a low Reynolds number world. It has thus attracted increasing research interests in the past few decades [11].

Based on a mathematical model of the microswimmer from the literature, we first derive the analytic expressions of complete swimming dynamics. This leads to the study of basic properties of time evolutions, symmetries in the dynamics, and net rotations. Moreover, we perform comprehensive investigations on the gaits with symmetric patterns. We then study the steering of the microswimmer to a prescribed location and orientation by formulating it into an optimization problem. This is subsequently solved by gradient descent algorithm with Automatic Differentiation tools. Various optimization results are discussed, especially those of steering by symmetric gaits. This project manifests nicely how mathematics can be utilized to analyze and solve physical problems.

II. BACKGROUND AND MATHEMATICAL MODEL

In this section, we will introduce some background knowledge about the microswimmer in a viscous fluid, and then present a mathematical model of three-link microswimmer [5, 7, 11].

A. A world with low Reynolds number

To study swimming behaviors, we start from the famous Navier-Stokes equation in fluid mechanics. The flow field $\mathbf{u}$ and pressure p of a fluid satisfy

$$-\nabla p + \eta \nabla^2 \mathbf{u} = \rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u}, \tag{1}$$

where ρ is the fluid density, η the viscosity, together with some appropriate boundary conditions.

The Navier-Stokes equation is one of the most difficult problems in mathematics, as it was among the seven Millennium Prize problems raised by the Clay Mathematics Institute in 2000. Fortunately, we do not need to solve this equation to study the behaviors of microswimmers—the physical nature greatly simplifies the mathematics. Define the *Reynolds (Re) number* of a swimming object as the ratio of its inertial force to viscous one, *i.e.*,

$$\mathrm{Re} = \frac{av\rho}{\eta},$$

where a represents the object size, and v the surrounding fluid velocity [12].

The Reynolds number is a dimensionless quantity that characterizes different flow regimes. Microswimmers have very low Reynolds numbers. For example, a swimming *E. Coli* in water at normal atmospheric pressure and room temperature has it around 10^{-4}. In comparison, a human swimming in water has Reynolds number around 10^6, and a goldfish around 10^2 [12].

In a low Reynolds number world, viscous force dominates inertial one. The physical consequence is that every action has only instantaneous effect, and time makes no difference—only shape changes. Therefore, it is safe to discard the time dependent terms in the right hand side of Eq. (1) (*i.e.*, inertial terms) and obtain the Stokes equation:

$$-\nabla p + \eta \nabla^2 \mathbf{u} = 0. \tag{2}$$

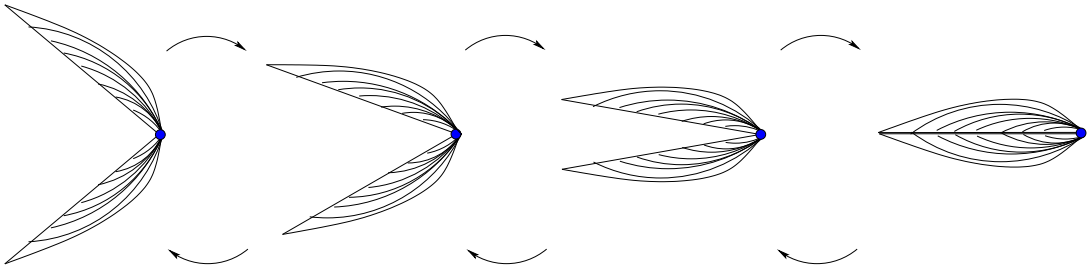


FIG. 1: In macro world, a scallop can propel itself toward right by opening its shell slowly and closing it quickly. However, this strategy does not work in a low Reynolds number world, since each opening makes a movement that will be canceled by later closings.

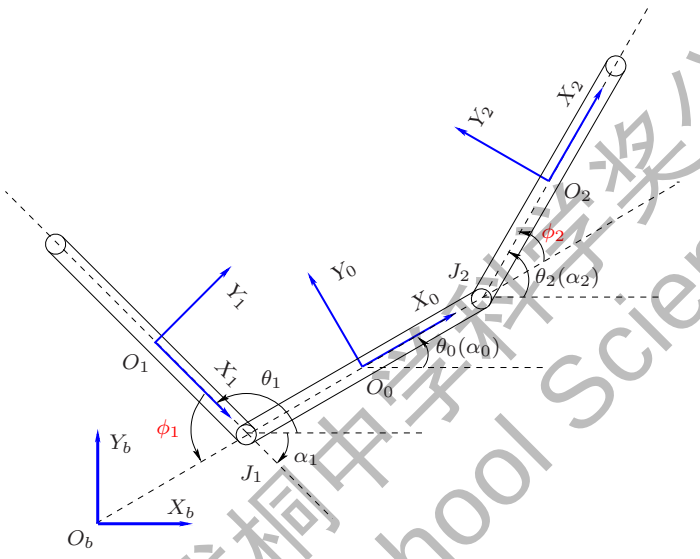


FIG. 2: Geometric configuration of a three-link microswimmer that can move around by varying its two joint angles ϕ_1 and ϕ_2.

A microswimmer can propel itself by deforming its shape in a certain pattern that is often called a *gait* (or *stroke*). A famous result is the Scallop theorem [12], which asserts that a reciprocal gait generates no net motion. Here reciprocal gait means a sequence of shape changes followed by the same sequence in reverse. This can be illustrated by Fig. 1. In macro world, a scallop can propel itself toward right by opening its shell slowly and closing it quickly,. However, this strategy does not work in a low Reynolds number world. This is because the scallop has only one joint—the movement gained from the opening will be canceled by later closings and thus prohibits it from moving anywhere.

Naturally, the simplest microswimmer that can move around has at least two joints. This leads to the three-link microswimmer originally proposed by Purcell [12] and recently realized in the laboratory [9]. In this research, we will investigate mathematical properties of its swimming and steering.

B. Mathematical model of three-link microswimmer

This subsection introduces a mathematical model of the three-link microswimmer. There exist several different models, and we mainly follow the one developed in Ref. [11].

Fig. 2 illustrates the geometric configuration of a three-link microswimmer. Label the central link as 0, and two lateral links as 1 and 2. For computational simplicity, assume the lengths for all three links as $2l_0$ and their masses are evenly distributed. Link 0 and link 1 are connected by joint J_1, and link 0 and link 2 by J_2. The joint angles ϕ_1 and ϕ_2 describe the rotations of link 1 and link 2 with respect to the central link.

Fix a coordinate system $O_bX_bY_b$ to which all the objects can be referenced. This is usually called *base* or *world* frame in mechanics [13]. Denote the middle point of link i as O_i and attach a body frame $O_iX_iY_i$ to it, where $i = 1$, 2, 3. The coordinate of O_i in the base frame $O_bX_bY_b$ is represented by a position vector $p_i = [x_i, y_i]^T$, and the

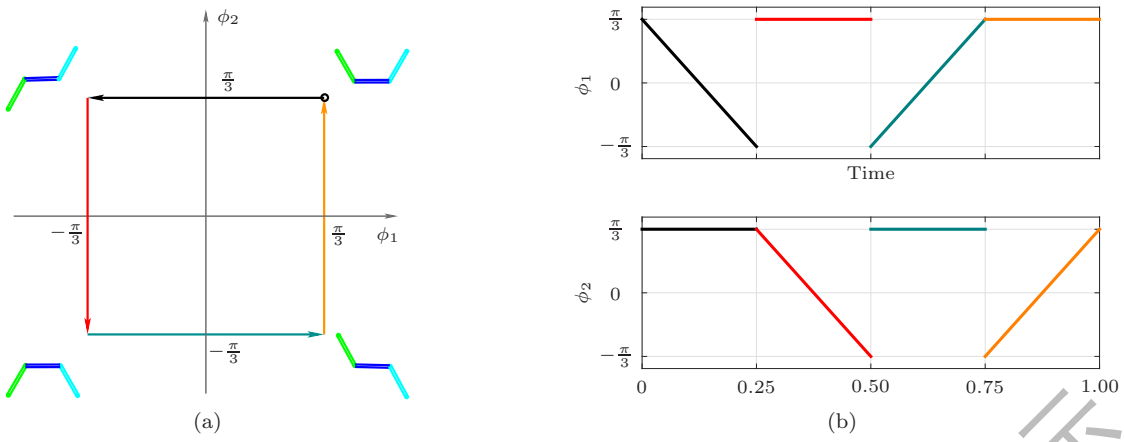


FIG. 3: (a) A square gait in counterclockwise direction for the three-link microswimmer; (b) joint angles ϕ_1 and ϕ_2 as functions of time t.

orientations of $\overrightarrow{J_1O_0}$, $\overrightarrow{J_1O_1}$, and $\overrightarrow{J_2O_2}$ are represented by θ_0, θ_1, and θ_2, respectively. Denote the positive x direction of link i as α_i in the base frame. Then, the angles θ_i can be written as

$$\theta_0 = \alpha_0, \quad \theta_1 = \pi + \alpha_1 = \pi + \theta_0 - \phi_1, \quad \theta_2 = \alpha_2 = \theta_0 + \phi_2. \tag{3}$$

Definition 1 (Posture and Gait) Define the *posture* of link i as a vector $\mathbf{x}_i$ consisting of the position vector p_i of its center O_i and the link orientation θ_i in the base frame, *i.e.*,

$$\mathbf{x}_i = \begin{bmatrix} p_i \\ \hline \theta_i \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \hline \theta_i \end{bmatrix}, \quad i = 0, 1, 2.$$

A *gait* (or *stroke*) is a continuous function $\phi(t) = [\phi_1(t), \phi_2(t)]^T$ of joint angles on time interval $[t_0, t_f]$ with $\phi(t_0) = \phi(t_f)$.

The posture describes the state of a link in the base frame, and the gait forms a closed path in the $\phi_1\phi_2$ plane. The three-link microswimmer moves around by applying gaits, that is, varying its two joint angles.

Example 1 (Square gait) Fig. 3 shows a square gait [12]. The initial condition $\phi_1(0) = \phi_2(0) = \frac{\pi}{3}$ is marked by a black circle. This gait contains four legs to form a closed path:

1. Leg 1 (black): ϕ_1 decreases from $\frac{\pi}{3}$ to $-\frac{\pi}{3}$, and ϕ_2 keeps constant at $\frac{\pi}{3}$;
2. Leg 2 (red): ϕ_1 keeps constant at $-\frac{\pi}{3}$, and ϕ_2 decreases from $\frac{\pi}{3}$ to $-\frac{\pi}{3}$;
3. Leg 3 (dark green): ϕ_1 increases from $-\frac{\pi}{3}$ to $\frac{\pi}{3}$, and ϕ_2 keeps constant at $-\frac{\pi}{3}$;
4. Leg 4 (orange): ϕ_1 keeps constant at $\frac{\pi}{3}$, and ϕ_2 increases from $-\frac{\pi}{3}$ to $\frac{\pi}{3}$.

Fig. 3(b) plots an example of how $\phi_1(t)$ and $\phi_2(t)$ evolve as functions of time. The gait can evolve in different time trajectories; however, this will not affect the resulting swimming trajectory as shown later.

From the geometric configuration in Fig. 2, the postures $\mathbf{x}_1$ and $\mathbf{x}_2$ can be represented by

$$\mathbf{x}_1 = \begin{bmatrix} x_1 \\ y_1 \\ \theta_1 \end{bmatrix} = \mathbf{x}_0 - l_0 \begin{bmatrix} \cos\theta_0 \\ \sin\theta_0 \\ 0 \end{bmatrix} - l_0 \begin{bmatrix} \cos\alpha_1 \\ \sin\alpha_1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \pi - \phi_1 \end{bmatrix}, \tag{4}$$

$$\mathbf{x}_2 = \begin{bmatrix} x_2 \\ y_2 \\ \theta_2 \end{bmatrix} = \mathbf{x}_0 + l_0 \begin{bmatrix} \cos\theta_0 \\ \sin\theta_0 \\ 0 \end{bmatrix} + l_0 \begin{bmatrix} \cos\alpha_2 \\ \sin\alpha_2 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \phi_2 \end{bmatrix}. \tag{5}$$

Take derivatives of Eqs. (4) and (5):

$$\begin{aligned}\dot{\mathbf{x}}_1 = \begin{bmatrix}\dot{x}_1\\ \dot{y}_1\\ \dot{\theta}_1\end{bmatrix} &= \dot{\mathbf{x}}_0 - l_0\dot{\theta}_0\begin{bmatrix}-\sin\theta_0\\ \cos\theta_0\\ 0\end{bmatrix} - l_0(\dot{\theta}_0-\dot{\phi}_1)\begin{bmatrix}-\sin\alpha_1\\ \cos\alpha_1\\ 0\end{bmatrix} - \begin{bmatrix}0\\0\\ \dot{\phi}_1\end{bmatrix}\\ &= \begin{bmatrix}\dot{x}_0 + l_0\sin\theta_0\dot{\theta}_0 + l_0\sin\alpha_1(\dot{\theta}_0-\dot{\phi}_1)\\ \dot{y}_0 - l_0\cos\theta_0\dot{\theta}_0 - l_0\cos\alpha_1(\dot{\theta}_0-\dot{\phi}_1)\\ \dot{\theta}_0-\dot{\phi}_1\end{bmatrix}\\ &= \underbrace{\begin{bmatrix}1 & 0 & l_0\sin\theta_0 + l_0\sin\alpha_1\\ 0 & 1 & -l_0\cos\theta_0 - l_0\cos\alpha_1\\ 0 & 0 & 1\end{bmatrix}}_{D_1}\begin{bmatrix}\dot{x}_0\\ \dot{y}_0\\ \dot{\theta}_0\end{bmatrix} + \underbrace{\begin{bmatrix}-l_0\sin\alpha_1 & 0\\ l_0\cos\alpha_1 & 0\\ -1 & 0\end{bmatrix}}_{E_1}\begin{bmatrix}\dot{\phi}_1\\ \dot{\phi}_2\end{bmatrix},\end{aligned} \tag{6}$$

and

$$\begin{aligned}\dot{\mathbf{x}}_2 = \begin{bmatrix}\dot{x}_2\\ \dot{y}_2\\ \dot{\theta}_2\end{bmatrix} &= \dot{\mathbf{x}}_0 + l_0\dot{\theta}_0\begin{bmatrix}-\sin\theta_0\\ \cos\theta_0\\ 0\end{bmatrix} + l_0(\dot{\theta}_0+\dot{\phi}_2)\begin{bmatrix}-\sin\alpha_2\\ \cos\alpha_2\\ 0\end{bmatrix} - \begin{bmatrix}0\\0\\ \dot{\phi}_2\end{bmatrix}\\ &= \begin{bmatrix}\dot{x}_0 - l_0\sin\theta_0\dot{\theta}_0 - l_0\sin\alpha_2(\dot{\theta}_0+\dot{\phi}_2)\\ \dot{y}_0 + l_0\cos\theta_0\dot{\theta}_0 + l_0\cos\alpha_2(\dot{\theta}_0+\dot{\phi}_2)\\ \dot{\theta}_0-\dot{\phi}_2\end{bmatrix}\\ &= \underbrace{\begin{bmatrix}1 & 0 & -l_0\sin\theta_0 - l_0\sin\alpha_2\\ 0 & 1 & l_0\cos\theta_0 + l_0\cos\alpha_2\\ 0 & 0 & 1\end{bmatrix}}_{D_2}\begin{bmatrix}\dot{x}_0\\ \dot{y}_0\\ \dot{\theta}_0\end{bmatrix} + \underbrace{\begin{bmatrix}0 & -l_0\sin\alpha_2\\ 0 & l_0\cos\alpha_2\\ 0 & 1\end{bmatrix}}_{E_2}\begin{bmatrix}\dot{\phi}_1\\ \dot{\phi}_2\end{bmatrix}.\end{aligned} \tag{7}$$

Denote the linear velocity of center O_i in the base frame as $\mathbf{v}_i$ and the angular velocity as ω_i, *i.e.*,

$$\mathbf{v}_0 = \dot{p}_0 = \begin{bmatrix}\dot{x}_0\\ \dot{y}_0\end{bmatrix},\quad \mathbf{v}_1 = \dot{p}_1 = \begin{bmatrix}\dot{x}_1\\ \dot{y}_1\end{bmatrix},\quad \mathbf{v}_2 = \dot{p}_2 = \begin{bmatrix}\dot{x}_2\\ \dot{y}_2\end{bmatrix},$$

and

$$\omega_0 = \dot{\theta}_0,\quad \omega_1 = \dot{\theta}_1 = \dot{\theta}_0 - \dot{\phi}_1,\quad \omega_2 = \dot{\theta}_2 = \dot{\theta}_0 + \dot{\phi}_2.$$

Then,

$$\dot{\mathbf{x}}_i = \begin{bmatrix}\dot{p}_i\\ \dot{\theta}_i\end{bmatrix} = \begin{bmatrix}\mathbf{v}_i\\ \omega_i\end{bmatrix}.$$

From Resistive Force Theory (RFT) [4], the viscous drag force $\mathbf{f}_i$ experienced by link i can be decomposed into the vector sum of a tangential force $\mathbf{f}_i^x$ and a normal force $\mathbf{f}_i^y$, where $\mathbf{f}_i^x$ is proportional to the tangential velocity $\mathbf{v}_i^x$ and $\mathbf{f}_i^y$ proportional to the normal velocity $\mathbf{v}_i^y$:

$$\mathbf{f}_i^x = -c^x l_0\mathbf{v}_i^x,\quad \mathbf{f}_i^y = -c^y l_0\mathbf{v}_i^y, \tag{8}$$

and $c^y = 2c^x$. Notice that the unit tangential vector is $[\cos\alpha_i, \sin\alpha_i]^T$ and the unit normal vector is $[-\sin\alpha_i, \cos\alpha_i]^T$. Thus we can write

$$\mathbf{v}_i^x = \begin{bmatrix}\cos\alpha_i\\ \sin\alpha_i\end{bmatrix}\begin{bmatrix}\cos\alpha_i & \sin\alpha_i\end{bmatrix}\mathbf{v}_i,\quad \mathbf{v}_i^y = \begin{bmatrix}-\sin\alpha_i\\ \cos\alpha_i\end{bmatrix}\begin{bmatrix}-\sin\alpha_i & \cos\alpha_i\end{bmatrix}\mathbf{v}_i. \tag{9}$$

Substitute Eq. (9) into (8):

$$\begin{aligned}\mathbf{f}_i^x &= -c^x l_0\begin{bmatrix}\cos\alpha_i\\ \sin\alpha_i\end{bmatrix}\begin{bmatrix}\cos\alpha_i & \sin\alpha_i\end{bmatrix}\mathbf{v}_i = -c^x l_0\begin{bmatrix}\cos^2\alpha_i & \sin\alpha_i\cos\alpha_i\\ \sin\alpha_i\cos\alpha_i & \sin^2\alpha_i\end{bmatrix}\mathbf{v}_i,\\ \mathbf{f}_i^y &= -c^y l_0\begin{bmatrix}-\sin\alpha_i\\ \cos\alpha_i\end{bmatrix}\begin{bmatrix}-\sin\alpha_i & \cos\alpha_i\end{bmatrix}\mathbf{v}_i = -c^y l_0\begin{bmatrix}\sin^2\alpha_i & -\sin\alpha_i\cos\alpha_i\\ -\sin\alpha_i\cos\alpha_i & \cos^2\alpha_i\end{bmatrix}\mathbf{v}_i.\end{aligned}$$

Hence,

$$\mathbf{f}_i = \mathbf{f}_i^x + \mathbf{f}_i^y = -c^x l_0 \begin{bmatrix} 1+\sin^2\alpha_i & -\sin\alpha_i\cos\alpha_i \\ -\sin\alpha_i\cos\alpha_i & 1+\cos^2\alpha_i \end{bmatrix} \mathbf{v}_i.$$

Also from RFT, the torque m_i on link i can be written as $m_i = -\frac{1}{6}c^x l_0^3 \omega_i$. Putting together, we have

$$\mathbf{F}_i = \begin{bmatrix} \mathbf{f}_i \\ m_i \end{bmatrix} = -c^x l_0 \underbrace{\begin{bmatrix} 1+\sin^2\alpha_i & -\sin\alpha_i\cos\alpha_i & 0 \\ -\sin\alpha_i\cos\alpha_i & 1+\cos^2\alpha_i & 0 \\ 0 & 0 & \frac{1}{6}l_0^2 \end{bmatrix}}_{P_i} \begin{bmatrix} \mathbf{v}_i \\ \omega_i \end{bmatrix}. \tag{10}$$

Moreover, the net hydrodynamic force and torque in the frame $O_0X_0Y_0$ is given by

$$\mathbf{F}_{\text{net}} = \mathbf{F}_0 + D_1^T\mathbf{F}_1 + D_2^T\mathbf{F}_2 = -c^x l_0(P_0\dot{\mathbf{x}}_0 + D_1^T P_1\dot{\mathbf{x}}_1 + D_2^T P_2\dot{\mathbf{x}}_2).$$

Plugging in Eqs. (6) and (7):

$$\begin{aligned} \mathbf{F}_{\text{net}}/(-c^x l_0) &= P_0\dot{\mathbf{x}}_0 + D_1^T P_1(D_1\dot{\mathbf{x}}_0 + E_1\dot{\phi}) + D_2^T P_2(D_2\dot{\mathbf{x}}_0 + E_2\dot{\phi}) \\ &= (P_0 + D_1^T P_1 D_1 + D_2^T P_2 D_2)\dot{\mathbf{x}}_0 + (D_1^T P_1 E_1 + D_2^T P_2 E_2)\dot{\phi}. \end{aligned} \tag{11}$$

We know that the microswimmer experiences zero net force and torque, *i.e.*, $\mathbf{F}_{\text{net}} = 0$. Rearranging Eq. (11), we obtain

$$\dot{\mathbf{x}}_0 = \begin{bmatrix} \mathbf{v}_0 \\ \omega_0 \end{bmatrix} = -(P_0 + D_1^T P_1 D_1 + D_2^T P_2 D_2)^{-1}(D_1^T P_1 E_1 + D_2^T P_2 E_2)\dot{\phi}. \tag{12}$$

This is the fundamental mathematical model that we will be using to study three-link microswimmer locomotion. It expresses the linear and angular velocities $\dot{\mathbf{x}}_0$ of the middle point O_0 of link 0 in terms of the applied gait ϕ and its velocity $\dot{\phi}$. The swimming trajectory $\mathbf{x}_0(t)$ can then be obtained by solving Eq. (12) with common numerical software such as Matlab.

III. SWIMMING DYNAMICS AND SYMMETRY

With the mathematical model of the microswimmer (12), we are ready to study its swimming dynamics and symmetry. We first derive the detailed analytic expressions for swimming dynamics, and then represent it as a dynamical system on the special Euclidean group $SE(2)$. We investigate basic properties of time evolutions and analyze the symmetry in dynamics, gait, and swimming trajectory. Using Green's formula, we reveal how gait symmetry affects net rotation.

A. Analytic expressions and basic properties of dynamics

We first derive the analytic expressions for Eq. (12), which is the governing equation of three-link microswimmer's locomotion.

Let $R(\theta_0)$ be a rotation matrix, which rotates a vector by an angle of θ_0 counterclockwise:

$$R(\theta_0) = \begin{bmatrix} \cos\theta_0 & -\sin\theta_0 \\ \sin\theta_0 & \cos\theta_0 \end{bmatrix}.$$

We know that $\mathbf{v}_0$ in Eq. (12) is the linear velocity of O_0 in the base frame, and $R(-\theta_0)\mathbf{v}_0$ is its velocity in the $O_0X_0Y_0$ frame. In this frame, the central link will not feel anything about θ_0. Therefore, $R(-\theta_0)\mathbf{v}_0$ cannot be a function of θ_0—it has to be a function of ϕ_1 and ϕ_2 only and we write it as

$$R(-\theta_0)\mathbf{v}_0 = \frac{1}{d_1}G_v\dot{\phi}, \tag{13}$$

where d_1 is the determinant of the matrix $P_0 + D_1^T P_1 D_1 + D_2^T P_2 D_2$:

$$\begin{aligned} d_1/l_0^2 = {} & 8(\cos\phi_1 + \cos\phi_2)(\cos\phi_1\cos\phi_2 - 2\sin\phi_1\sin\phi_2 + 8) + 24\cos\phi_1\cos\phi_2 - 32\sin\phi_1\sin\phi_2 \\ & + 12(\cos^2\phi_1 + \cos^2\phi_2) + 7\cos^2\phi_1\cos^2\phi_2 - 11\cos\phi_1\cos\phi_2\sin\phi_1\sin\phi_2 + 98. \end{aligned} \tag{14}$$

Then Eq. (12) can be rewritten as

$$\begin{aligned}\dot{p}_0 = \mathbf{v}_0 &= \frac{1}{d_1} R(\theta_0) \underbrace{\begin{bmatrix} G_v^{11} & G_v^{12} \\ G_v^{21} & G_v^{22} \end{bmatrix}}_{G_v} \dot{\phi}, \\ \dot{\theta}_0 = \omega_0 &= \frac{1}{d_1} \underbrace{\begin{bmatrix} G_\omega^{11} & G_\omega^{12} \end{bmatrix}}_{G_\omega} \dot{\phi},\end{aligned} \tag{15}$$

Using Matlab Symbolic Toolbox (see Appendix A for codes), we can explicitly calculate the analytic expressions of the entries in G_v as

$$\begin{aligned}\frac{G_v^{11}}{l_0^3} =& -39\sin\phi_1 - \frac{55}{12}\sin 2\phi_1 + \frac{31}{2}\sin\phi_2 + \frac{41}{12}\sin 2\phi_2 - 4\sin(\phi_1 - 2\phi_2) - \frac{1}{3}\sin(\phi_1 + 2\phi_2) \\ & - \frac{11}{6}\sin(2\phi_1 + \phi_2) - 24\sin(\phi_1 - \phi_2) - \frac{11}{12}\sin(2\phi_1 + 2\phi_2) - 8\sin(\phi_1 + \phi_2), \\ \frac{G_v^{12}}{l_0^3} =& \ 39\sin\phi_2 + \frac{55}{12}\sin 2\phi_2 - \frac{31}{2}\sin\phi_1 - \frac{41}{12}\sin 2\phi_1 - 4\sin(2\phi_1 - \phi_2) + \frac{1}{3}\sin(2\phi_1 + \phi_2) \\ & + \frac{11}{6}\sin(\phi_1 + 2\phi_2) - 24\sin(\phi_1 - \phi_2) + \frac{11}{12}\sin(2\phi_1 + 2\phi_2) + 8\sin(\phi_1 + \phi_2), \\ \frac{G_v^{21}}{l_0^3} =& -\frac{80}{3}\cos\phi_1 - \frac{28}{3}\cos\phi_2 - \frac{22}{3}\cos^2\phi_1 - \frac{14}{3}\cos^2\phi_2 - 36\cos\phi_1\cos\phi_2 \\ & + \frac{11}{3}\cos\phi_1\sin\phi_1\sin\phi_2 - \frac{34}{3}\cos\phi_1\cos^2\phi_2 - \frac{11}{3}\cos^2\phi_1\cos\phi_2 - \frac{2}{3}\cos\phi_2\sin\phi_1\sin\phi_2, \\ \frac{G_v^{22}}{l_0^3} =& -\frac{80}{3}\cos\phi_2 - \frac{28}{3}\cos\phi_1 - \frac{22}{3}\cos^2\phi_2 - \frac{14}{3}\cos^2\phi_1 - 36\cos\phi_1\cos\phi_2 \\ & + \frac{11}{3}\cos\phi_2\sin\phi_1\sin\phi_2 - \frac{34}{3}\cos^2\phi_1\cos\phi_2 - \frac{11}{3}\cos\phi_1\cos^2\phi_2 - \frac{2}{3}\cos\phi_1\sin\phi_1\sin\phi_2,\end{aligned} \tag{16}$$

and the entries in G_ω as

$$\begin{aligned}\frac{G_\omega^{11}}{l_0^2} =& \ 32\cos\phi_1 + 12\cos\phi_1\cos\phi_2 - 16\sin\phi_1\sin\phi_2 - 4\cos^2\phi_2 + 4\cos\phi_1\cos^2\phi_2 \\ & + \frac{11}{3}\cos^2\phi_1\cos^2\phi_2 - 8\cos\phi_2\sin\phi_1\sin\phi_2 - \frac{11}{3}\cos\phi_1\cos\phi_2\sin\phi_1\sin\phi_2 + \frac{82}{3}, \\ \frac{G_\omega^{12}}{l_0^2} =& -32\cos\phi_2 - 12\cos\phi_1\cos\phi_2 + 16\sin\phi_1\sin\phi_2 + 4\cos^2\phi_1 - 4\cos^2\phi_1\cos\phi_2 \\ & - \frac{11}{3}\cos^2\phi_1\cos^2\phi_2 + 8\cos\phi_1\sin\phi_1\sin\phi_2 + \frac{11}{3}\cos\phi_1\cos\phi_2\sin\phi_1\sin\phi_2 - \frac{82}{3}.\end{aligned} \tag{17}$$

We now represent the posture dynamics in Eq. (15) by a dynamical system on the Lie group $SE(2)$. Let us introduce the following definition [16].

Definition 2 The special Euclidean group $SE(n)$ is defined as

$$SE(n) = \left\{ A \in \mathbb{R}^{(n+1)\times(n+1)} \;\middle|\; A = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}, RR^T = I, p \in \mathbb{R}^n \right\}.$$

The posture of the central link $\mathbf{x}_0(t) = [x_0(t), y_0(t), \theta_0(t)]^T$ can be represented by a matrix in $SE(2)$:

$$T(\mathbf{x}_0(t)) = \left[\begin{array}{cc|c} \cos\theta_0(t) & -\sin\theta_0(t) & x_0(t) \\ \sin\theta_0(t) & \cos\theta_0(t) & y_0(t) \\ \hline 0 & 0 & 1 \end{array}\right]. \tag{18}$$

Taking derivative of Eq. (18) and plugging in Eq. (15), we obtain

$$\begin{aligned}\dot{T} &= \left[\begin{array}{cc|c} -\dot\theta_0\sin\theta_0 & -\dot\theta_0\cos\theta_0 & \dot x_0 \\ \dot\theta_0\cos\theta_0 & -\dot\theta_0\sin\theta_0 & \dot y_0 \\ \hline 0 & 0 & 0 \end{array}\right] = \left[\begin{array}{cc|c} -\dot\theta_0\sin\theta_0 & -\dot\theta_0\cos\theta_0 & \frac{1}{d_1}R(\theta_0)G_v\dot\phi \\ \dot\theta_0\cos\theta_0 & -\dot\theta_0\sin\theta_0 & \\ \hline 0 & 0 & 0 \end{array}\right] \\ &= \left[\begin{array}{cc|c} \cos\theta_0 & -\sin\theta_0 & x_0 \\ \sin\theta_0 & \cos\theta_0 & y_0 \\ \hline 0 & 0 & 1 \end{array}\right] \left[\begin{array}{cc|c} 0 & -\dot\theta_0 & \frac{1}{d_1}G_v\dot\phi \\ \dot\theta_0 & 0 & \\ \hline 0 & 0 & 0 \end{array}\right] \\ &= \left[\begin{array}{cc|c} \cos\theta_0 & -\sin\theta_0 & x_0 \\ \sin\theta_0 & \cos\theta_0 & y_0 \\ \hline 0 & 0 & 1 \end{array}\right] \underbrace{\frac{1}{d_1}\left[\begin{array}{cc|c} 0 & -G_\omega\dot\phi & G_v\dot\phi \\ G_\omega\dot\phi & 0 & \\ \hline 0 & 0 & 0 \end{array}\right]}_{A(\phi,\dot\phi)}.\end{aligned} \tag{19}$$

Hence we can obtain

$$\dot{T} = TA(\phi,\dot\phi), \quad T(t_0) = T(\mathbf{x}_0(t_0)), \tag{20}$$

where $\mathbf{x}_0(t_0)$ is the initial posture.

For the gait $\phi(t) = [\phi_1(t), \phi_2(t)]^T$ on $[t_0, t]$, the state transition matrix $\Phi(t, t_0)$ can be defined as the swimming trajectory starting from the initial posture $\mathbf{x}_0(t_0) = [0, 0, 0]^T$. That is, it satisfies

$$\dot\Phi(t, t_0) = \Phi(t, t_0)A(\phi,\dot\phi), \quad \Phi(t_0, t_0) = I. \tag{21}$$

Now the swimming trajectory $T(t)$ can be written as $T(t) = T(\mathbf{x}_0(t_0))\Phi(t, t_0)$, since

$$\begin{aligned}\dot T &= T(\mathbf{x}_0(t_0))\dot\Phi = T(\mathbf{x}_0(t_0))\Phi(t, t_0)A(\phi,\dot\phi) = TA(\phi,\dot\phi), \quad \text{and} \\ T(t_0) &= T(\mathbf{x}_0(t_0))\Phi(t_0, t_0) = T(\mathbf{x}_0(t_0)).\end{aligned}$$

As its name suggests, the state transition matrix $\Phi(t, t_0)$ describes the mapping from the initial posture $T(t_0)$ to the final one $T(t)$, and it has these properties [3]:

$$\Phi(t, t_0) = \Phi(t, t_1)\Phi(t_1, t_0), \tag{22}$$

$$\Phi^{-1}(t, t_0) = \Phi(t_0, t). \tag{23}$$

From the expression of A in Eq. (19), we can obtain the following propositions on the time evolution of swimming dynamics.

Proposition 1 (Time invariant) If a gait $\phi(t)$ is delayed by τ, the resulting state transition matrix is also delayed by τ.

Proof: Let $\Psi(t, t_0 + \tau)$ be the state transition matrix for $\phi(t - \tau)$ with $t \geq t_0 + \tau$. Then

$$\dot\Psi(t, t_0 + \tau) = \Psi(t, t_0 + \tau)A(\phi(t - \tau), \dot\phi(t - \tau)), \quad \Psi(t_0 + \tau, t_0 + \tau) = I.$$

Let $t_1 = t - \tau$, where $t_1 \geq t_0$. We have $A(\phi(t - \tau), \dot\phi(t - \tau)) = A(\phi(t_1), \dot\phi(t_1))$. Hence

$$\dot\Psi(t_1 + \tau, t_0 + \tau) = \Psi(t_1 + \tau, t_0 + \tau)A(\phi(t_1), \dot\phi(t_1)), \quad \Psi(t_0 + \tau, t_0 + \tau) = I.$$

Comparing with Eq. (21), we know that $\Psi(t + \tau, t_0 + \tau) = \Phi(t, t_0)$, *i.e.*, $\Psi(t, t_0 + \tau) = \Phi(t - \tau, t_0)$. ■

Hence, the dynamical behavior is time invariant, and it is independent of the initial time t_0. We can then assume $t_0 = 0$ and write the state transition matrix as $\Phi(t)$.

Proposition 2 (Time scaling) If the time of a gait $\phi(t)$ is scaled by a factor k, the time of the resulting state transition matrix is also scaled by k.

Proof: Let $\Psi(t)$ be the state transition matrix for $\phi(kt)$ with $t \geq 0$. Then

$$\dot{\Psi}(t) = \Psi(t)A(\phi(kt), \dot{\phi}(kt)), \quad \Psi(0) = I.$$

Let $t_1 = kt$, where $t_1 \geq 0$. We have $A(\phi(kt), \dot{\phi}(kt)) = kA(\phi(t_1), \dot{\phi}(t_1))$. Hence

$$\frac{d}{dt_1}\Psi(t_1/k) = \Psi(t_1/k)A(\phi(t_1), \dot{\phi}(t_1)), \quad \Psi(0) = I.$$

Comparing with Eq. (21), we know that $\Psi(t/k) = \Phi(t)$, *i.e.*, $\Psi(t) = \Phi(kt)$. ■

This tells us that the swimming trajectory depends only on the gait shape of joint angles but not on how fast it traverses the gait, which was also previously revealed in [12]. Without loss of generality, we can fix a final time, say, $t_f = 1$, in our numerical simulations.

With these two properties, Eqs. (22) and (23) can be rewritten as

$$\Phi(t_1 + t_2) = \Phi(t_1)\Phi(t_2), \tag{24}$$
$$\Phi^{-1}(t) = \Phi(-t). \tag{25}$$

Furthermore, we have the following proposition if reversing the time evolution of $\phi(t)$.

Proposition 3 (Time reversal) For the gait $\phi(-t)$, the resulting state transition matrix is

$$\Phi(-t) = \Phi^{-1}(t) = \begin{bmatrix} R(-\theta_0(t)) & -R(-\theta_0(t))p_0(t) \\ 0 & 1 \end{bmatrix}. \tag{26}$$

Proof: Let $\Psi(t)$ be the state transition matrix for $\phi(-t)$ with $t \geq 0$. In Proposition 2, let $k = -1$. We obtain that $\Psi(t) = \Phi(-t) = \Phi^{-1}(t)$. Explicit calculation of $\Phi^{-1}(t)$ leads to the second equality in Eq. (26). ■

Remark 1 This proposition can be used to interpret Purcell's scallop theorem. If a microswimmer first applies a gait $\phi(t)$ and then its time reverse $\phi(-t)$, the complete state transition matrix is

$$\Phi(t)\Phi(-t) = I,$$

which indicates that it does not have any net motion.

B. Symmetry in dynamics

We now study the symmetry in the swimming dynamics (21). In particular, we are interested in finding out what if we flip the sign of ϕ, *i.e.*, ϕ becomes $-\phi$; or interchange the order of ϕ_1 and ϕ_2, *i.e.*, ϕ becomes $J\phi$, where

$$J = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The sign flip corresponds to the symmetry with respect to the origin, whereas order interchange the symmetry to the line $\phi_2 = \phi_1$.

Some of these properties were previously examined by applying advanced mathematical techniques [1, 7]. Observing that the analytic expressions of d_1, G_v, and G_ω in Eqs. (14), (16), and (17) are all composed of sinusoidal functions of ϕ_1 and ϕ_2, we will directly make use of odd and even symmetries in sine and cosine functions.

From the expression of d_1 in Eq. (14), we know that d_1 remains the same under sign flip and order interchange, *i.e.*,

$$d_1(\phi) = d_1(-\phi) = d_1(J\phi). \tag{27}$$

Next check G_v and G_ω in Eq. (15). For sign flip, the first row of G_v is odd symmetric:

$$G_v^1(-\phi) = -G_v^1(\phi), \tag{28}$$

and the second row of G_v and G_ω are even symmetric:

$$G_v^2(-\phi) = G_v^2(\phi), \quad G_\omega(-\phi) = G_\omega(\phi). \tag{29}$$

For order interchange, we have that

$$G_v^{11}(\phi_2,\phi_1) = -G_v^{12}(\phi_1,\phi_2), \quad G_v^{21}(\phi_2,\phi_1) = G_v^{22}(\phi_1,\phi_2), \quad G_\omega^{11}(\phi_2,\phi_1) = -G_\omega^{12}(\phi_1,\phi_2),$$

which can be written compactly as

$$G_v^1(J\phi) = -G_v^1(\phi)J, \quad G_v^2(J\phi) = G_v^2(\phi)J, \quad G_\omega(J\phi) = -G_\omega(\phi)J. \tag{30}$$

We have the following results on the relationship between gait symmetry and swimming trajectories.

Proposition 4 (Sign flip) Suppose that a three-link microswimmer applies a gait $\phi(t)$ and obtains the state transition matrix $\Phi(t)$. If the microswimmer applies $-\phi(t)$, the resulting state transition matrix is

$$\Phi_1(t) = \Lambda_1\Phi(t)\Lambda_1 = \left[\begin{array}{cc|c} \cos\theta_0(t) & \sin\theta_0(t) & x_0(t) \\ -\sin\theta_0(t) & \cos\theta_0(t) & -y_0(t) \\ \hline 0 & 0 & 1 \end{array}\right], \tag{31}$$

where $\Lambda_1 = \mathrm{diag}\{1,-1,1\}$.

Proof: Taking derivative of $\Phi_1(t)$:

$$\dot\Phi_1 = \Lambda_1\dot\Phi\Lambda_1 = \Lambda_1\Phi A(\phi,\dot\phi)\Lambda_1 = \Lambda_1\Phi\Lambda_1\Lambda_1 A(\phi,\dot\phi)\Lambda_1 = \Phi_1\Lambda_1 A(\phi,\dot\phi)\Lambda_1. \tag{32}$$

From Eq. (19), we get

$$\begin{aligned} A(-\phi,-\dot\phi) &= \frac{1}{d_1(-\phi)}\begin{bmatrix} 0 & -G_\omega(-\phi)(-\dot\phi) & G_v^1(-\phi)(-\dot\phi) \\ G_\omega(-\phi)(-\dot\phi) & 0 & G_v^2(-\phi)(-\dot\phi) \\ 0 & 0 & 1 \end{bmatrix} = \frac{1}{d_1(\phi)}\begin{bmatrix} 0 & G_\omega(\phi)\dot\phi & G_v^1(\phi)\dot\phi \\ -G_\omega(\phi)\dot\phi & 0 & -G_v^2(\phi)\dot\phi \\ 0 & 0 & 1 \end{bmatrix} \\ &= \Lambda_1 A(\phi,\dot\phi)\Lambda_1, \end{aligned} \tag{33}$$

where the second equality is obtained from Eqs. (27), (28), and (29). Substituting Eq. (33) into (32), we have

$$\dot\Phi_1 = \Phi_1 A(-\phi,-\dot\phi), \quad \Phi_1(0) = \Lambda_1\Phi(0)\Lambda_1 = I.$$

Therefore $\Phi_1(t)$ is the state transition matrix for $-\phi(t)$. ■

The sign flip maps a gait $\phi(t)$ to $-\phi(t)$. From Eq. (31), this transforms the original swimming trajectory from $[x_0(t), y_0(t), \theta_0(t)]^T$ to $[x_0(t), -y_0(t), -\theta_0(t)]^T$.

Proposition 5 (Order interchange) Suppose that a three-link microswimmer applies a gait $\phi(t)$ and obtains the state transition matrix $\Phi(t)$. If the microswimmer applies $J\phi(t)$, the resulting state transition matrix is

$$\Phi_2(t) = \Lambda_2\Phi(t)\Lambda_2 = \left[\begin{array}{cc|c} \cos\theta_0(t) & \sin\theta_0(t) & -x_0(t) \\ -\sin\theta_0(t) & \cos\theta_0(t) & y_0(t) \\ \hline 0 & 0 & 1 \end{array}\right]. \tag{34}$$

where $\Lambda_2 = \mathrm{diag}\{-1,1,1\}$.

Proof: Taking derivative of $\Phi_2(t)$:

$$\dot\Phi_2 = \Lambda_2\dot\Phi\Lambda_2 = \Lambda_2\Phi A(\phi,\dot\phi)\Lambda_2 = \Lambda_2\Phi\Lambda_2\Lambda_2 A(\phi,\dot\phi)\Lambda_2 = \Phi_2\Lambda_2 A(\phi,\dot\phi)\Lambda_2. \tag{35}$$

From Eq. (19), we get

$$\begin{aligned} A(J\phi,J\dot\phi) &= \frac{1}{d_1(J\phi)}\begin{bmatrix} 0 & -G_\omega(J\phi)(J\dot\phi) & G_v^1(J\phi)(J\dot\phi) \\ G_\omega(J\phi)(J\dot\phi) & 0 & G_v^2(J\phi)(J\dot\phi) \\ 0 & 0 & 1 \end{bmatrix} \\ &= \frac{1}{d_1(\phi)}\begin{bmatrix} 0 & G_\omega(\phi)J(J\dot\phi) & -G_v^1(\phi)J(J\dot\phi) \\ -G_\omega(\phi)J(J\dot\phi) & 0 & G_v^2(\phi)J(J\dot\phi) \\ 0 & 0 & 1 \end{bmatrix} \\ &= \Lambda_2 A(\phi,\dot\phi)\Lambda_2, \end{aligned} \tag{36}$$

where the second equality is from Eqs. (27) and (30). Substituting Eq. (36) into (35), we have

$$\dot{\Phi}_2 = \Phi_2 A(J\phi, J\dot{\phi}), \quad \Phi_2(0) = \Lambda_2 \Phi(0) \Lambda_2 = I.$$

Therefore Φ_2 is the state transition matrix for $J\phi(t)$. ■

The order interchange maps a gait $[\phi_1(t), \phi_2(t)]$ to $[\phi_2(t), \phi_1(t)]$. From Eq. (34), this transforms the original swimming trajectory from $[x_0(t), y_0(t), \theta_0(t)]^T$ to $[-x_0(t), y_0(t), -\theta_0(t)]^T$.

Remark 2 Combining these two leads to a new symmetry with respect to the line $\phi_2 = -\phi_1$, which maps $[\phi_1(t), \phi_2(t)]$ to $[-\phi_2(t), -\phi_1(t)]$ and transforms the swimming trajectory from $[x_0(t), y_0(t), \theta_0(t)]^T$ to $[-x_0(t), -y_0(t), \theta_0(t)]^T$.

C. Green's formula and net rotation

We observe that the dynamics of orientation θ_0 in Eq. (15) depends on only the applied gait ϕ and its velocity $\dot{\phi}$. From Green's formula

$$\oint P dx + Q dy = \iint_D \left(\frac{\partial Q}{\partial x} - \frac{\partial P}{\partial y} \right) dx dy,$$

we can obtain

$$\begin{aligned} \theta_0(t_f) - \theta_0(0) &= \oint \frac{G_\omega^{11}(\phi)}{d_1(\phi)} d\phi_1 + \frac{G_\omega^{12}(\phi)}{d_1(\phi)} d\phi_2 = \iint_D \left(\frac{\partial}{\partial \phi_1} \frac{G_\omega^{12}(\phi)}{d_1(\phi)} - \frac{\partial}{\partial \phi_2} \frac{G_\omega^{11}(\phi)}{d_1(\phi)} \right) d\phi_1 d\phi_2 \\ &= \iint_D \frac{W(\phi)}{24 d_1^2(\phi)} d\phi_1 d\phi_2. \end{aligned} \tag{37}$$

Using Matlab Symbolic Toolbox, we can get the analytic expression of $W(\phi)$ as

$$\begin{aligned} W(\phi_1, \phi_2) = &-19278(\sin\phi_1 + \sin\phi_2) - 14960(\sin 2\phi_1 + \sin 2\phi_2) - 1342(\sin 3\phi_1 + \sin 3\phi_2) - 22(\sin 4\phi_1 + \sin 4\phi_2) \\ &- 5012\sin(\phi_1 + \phi_2) + 3420\sin(2\phi_1 + 2\phi_2) + 292\sin(3\phi_1 + 3\phi_2) + 66(\sin(3\phi_1 + 4\phi_2) + \sin(4\phi_1 + 3\phi_2)) \\ &+ 10380(\sin(\phi_1 - 2\phi_2) + \sin(\phi_2 - 2\phi_1)) - 2248(\sin(\phi_1 + 2\phi_2) + \sin(2\phi_1 + \phi_2)) \\ &+ 1276(\sin(\phi_1 - 3\phi_2) + \sin(\phi_2 - 3\phi_1)) - 920(\sin(\phi_1 + 3\phi_2) + \sin(3\phi_1 + \phi_2)) \\ &+ 22(\sin(\phi_1 - 4\phi_2) + \sin(\phi_2 - 4\phi_1)) - 132(\sin(\phi_1 + 4\phi_2) + \sin(4\phi_1 + \phi_2)) \\ &+ 240(\sin(2\phi_1 - 3\phi_2) + \sin(2\phi_2 - 3\phi_1)) + 756(\sin(2\phi_1 + 3\phi_2) + \sin(3\phi_1 + 2\phi_2)) \\ &+ 11(\sin(2\phi_1 - 4\phi_2) + \sin(2\phi_2 - 4\phi_1)) + 121(\sin(2\phi_1 + 4\phi_2) + \sin(4\phi_1 + 2\phi_2)). \end{aligned} \tag{38}$$

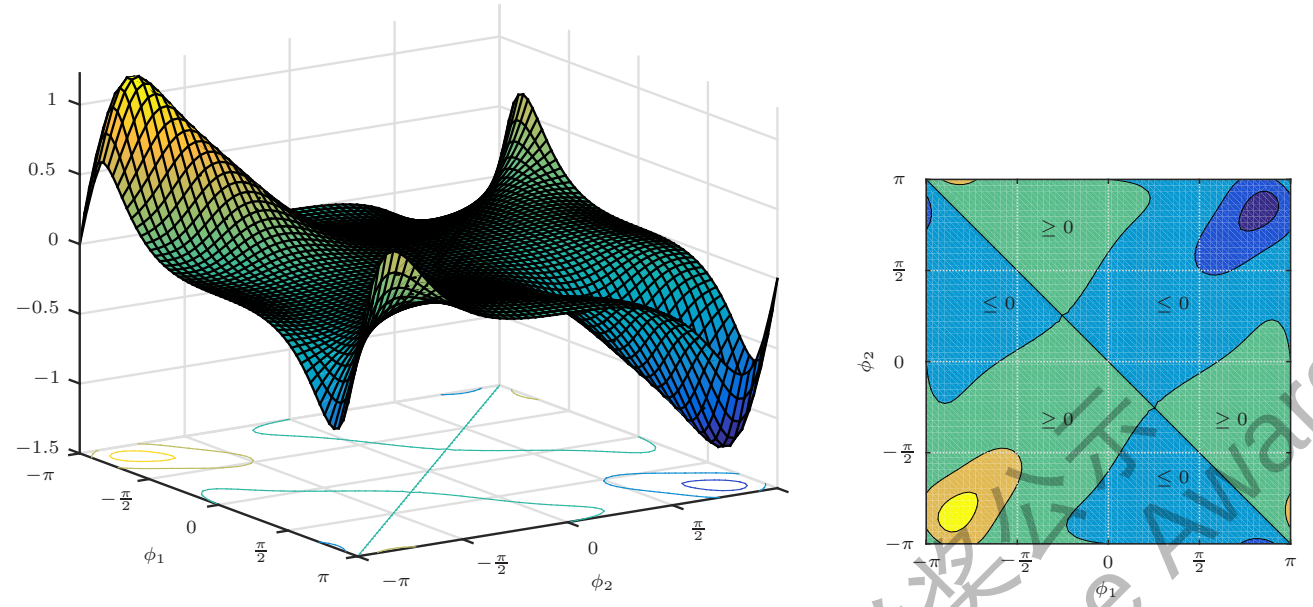
From Eq. (37), we know that the net rotation generated by a gait $\phi(t)$ at the final time t_f is equal to the double integral of $W/24d_1^2$ over the interior area encircled by $\phi(t)$. If $\phi(t)$ moves counterclockwise and gains more positive double integral than negative one, the microswimmer gets a positive net rotation. It is easy to see that W has the following symmetry:

$$W(-\phi_1, -\phi_2) = -W(\phi_1, \phi_2), \quad W(\phi_2, \phi_1) = W(\phi_1, \phi_2), \quad W(-\phi_2, -\phi_1) = -W(\phi_1, \phi_2). \tag{39}$$

We plot the surface and contour of $W/24d_1^2$ vs ϕ_1 and ϕ_2 in Fig. 4.

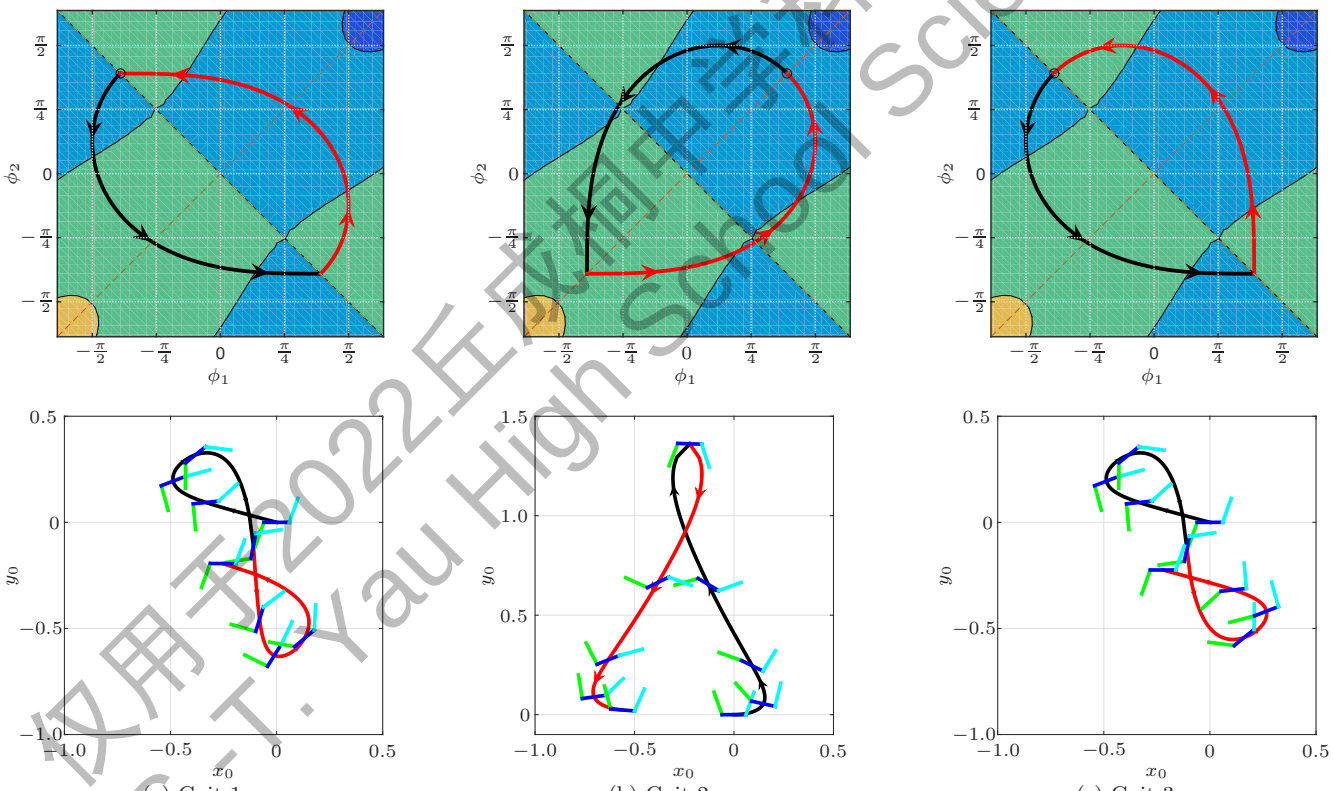
Example 2 Consider the net rotations generated by the three gaits in Fig. 5. These gaits do not intersect themselves and keep the same orientation. Define a positive orientation as when one moves along a gait, the encircled region is always on the left. The starting point is marked by a black circle.

- Fig. 5(a): The gait is odd symmetric with respect to the origin. From Eq. (39), it encircles positive integral exactly canceled by negative one, and thus the microswimmer gets no net rotation from this gait.
- Fig. 5(b): The gait is symmetric with respect to the line $\phi_2 = \phi_1$. In general positive integral cannot be canceled by negative one, and thus the microswimmer gains a net rotation.
- Fig. 5(c): The gait is symmetric with respect to the line $\phi_2 = -\phi_1$. It encircles positive integral exactly canceled by negative one, and thus the microswimmer receives no net rotation.



(a) Surface. (b) Contour.

FIG. 4: The function $W/24d_1^2$ vs ϕ_1 and ϕ_2.



(a) Gait 1. (b) Gait 2. (c) Gait 3.

FIG. 5: (Color online) Top: three symmetric gaits plotted in colored closed paths. Each gait has two legs: the first in black and the second in red. The background is the contour of $W/24d_1^2$ as in Fig. 4(b), and the starting point is marked by a black circle. Gait 1 is symmetric with respect to the origin, Gait 2 to $\phi_2 = \phi_1$, and Gait 3 to $\phi_2 = -\phi_1$. Bottom: the resulting swimming trajectories.

IV. SYMMETRIC GAITS AND TRAJECTORIES

In this section, we investigate a variety of symmetric gaits. We calculate the state transition matrix for each symmetric segment of a gait and then piece them together to obtain the final posture. Some of these gaits yield particularly preferable locomotions such as pure movements along x_0 or y_0 axis. These gaits can be used to form a repertoire to accomplish certain maneuvering tasks.

A. Single symmetric gaits

We first study the gait that is symmetric with respect to a single point or axis, namely, the origin, the line $\phi_2 = \phi_1$, or the line $\phi_2 = -\phi_1$. Such a gait consists of two symmetric legs. Because of the time evolution properties of swimming dynamics discussed in Propositions 1 and 2, we can consider each leg evolves on the time interval $[0, \frac{T}{2}]$ at the same speed. Further, denote the first leg of the gait as $\phi(t) = [\phi_1(t), \phi_2(t)]^T$ and the state transition matrix at the end of the first leg (*i.e.*, $t = \frac{T}{2}$) as

$$\Phi_1 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & -\sin\theta_0(\frac{T}{2}) & x_0(\frac{T}{2}) \\ \sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & y_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{40}$$

We start from Gaits 1, 2, and 3 shown in Fig. 5. These are all simple curves, *i.e.*, not crossing themselves.

Gait 1 (Origin). Consider Gait 1 as shown in Fig. 5(a), whose two legs are symmetric with respect to the origin. Each leg is obtained by rotating and translating a lemniscate curve in the first quadrant[1]

$$\rho^2 = 2\gamma^2 \cos 2\beta, \quad \beta \in \left[0, \frac{\pi}{4}\right]. \tag{41}$$

It starts from a point on the line $\phi_2 = -\phi_1$, which is marked by a black circle.

The second leg of the gait is then $-\phi(t)$. From Proposition 4, we know that it generates the following state transition matrix

$$\Phi_2 = \Lambda_1 \Phi_1 \Lambda_1 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & \sin\theta_0 & x_0(\frac{T}{2}) \\ -\sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & -y_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{42}$$

The total state transition matrix for Gait 1 is then

$$\Phi = \Phi_1 \Phi_2 = \begin{bmatrix} 1 & 0 & x_0(\frac{T}{2})(1+\cos\theta_0(\frac{T}{2})) + y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) \\ 0 & 1 & x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2})) \\ 0 & 0 & 1 \end{bmatrix}. \tag{43}$$

It yields no net rotation as discussed earlier.

Gait 2 ($\phi_2 = \phi_1$). Consider Gait 2 as shown in Fig. 5(b), whose two legs are symmetric with respect to the line $\phi_2 = \phi_1$. It starts from a point on the line $\phi_2 = \phi_1$, which is marked by a black circle.

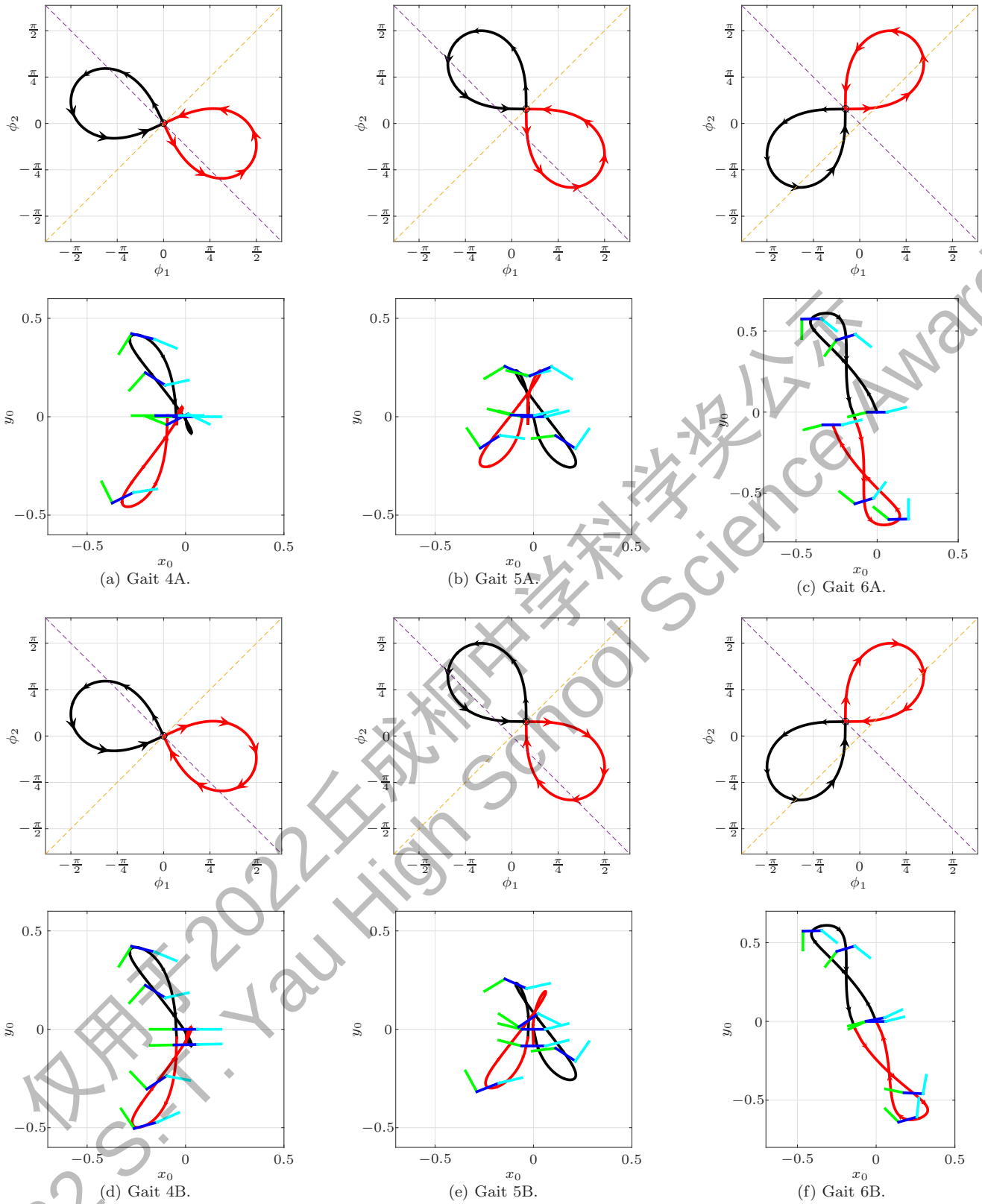
The second leg is $[\phi_2(\frac{T}{2} - t), \phi_1(\frac{T}{2} - t)]$, which is effectively the same as $[\phi_2(-t), \phi_1(-t)]$ from Proposition 1. Moreover, from Propositions 3 and 5, it yields the following state transition matrix :

$$\Phi_2 = \Lambda_2 \Phi_1^{-1} \Lambda_2 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & -\sin\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})\cos\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) \\ \sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) - y_0(\frac{T}{2})\cos\theta_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{44}$$

The total state transition matrix for Gait 2 is

$$\Phi = \Phi_1 \Phi_2 = \begin{bmatrix} \cos 2\theta_0(\frac{T}{2}) & -\sin 2\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})(1+\cos 2\theta_0(\frac{T}{2})) + y_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}) \\ \sin 2\theta_0(\frac{T}{2}) & \cos 2\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1-\cos 2\theta_0(\frac{T}{2})) \\ 0 & 0 & 1 \end{bmatrix}. \tag{45}$$

[1] All the other curved gaits in this section are obtained by rotating and translating lemniscates.



(a) Gait 4A. (b) Gait 5A. (c) Gait 6A.

(d) Gait 4B. (e) Gait 5B. (f) Gait 6B.

FIG. 6: (Color online) Symmetric gaits and their swimming trajectories. The 1st leg is in black and the 2nd in red. Gaits 4A & 4B are symmetric with respect to the origin, Gaits 5A & 5B to the line $\phi_2 = \phi_1$, and Gaits 6A & 6B to the line $\phi_2 = -\phi_1$. Gaits 4A, 5A, and 6A keep orientation the same in both legs, whereas Gaits 4B, 5B, and 6B change it in the 2nd leg.

It receives a net rotation of $2\theta_0(\frac{T}{2})$.

Gait 3 ($\phi_2 = -\phi_1$). Consider Gait 3 as shown in Fig. 5(c), whose two legs are symmetric with respect to the line $\phi_2 = -\phi_1$. It starts from a point on the line $\phi_2 = -\phi_1$, which is marked by a black circle.

The second segment is $[-\phi_2(\frac{T}{2} - t), -\phi_1(\frac{T}{2} - t)]$. From Propositions 3, 4, and 5, we know that the second leg generates the following state transition matrix

$$\Phi_2 = \Lambda_2\Lambda_1\Phi_1^{-1}\Lambda_1\Lambda_2 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & \sin\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})\cos\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) \\ -\sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})\cos\theta_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{46}$$

The total state transition matrix for Gait 3 is

$$\Phi = \Phi_1\Phi_2 = \begin{bmatrix} 1 & 0 & 2x_0(\frac{T}{2}) \\ 0 & 1 & 2y_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{47}$$

It generates no net rotation as discussed earlier.

Next we study some two-segment symmetric gaits that cross themselves once, *e.g.,* in the shape of figure 8. Assume that they start from the intersection point.

Gaits 4A & 4B (Origin). Consider Gaits 4A and 4B whose two legs are symmetric with respect to the origin as illustrated in Figs. 6(a) and 6(d), where 4A keeps and 4B changes the orientation in the second leg. Both gaits start from the origin.

For Gait 4A, the second leg is $-\phi(t)$. Therefore it is the same case as Gait 1, and gains no net rotation at the final time. Gait 4B changes the orientation in the second leg and it is then given by $-\phi(\frac{T}{2} - t)$. From Propositions 3 and 4, we know that the second leg generates the following state transition matrix

$$\Phi_2 = \Lambda_1\Phi_1^{-1}\Lambda_1 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & -\sin\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2})\cos\theta_0(\frac{T}{2}) - y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) \\ \sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})\cos\theta_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{48}$$

The total state transition matrix for Gait 4B is

$$\Phi = \Phi_1\Phi_2 = \begin{bmatrix} \cos 2\theta_0(\frac{T}{2}) & -\sin 2\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})(1 - \cos 2\theta_0(\frac{T}{2})) - y_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}) \\ \sin 2\theta_0(\frac{T}{2}) & \cos 2\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1 + \cos 2\theta_0(\frac{T}{2})) \\ 0 & 0 & 1 \end{bmatrix}. \tag{49}$$

Thus it gets a net rotation of $2\theta_0(\frac{T}{2})$.

Gaits 5A & 5B ($\phi_2 = \phi_1$). Consider Gaits 5A and 5B whose two legs are symmetric with respect to the line $\phi_2 = \phi_1$ as illustrated in Figs. 6(b) and 6(e), where 5A keeps and 5B changes the orientation. Both gaits start from a point on the line $\phi_2 = \phi_1$.

For Gait 5A, the second leg is $[\phi_2(\frac{T}{2} - t), \phi_1(\frac{T}{2} - t)]$. Therefore it is the same case as Gait 2, and it gets a net rotation of $2\theta_0(\frac{T}{2})$. Gait 5B changes the orientation in the second leg and thus it is $[\phi_2(t), \phi_1(t)]$. From Proposition 5, we know that the second leg generates the following state transition matrix

$$\Phi_2 = \Lambda_2\Phi_1\Lambda_2 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & \sin\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2}) \\ -\sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & y_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{50}$$

The total state transition matrix for Gait 5B is

$$\Phi = \Phi_1\Phi_2 = \begin{bmatrix} 1 & 0 & x_0(\frac{T}{2})(1 - \cos\theta_0(\frac{T}{2})) - y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) \\ 0 & 1 & -x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1 + \cos\theta_0(\frac{T}{2})) \\ 0 & 0 & 1 \end{bmatrix}. \tag{51}$$

Then it gets no net rotation.

Gait	**Symmetry wrt**	**Orient. change**	**second leg**	Φ_2	$\theta_0(T)$	**Final position** $[x_0(T), y_0(T)]$
1, 4A	origin	No	$[-\phi_1(t), -\phi_2(t)]$	$\Lambda_1\Phi_1\Lambda_1$	0	$\big[x_0(\frac{T}{2})(1+\cos\theta_0(\frac{T}{2})) + y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}),$ $x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2}))\big]$
4B	origin	Yes	$[-\phi_1(-t), -\phi_2(-t)]$	$\Lambda_1\Phi_1^{-1}\Lambda_1$	$2\theta_0(\frac{T}{2})$	$\big[x_0(\frac{T}{2})(1-\cos 2\theta_0(\frac{T}{2})) - y_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}),$ $-x_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1+\cos 2\theta_0(\frac{T}{2}))\big]$
2, 5A	$\phi_2 = \phi_1$	No	$[\phi_2(-t), \phi_1(-t)]$	$\Lambda_2\Phi_1^{-1}\Lambda_2$	$2\theta_0(\frac{T}{2})$	$\big[x_0(\frac{T}{2})(1+\cos 2\theta_0(\frac{T}{2})) + y_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}),$ $x_0(\frac{T}{2})\sin 2\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1-\cos 2\theta_0(\frac{T}{2}))\big]$
5B	$\phi_2 = \phi_1$	Yes	$[\phi_2(t), \phi_1(t)]$	$\Lambda_2\Phi_1\Lambda_2$	0	$\big[x_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2})) - y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}),$ $-x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1+\cos\theta_0(\frac{T}{2}))\big]$
3, 6A	$\phi_2 = -\phi_1$	No	$[-\phi_2(-t), -\phi_1(-t)]$	$\Lambda_2\Lambda_1\Phi_1^{-1}\Lambda_1\Lambda_2$	0	$\big[2x_0(\frac{T}{2}), 2y_0(\frac{T}{2})\big]$
6B	$\phi_2 = -\phi_1$	Yes	$[-\phi_2(t), -\phi_1(t)]$	$\Lambda_2\Lambda_1\Phi_1\Lambda_1\Lambda_2$	$2\theta_0(\frac{T}{2})$	$\big[x_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2})) + y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}),$ $-x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2}))\big]$

TABLE I: Symmetry gaits 1–6B and their properties.

Gaits 6A & 6B ($\phi_2 = -\phi_1$). Consider Gaits 6A and 6B whose two legs are symmetric with respect to the line $\phi_2 = -\phi_1$ as illustrated in Figs. 6(c) and 6(f), where 6A keeps and 6B changes the orientation. Both gaits start from a point on the line $\phi_2 = -\phi_1$.

For Gait 6A, the second leg is $[-\phi_2(\frac{T}{2}-t), -\phi_1(\frac{T}{2}-t)]$. Therefore it is the same case as Gait 3, and gets no net rotation. Gait 6B changes the orientation in the second leg and then it is $[-\phi_2(t), -\phi_1(t)]$. From Proposition 5, we know that the second leg generates the following state transition matrix

$$\Phi_2 = \Lambda_2\Lambda_1\Phi_1\Lambda_1\Lambda_2 = \begin{bmatrix} \cos\theta_0(\frac{T}{2}) & -\sin\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2}) \\ \sin\theta_0(\frac{T}{2}) & \cos\theta_0(\frac{T}{2}) & -y_0(\frac{T}{2}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{52}$$

The total state transition matrix for Gait 6B is

$$\Phi = \Phi_1\Phi_2 = \begin{bmatrix} \cos 2\theta_0(\frac{T}{2}) & -\sin 2\theta_0(\frac{T}{2}) & x_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2})) + y_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) \\ \sin 2\theta_0(\frac{T}{2}) & \cos 2\theta_0(\frac{T}{2}) & -x_0(\frac{T}{2})\sin\theta_0(\frac{T}{2}) + y_0(\frac{T}{2})(1-\cos\theta_0(\frac{T}{2})) \\ 0 & 0 & 1 \end{bmatrix}. \tag{53}$$

Then it gets a net rotation of $2\theta_0(\frac{T}{2})$.

Compiling all these results together, we obtain Table I. Moreover, we conclude the following proposition.

Proposition 6 A two-leg gait $\phi(t)$ yields no net rotation if

- its two legs are symmetric with respect to the origin, and the second leg keeps the same orientation as the first one; or
- its two legs are symmetric with respect to the line $\phi_2 = \phi_1$, and the second leg changes the orientation from the first one; or
- its two legs are symmetric with respect to the line $\phi_2 = -\phi_1$, and the second leg keeps the same orientation as the first one.

B. Double symmetric gaits

Now we look into the gaits that are symmetric with respect to both the origin and the line $\phi_2 = \phi_1$, which in turn implies that they are symmetric to $\phi_2 = -\phi_1$ as well. Consider the gaits possessing four segments, and each segment evolves on the time interval $[0, \frac{T}{4}]$ at the same speed. Denote the gait for the first leg as $\phi(t) = [\phi_1(t), \phi_2(t)]$ and the state transition matrix at the end of the first leg (*i.e.*, $t = \frac{T}{4}$) as

$$\Phi_1 = \begin{bmatrix} \cos\theta_0(\frac{T}{4}) & -\sin\theta_0(\frac{T}{4}) & x_0(\frac{T}{4}) \\ \sin\theta_0(\frac{T}{4}) & \cos\theta_0(\frac{T}{4}) & y_0(\frac{T}{4}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{54}$$

Gait 7. Consider the square Gait 7 as shown in Fig. 7(a), which is also presented in Fig. 3 and originally proposed by Purcell [12]. The starting point is on the line $\phi_2 = \phi_1$, which is marked by a black circle.

The second leg is $[-\phi_2(\frac{T}{4} - t), -\phi_1(\frac{T}{4} - t)]$, the third $[-\phi_1(t), -\phi_2(t)]$, and the fourth $[\phi_2(\frac{T}{4} - t), \phi_1(\frac{T}{4} - t)]$. From Propositions 4 and 5, we know these three legs generate the state transition matrices as

$$\begin{aligned} \Phi_2 &= \Lambda_1\Lambda_2\Phi_1^{-1}\Lambda_2\Lambda_1, \\ \Phi_3 &= \Lambda_1\Phi_1\Lambda_1, \\ \Phi_4 &= \Lambda_2\Phi_1^{-1}\Lambda_2. \end{aligned} \tag{55}$$

The complete state transition matrix for Gait 7 can be calculated as

$$\Phi = \Phi_1\Phi_2\Phi_3\Phi_4 = \begin{bmatrix} 1 & 0 & 4x_0(\frac{T}{4}) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{56}$$

Therefore, this gait generates a pure x_0-axis displacement with no net rotation.

Gaits 8A & 8B. Consider the gaits in Figs. 7(b) and 7(d). Both start from the origin. For Gait 8A, the second leg is $[-\phi_2(\frac{T}{4} - t), -\phi_1(\frac{T}{4} - t)]$, the third $[-\phi_1(t), -\phi_2(t)]$, and the fourth $[\phi_2(\frac{T}{4} - t), \phi_1(\frac{T}{4} - t)]$. This is indeed the same as Gait 7, which generates a pure x_0 displacement.

For Gait 8B, the second leg is also $[-\phi_2(\frac{T}{4} - t), -\phi_1(\frac{T}{4} - t)]$. However, it changes orientation in leg 3 and leg 4. The third leg is then $[\phi_2(t), \phi_1(t)]$, and the fourth $[-\phi_1(\frac{T}{4} - t), -\phi_2(\frac{T}{4} - t)]$. From Propositions 4 and 5, we know these three legs generate the state transition matrices as

$$\begin{aligned} \Phi_2 &= \Lambda_1\Lambda_2\Phi_1^{-1}\Lambda_2\Lambda_1, \\ \Phi_3 &= \Lambda_2\Phi_1\Lambda_2, \\ \Phi_4 &= \Lambda_1\Phi_1^{-1}\Lambda_1. \end{aligned} \tag{57}$$

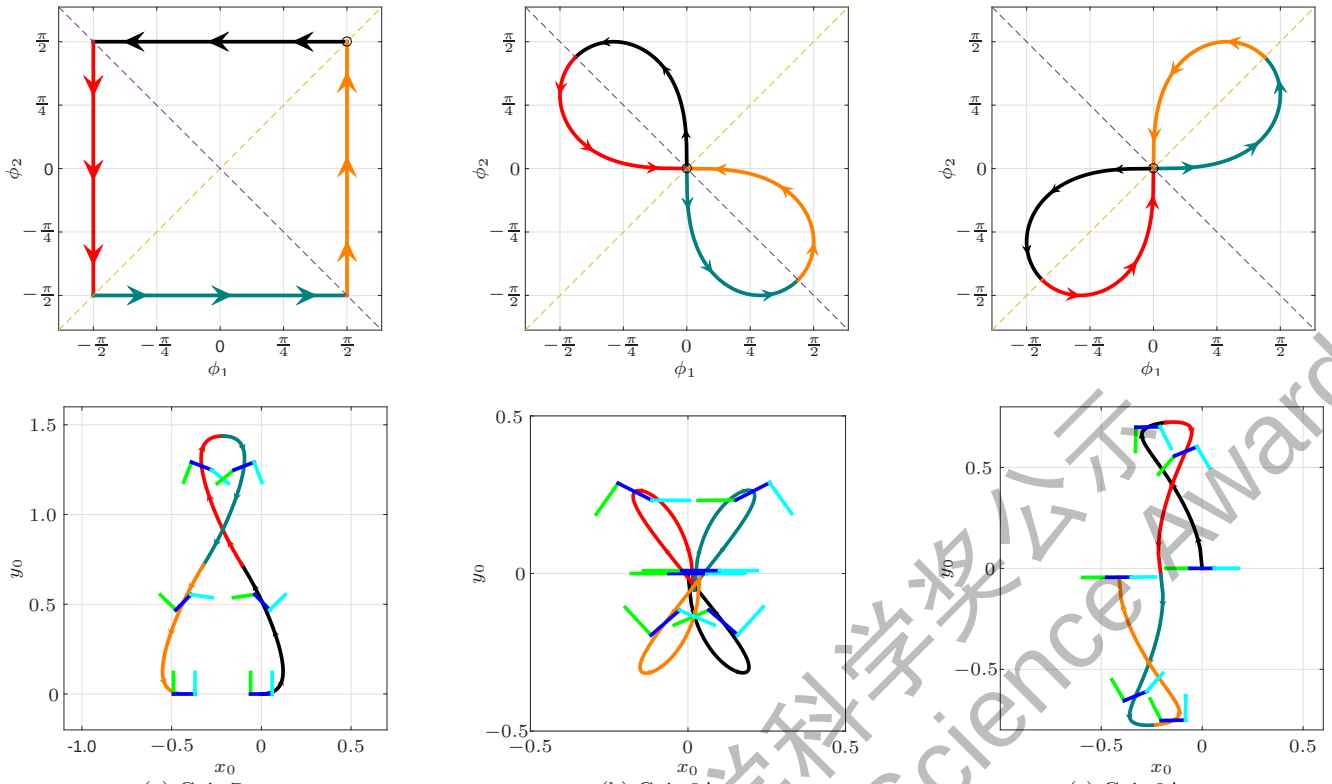
The complete state transition matrix for Gait 8B is calculated as

$$\Phi = \Phi_1\Phi_2\Phi_3\Phi_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 4y_0(\frac{T}{4}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{58}$$

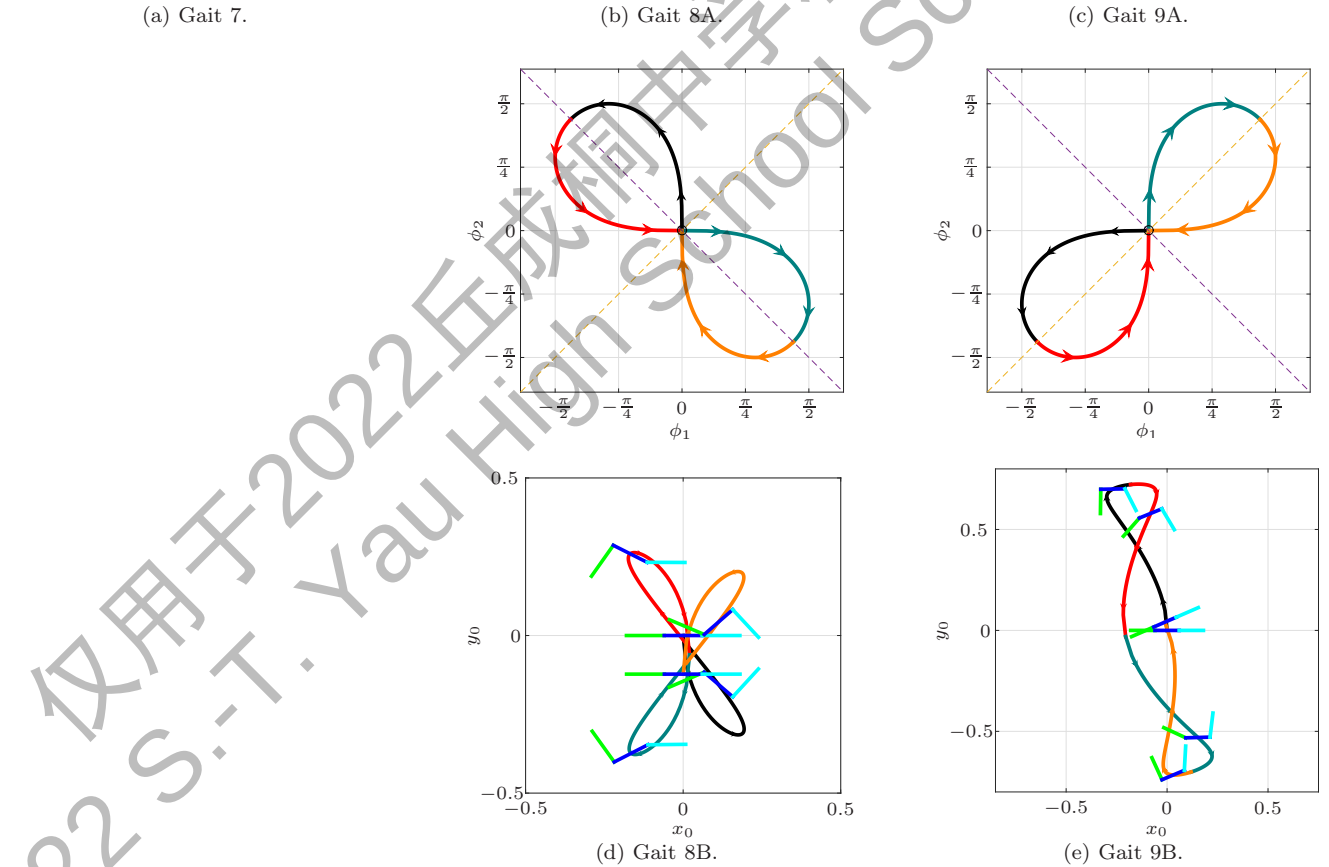
Therefore, this gait generates a pure y_0-axis displacement with no net rotation.

Gaits 9A & 9B. Consider the gaits in Figs. 7(c) and 7(e). Both start from the origin. For Gait 9A, the second leg is $[\phi_2(\frac{T}{4} - t), \phi_1(\frac{T}{4} - t)]$, the third $[-\phi_1(t), -\phi_2(t)]$, and the fourth $[-\phi_2(\frac{T}{4} - t), -\phi_1(\frac{T}{4} - t)]$. From Propositions 4 and 5, we know these three legs generate the state transition matrices as

$$\begin{aligned} \Phi_2 &= \Lambda_2\Phi_1^{-1}\Lambda_2, \\ \Phi_3 &= \Lambda_1\Phi_1\Lambda_1, \\ \Phi_4 &= \Lambda_1\Lambda_2\Phi_1^{-1}\Lambda_2\Lambda_1. \end{aligned} \tag{59}$$



(a) Gait 7. (b) Gait 8A. (c) Gait 9A.



(d) Gait 8B. (e) Gait 9B.

FIG. 7: (Color online) These gaits are symmetric with respect to both the origin and $\phi_2 = \phi_1$ and have four legs. The first leg is in black, the second in red, the third in dark green, and the fourth in orange. The starting point is marked by a black circle.

Gait	Change Orient.	Legs 2, 3, and 4	$\theta_0(T)$	Final position $[x_0(T), y_0(T)]$
7, 8A	No	$\Phi_2 = \Lambda_1\Lambda_2\Phi_1^{-1}\Lambda_2\Lambda_1$, $\Phi_3 = \Lambda_1\Phi_1\Lambda_1$, $\Phi_4 = \Lambda_2\Phi_1^{-1}\Lambda_2$	0	$\left[4x_0(\frac{T}{4}), 0\right]$
8B	Yes	$\Phi_2 = \Lambda_1\Lambda_2\Phi_1^{-1}\Lambda_2\Lambda_1$, $\Phi_3 = \Lambda_2\Phi_1\Lambda_2$, $\Phi_4 = \Lambda_1\Phi_1^{-1}\Lambda_1$	0	$\left[0, 4y_0(\frac{T}{4})\right]$
9A	No	$\Phi_2 = \Lambda_2\Phi_1^{-1}\Lambda_2$, $\Phi_3 = \Lambda_1\Phi_1\Lambda_1$, $\Phi_4 = \Lambda_1\Lambda_2\Phi_1^{-1}\Lambda_2\Lambda_1$	0	$\big[2x_0(\frac{T}{4})(1+\cos 2\theta_0(\frac{T}{4})) + 2y_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4}),$ $2y_0(\frac{T}{4})(1-\cos 2\theta_0(\frac{T}{4})) + 2x_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4})\big]$
9B	Yes	$\Phi_2 = \Lambda_2\Phi_1^{-1}\Lambda_2$, $\Phi_3 = \Lambda_2\Lambda_1\Phi_1\Lambda_1\Lambda_2$, $\Phi_4 = \Lambda_1\Phi_1^{-1}\Lambda_1$	$4\theta_0(\frac{T}{4})$	$\big[2(1-\cos 2\theta_0(\frac{T}{4}))(x_0(\frac{T}{4}) + x_0(\frac{T}{4})\cos 2\theta_0(\frac{T}{4}) + y_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4})),$ $-2\cos 2\theta_0(\frac{T}{4})(y_0(\frac{T}{4}) - y_0(\frac{T}{4})\cos 2\theta_0(\frac{T}{4}) + x_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4}))\big]$

TABLE II: Symmetric gaits 7–9B and their properties.

The complete state transition matrix for Gait 9A is calculated as

$$\Phi = \Phi_1\Phi_2\Phi_3\Phi_4 = \begin{bmatrix} 1 & 0 & 2x_0(\frac{T}{4})(1+\cos 2\theta_0(\frac{T}{4})) + 2y_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4}) \\ 0 & 1 & 2y_0(\frac{T}{4})(1-\cos 2\theta_0(\frac{T}{4})) + 2x_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4}) \\ 0 & 0 & 1 \end{bmatrix}. \tag{60}$$

For Gait 9B, the second leg is $[\phi_2(\frac{T}{4}-t), \phi_1(\frac{T}{4}-t)]$, the third $[-\phi_2(t), -\phi_1(t)]$, and the fourth $[-\phi_1(\frac{T}{4}-t), -\phi_2(\frac{T}{4}-t)]$. From Propositions 4 and 5, we know these three legs generate the state transition matrices as

$$\begin{aligned} \Phi_2 &= \Lambda_2\Phi_1^{-1}\Lambda_2, \\ \Phi_3 &= \Lambda_2\Lambda_1\Phi_1\Lambda_1\Lambda_2, \\ \Phi_4 &= \Lambda_1\Phi_1^{-1}\Lambda_1. \end{aligned} \tag{61}$$

The complete state transition matrix for four legs is calculated as

$$\Phi = \Phi_1\Phi_2\Phi_3\Phi_4 = \begin{bmatrix} \cos 4\theta_0(\frac{T}{4}) & -\sin 4\theta_0(\frac{T}{4}) & 2(1-\cos 2\theta_0(\frac{T}{4}))(x_0(\frac{T}{4}) + x_0(\frac{T}{4})\cos 2\theta_0(\frac{T}{4}) + y_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4})) \\ \sin 4\theta_0(\frac{T}{4}) & \cos 4\theta_0(\frac{T}{4}) & -2\cos 2\theta_0(\frac{T}{4})(y_0(\frac{T}{4}) - y_0(\frac{T}{4})\cos 2\theta_0(\frac{T}{4}) + x_0(\frac{T}{4})\sin 2\theta_0(\frac{T}{4})) \\ 0 & 0 & 1 \end{bmatrix}. \tag{62}$$

These properties of Gaits 7-9B and their final orientations and positions are summarized in Table II.

V. STEERING THE MICROSWIMMER

In the previous section, we have looked into a collection of symmetric gaits that can drive the three-link microswimmer to certain final posture. In this section, we study the converse problem, that is, finding a gait that steers the microswimmer to some desired final posture.

A. A constructive approach

We begin with a simple constructive approach to solve this problem. To steer the microswimmer to a desired location, we can divide the task into a series of movements along x_0 or y_0 directions. For example, if we want to start the swimmer from the origin and move it to the target location $[x_0^d, y_0^d]$ marked as a red dot in Fig. 8, we can first move it along x_0 direction to $[x_0^d, 0]$ and then along y_0 direction to $[x_0^d, y_0^d]$, or first along y_0 direction to $[0, y_0^d]$ and then along x_0 direction to $[x_0^d, y_0^d]$. Of course, there are infinitely many such paths consisting of a series of pure x_0 and y_0 movements. It is clear that if we can design gaits to move any arbitrary distance along the x_0- and y_0-axis, we can achieve anywhere in the plane.

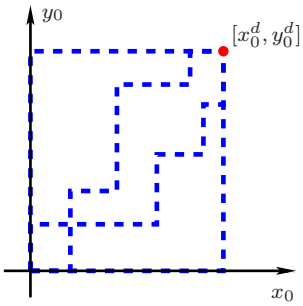
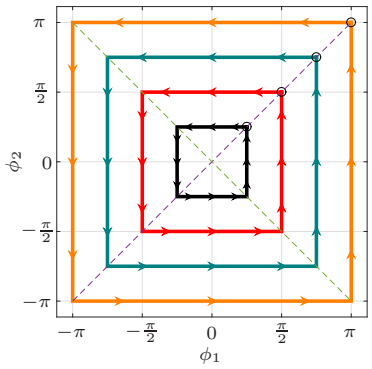
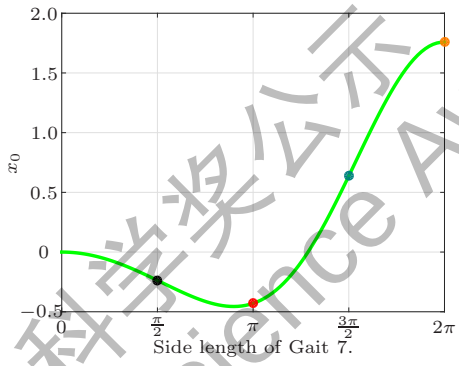


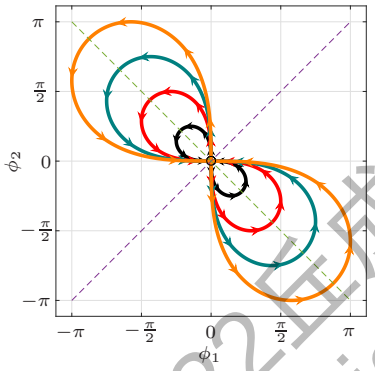
FIG. 8: Different paths to achieve the target location $[x_0^d, y_0^d]$.



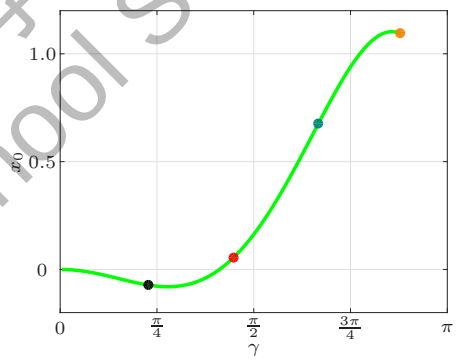
(a) Gait 7 with different side lengths.



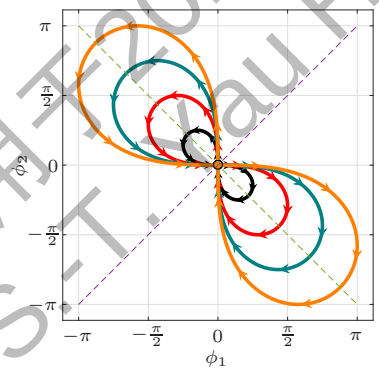
(b) Final x_0 achieved. Min: -0.4559; Max: 1.7578.



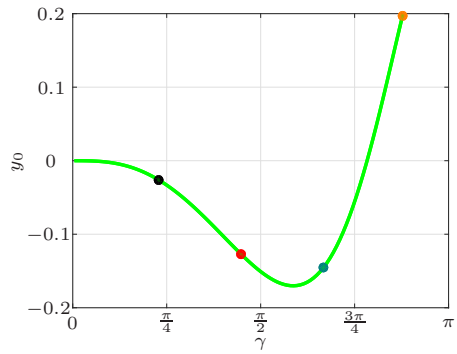
(c) Gait 8A with different parameter γ.



(d) Final x_0 achieved. Min: -0.0800; Max: 1.1026.



(e) Gait 8B with different parameter γ.



(f) Final y_0 achieved. Min: -0.1702; Max: 0.1969.

FIG. 9: (Color online) The final x_0 or y_0 location as a function of gait amplitude. The colored dot in the right figure indicates the final location achieved by the gait in the same color in the left.

From Table II, we observe that both Gaits 7 and 8A steer the microswimmer to a final location on the x_0-axis, and Gait 8B moves it on the y_0-axis. We just need to find appropriate amplitudes for these gaits to achieve a desired x_0 or y_0 movement.

For Gait 7, we vary the side length of the square to obtain a series of similar gaits as shown in Fig. 9(a). The final achieved positions on the x_0-axis are plotted as a function of the side length in Fig. 9(b), where the black, red, dark green, and orange dots represent the final x_0 position achieved by the gait in the same color in Fig. 9(a). For the side length in $[0, 2\pi]$, we get from Fig. 9(b) that the attained range for x_0 is $[-0.4559, 1.7578]$. To steer the microswimmer to an x_0^d within this range, we can read out a side length corresponding to x_0^d from Fig. 9(b) and then apply Gait 7 with this side length. Note that for negative value of x_0^d, there exist two different feasible side lengths and we can choose an appropriate one. If the desired x_0^d is outside this range, we can repeatedly apply Gait 7.

Gaits 8A and 8B are pieced together by four branches of lemniscate curves whose analytic expressions are given in Eq. (41). We can vary the parameter γ therein to obtain a series of similar gaits with different amplitudes as shown in Figs. 9(c) and 9(e). The final achieved locations on the x_0-axis and y_0-axis are plotted as functions of γ in Figs. 9(d) and 9(f), where the colored dots represent the final position achieved by the gait in the same color in the left figures. The achieved x_0 range from Gait 8A is $[-0.0800, 1.1026]$, and y_0 range from Gait 8B is $[-0.1701, 0.1969]$. The same technique discussed for Gait 7 above can be used for Gait 8A to achieve a desired x_0^d location and for Gait 8B to a desired y_0^d location. Combining these three gaits, the microswimmer can move to any location in the plane.

Comparing Gaits 7 and 8A, we notice that Gait 7 steers the microswimmer along x_0-axis farther than 8A, provided that they have similar amplitudes. Moreover, Figs. 9(d) and 9(f) show that it is easier for the microswimmer to move along x_0 than y_0 direction.

B. Optimization approach and Automatic Differentiation

We consider an optimization approach to solve the steering of microswimmer, that is, formulating it as an optimization problem. Some early results were reported in [8, 11, 14, 17]. In the current work, we will use the gradient descent algorithm to find steering gaits and apply symmetry properties studied in preceding sections. In particular, we will utilize the Automatic Differentiation (AutoDiff) technique to compute the gradients efficiently.

Suppose that the desired posture is $[x_0^d, y_0^d, \theta_0^d]^T$, and the actual posture achieved by the microswimmer at the final time is $[x_0(T), y_0(T), \theta_0(T)]^T$. We can use a quadratic function to quantify the difference between these two postures:

$$J_0(\phi(t)) = \frac{1}{2}\left(x_0(T) - x_0^d\right)^2 + \frac{1}{2}\left(y_0(T) - y_0^d\right)^2 + \frac{1}{2}\left(\theta_0(T) - \theta_0^d\right)^2. \tag{63}$$

We take J_0 as the cost function (also known as penalty function) of an optimization problem, and the gait $\phi(t)$ as the optimization variable. Different gait $\phi(t)$ yields different final posture, and thus results in different cost. Our objective is to find a gait $\phi(t)$ that minimizes this cost function. In the ideal case when $J_0 = 0$, we obtain a gait that can drive the swimmer to the desired posture exactly.

The steering of microswimmer can then be formulated as the following optimization problem:

$$\begin{aligned} &\min_{\phi(t)} J_0(\phi(t)), \\ \text{subject to} \quad &\dot{p}_0 = \frac{1}{d_1} R(\theta_0) G_v \dot{\phi}, \\ &\dot{\theta}_0 = \frac{1}{d_1} G_\omega \dot{\phi}. \end{aligned} \tag{64}$$

The optimization variable $\phi(t)$ is a continuous function, and in general it is difficult to optimize over functions. Hence we consider the Fourier expansion of $\phi(t)$ on $[0, T]$:

$$\begin{aligned} \phi_1(t) &= a_0 + \sum\nolimits_{n=1}^{N} (a_n \cos n\omega t + b_n \sin n\omega t), \\ \phi_2(t) &= c_0 + \sum\nolimits_{n=1}^{N} (c_n \cos n\omega t + d_n \sin n\omega t), \end{aligned} \tag{65}$$

where $\omega = 2\pi/T$. Note that the expansion is truncated at the order of N, and the conditions $\phi_1(0) = \phi_1(T)$ and $\phi_2(0) = \phi_2(T)$ are automatically satisfied. The gait functions $\phi_1(t)$ and $\phi_2(t)$ are completely characterized by the expansion coefficients, which can form a vector:

$$\mu = [a_0, a_1, b_1, \cdots, a_N, b_N, c_0, c_1, d_1, \cdots, c_N, d_N]^T. \tag{66}$$

The search for a steering gait is converted to find an appropriate coefficient vector μ^* that minimizes the cost function J_0.

There exist many optimization algorithms to solve this problem. In this work, we adopt gradient descent algorithm, which is a simple yet widely used optimization algorithm. Gradient algorithm is an iterative procedure. At each step, we compute the gradient of the cost function with respect to the optimization variables. This gives a direction to reduce the cost function, which can be used to update the current solution to a better one. The procedure of the gradient descent algorithm can be listed as follows.

1. Choose an initial guess of the Fourier coefficient vector μ^0;
2. At the k-th iteration, compute $J_0(\mu^k)$ and the gradient $\nabla^T_{\mu^k} J_0(\mu^k)$;
3. Update the coefficient vector by $\mu^{k+1} = \mu^k - \epsilon \nabla^T_{\mu^k} J_0(\mu^k)$, where ϵ is a chosen small positive number;
4. Repeat 2-3 until a desired convergence is reached.

The implementation of gradient algorithm is straightforward, and the key step is to compute the gradient $\nabla^T_{\mu^k} J_0(\mu^k)$. There exist multiple methods to do this computation. The most common one is finite difference; however, it has intrinsic approximating errors and may result in numerical instability. Another method is symbolic differentiation, which can lead to inefficient codes and is usually difficult to deal with computation programs.

Automatic differentiation (AutoDiff) technique overcomes these difficulties and have made significant progresses in recent years [2]. In general, a function can be calculated by executing a series of basic operations ($+$, $-$, $*$, $/$) and elementary functions such as exp, log, sin, cos, etc. The basic idea of AutoDiff is to automatically analyze and track this series of operations and functions, and then to repeatedly apply the chain rule to compute the derivatives. There are several AutoDiff tools available, including ADOL-C [15], Google TensorFlow [6], and Matlab Deep Learning Toolbox. In our case, we need to take the derivative of final posture $[x_0(T), y_0(T), \theta_0(T)]$, which is the solution of differential equation (64) at the final time. We will use Matlab AutoDiff tools to calculate the gradients because it requires only minimum programming efforts[2]. The detailed code is included in Appendix B.

Without imposing any constraints on gait symmetry for the time being, we apply gradient descent algorithm to investigate the following two steering problems.

Example 3 Consider to steer the microswimmer from an initial position $[0, 0]$ to a final position $[-0.5, 0.2]$. The initial orientation $\theta_0(0)$ is set at 0, and the final orientation $\theta_0(T)$ is left free. Set the truncation order $N = 5$. The gradient algorithm yields two different results as shown in Fig. 10: (a) plots the two obtained gaits, (b) the corresponding swimming trajectories, and (c) how the cost functions decrease versus iteration number. The difference between these two results is that they start from different initial guesses. For the top row, the initial gait is chosen to be

$$\phi_1(t) = 0.5 \cos \omega t, \ \phi_2(t) = 0.5 \sin \omega t,$$

and thus the initial coefficient vector is

$$\mu^0 = [0,\ 0.5,\ \underbrace{0,\ \cdots,\ 0}_{2N-1},\ 0,\ 0,\ 0.5,\ \underbrace{0,\ \cdots,\ 0}_{2N-2}]^T.$$

For the bottom row, it is

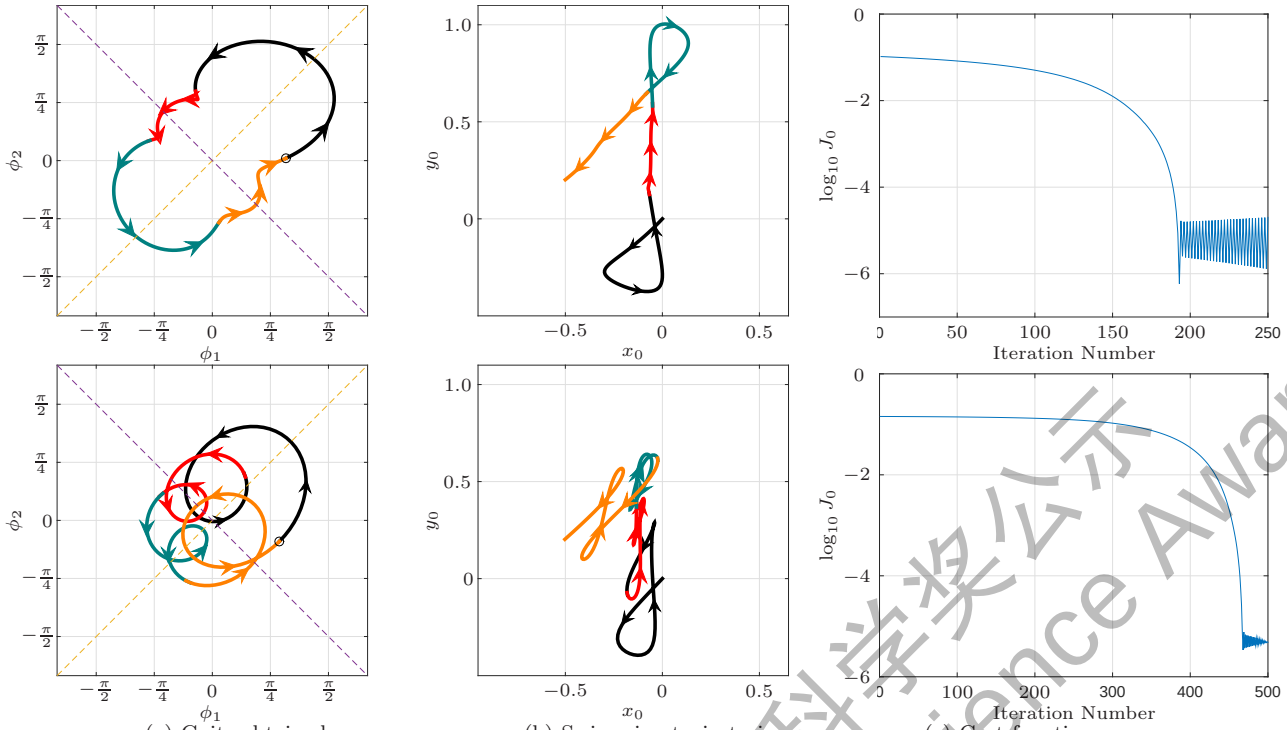
$$\phi_1(t) = 0.1 \cos \omega t, \ \phi_2(t) = 0.1 \sin \omega t,$$

and thus

$$\mu^0 = [0,\ 0.1,\ \underbrace{0,\ \cdots,\ 0}_{2N-1},\ 0,\ 0,\ 0.1,\ \underbrace{0,\ \cdots,\ 0}_{2N-2}]^T.$$

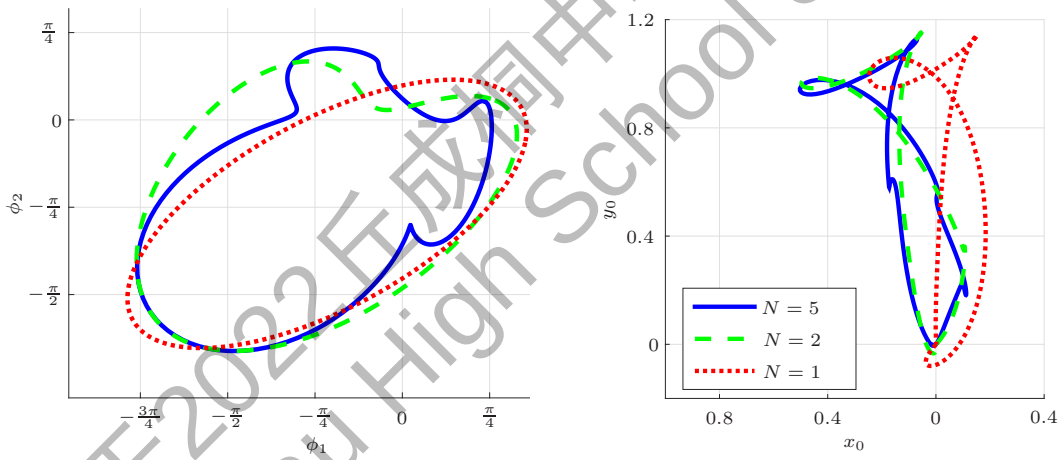
It can be seen that both gaits steer the microswimmer to the desired location $[-0.5, 0.2]$ with high accuracy; however, these two gaits look very different. This is because gradient descent algorithm is a local method, and the optimization result highly depends on the initial guess.

[2] The procedure of solving differentiation equation for AutoDiff can be done by Matlab command `dlode45`, which is available only in Matlab version 2021b and later.



(a) Gaits obtained. (b) Swimming trajectories. (c) Cost function convergence.

FIG. 10: (Color online) Gaits obtained from gradient descent algorithm that steer the microswimmer from $[0, 0]$ to $[-0.5, 0.2]$.



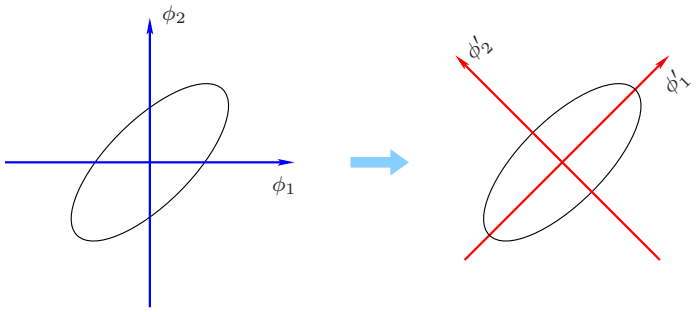
(a) Gaits obtained. (b) Swimming trajectories.

FIG. 11: (Color online) Gaits to achieve a net rotation of $\frac{\pi}{4}$.

Example 4 Consider to achieve a pure rotation, *e.g.*, starting from an initial posture $[0, 0, 0]$ to a final one $[0, 0, \frac{\pi}{4}]$. Set the initial gait as

$$\phi_1(t) = \cos \omega t, \ \ \phi_2(t) = \sin \omega t.$$

Fig. 11(a) shows the obtained gaits when the truncation order N is set to be 1, 2, and 5, and (b) plots the corresponding swimming trajectories. All of these three gaits generate swimming trajectories that can achieve the desired pure rotation with high accuracy. However, we observe that larger truncation order N yields more curved gaits with cusps, because larger N results in more high frequency components in the gait $\phi(t)$.



(a) Coordinate system $\phi_1\phi_2$ (b) Coordinate system $\phi_1'\phi_2'$

FIG. 12: A new coordinate system $\phi_1'\phi_2'$ obtained by rotating $\phi_1\phi_2$ for $\pi/4$ counterclockwise.

C. Simplified Fourier expansions for symmetric gaits

Examples 3 and 4 manifest that there may exist many different gaits that can achieve the same final posture. We now consider to steer with symmetric gaits. The symmetry in the geometric shape of a gait imposes certain constraints on its Fourier expansion coefficients and can considerably simplify them. This reduces the dimension of optimization variables and consequently alleviates the computational load of the optimization procedure. We hereby derive simplified Fourier expansions for the symmetric gaits investigated in Sec. IV.

Take Gaits 7 and 8A for example. They are symmetric with respect to both the lines $\phi_2 = \phi_1$ and $\phi_2 = -\phi_1$. For simplicity, we represent the gait in a new coordinate system $\phi_1'\phi_2'$ defined by these two lines, as shown in Fig. 12(b). This way, the symmetry with respect to the line $\phi_2 = \phi_1$ becomes mirror image to the horizontal ϕ_1' axis, and the symmetry to the line $\phi_2 = -\phi_1$ is mirror image to the vertical ϕ_2' axis. The coordinates of the gait in the original $\phi_1\phi_2$ frame can be obtained by $\phi(t) = R(\frac{\pi}{4})\phi'(t)$.

Suppose that the Fourier expansions for Gaits 7 and 8A in the $\phi_1'\phi_2'$ frame are given by [3]

$$\begin{aligned}\phi_1'(t) &= a_0 + \sum\nolimits_{n=1}^{N}(a_n\cos n\omega t + b_n \sin n\omega t),\\ \phi_2'(t) &= c_0 + \sum\nolimits_{n=1}^{N}(c_n\cos n\omega t + d_n \sin n\omega t),\end{aligned} \tag{67}$$

where $t \in [0, T]$. We then consider leg 2, leg 3, and leg 4.

Leg 2. From Fig. 7(a), for $t \in [\frac{T}{4}, \frac{T}{2}]$, we have $\phi_1'(t) = -\phi_1'(\frac{T}{2} - t)$ and $\phi_2'(t) = \phi_2'(\frac{T}{2} - t)$. Because $n\omega\left(\frac{T}{2} - t\right) = n\pi - n\omega t$, we get

$$\begin{aligned}-\phi_1'\left(\frac{T}{2} - t\right) &= -a_0 - \sum\nolimits_{n=1}^{N}(a_n\cos(n\pi - n\omega t) + b_n\sin(n\pi - n\omega t))\\ &= -a_0 + \sum\nolimits_{n=1}^{N}((-1)^{n+1}a_n\cos n\omega t + (-1)^n b_n \sin n\omega t),\\ \phi_2'\left(\frac{T}{2} - t\right) &= c_0 + \sum\nolimits_{n=1}^{N}(c_n\cos(n\pi - n\omega t) + d_n\sin(n\pi - n\omega t))\\ &= c_0 + \sum\nolimits_{n=1}^{N}((-1)^n c_n\cos n\omega t + (-1)^{n+1} d_n \sin n\omega t).\end{aligned} \tag{68}$$

Equating the corresponding coefficients in Eq. (68) with those in (67), we obtain that

$$a_0 = 0, \quad a_{\text{even}} = 0, \quad b_{\text{odd}} = 0, \quad c_{\text{odd}} = 0, \quad d_{\text{even}} = 0. \tag{69}$$

Leg 3. For $t \in [\frac{T}{2}, \frac{3T}{4}]$, we have $\phi_1'(t) = -\phi_1'(t - \frac{T}{2})$ and $\phi_2'(t) = -\phi_2'(t - \frac{T}{2})$. Then $n\omega\left(t - \frac{T}{2}\right) = n\omega t - n\pi$. We

[3] With a slight abuse of notation, we use same symbols for Fourier expansion coefficients as in Eq. (65).

get

$$\begin{aligned}
-\phi_1'\left(t-\frac{T}{2}\right) &= -a_0 - \sum\nolimits_{n=1}^{N}(a_n\cos(n\omega t - n\pi) + b_n\sin(n\omega t - n\pi)) \\
&= -a_0 + \sum\nolimits_{n=1}^{N}((-1)^{n+1}a_n\cos n\omega t + (-1)^{n+1}b_n\sin n\omega t), \\
-\phi_2'\left(t-\frac{T}{2}\right) &= -c_0 - \sum\nolimits_{n=1}^{N}(c_n\cos(n\omega t - n\pi) + d_n\sin(n\omega t - n\pi) \\
&= -c_0 + \sum\nolimits_{n=1}^{N}((-1)^{n+1}c_n\cos n\omega t + (-1)^{n+1}d_n\sin n\omega t.
\end{aligned} \tag{70}$$

Equating the corresponding coefficients in Eq. (70) with those in (67), we get

$$a_0 = 0, \quad a_{\text{even}} = 0, \quad b_{\text{even}} = 0, \quad c_0 = 0, \quad c_{\text{even}} = 0, \quad d_{\text{even}} = 0. \tag{71}$$

Leg 4. For $t \in [\frac{3T}{4}, T]$, we have $\phi_1'(t) = \phi_1'(T-t)$ and $\phi_2'(t) = -\phi_2'(T-t)$. Then $n\omega\,(T-t) = 2n\pi - n\omega t$. We have

$$\begin{aligned}
\phi_1'\,(T-t) &= a_0 + \sum\nolimits_{n=1}^{N}(a_n\cos(2n\pi - n\omega t) + b_n\sin(2n\pi - n\omega t)) \\
&= a_0 + \sum\nolimits_{n=1}^{N}(a_n\cos n\omega t - b_n\sin n\omega t), \\
-\phi_2'\,(T-t) &= -c_0 - \sum\nolimits_{n=1}^{N}(c_n\cos(2n\pi - n\omega t) + d_n\sin(2n\pi - n\omega t) \\
&= -c_0 + \sum\nolimits_{n=1}^{N}(-c_n\cos n\omega t + d_n\sin n\omega t).
\end{aligned} \tag{72}$$

Equating the corresponding coefficients in Eq. (72) with those in (67), we get

$$b_n = 0, \quad c_0 = 0, \quad c_n = 0. \tag{73}$$

Combining Eqs. (69), (71), and (73), we obtain

$$a_0 = 0, \quad a_{\text{even}} = 0, \quad b_n = 0, \quad c_0 = 0, \quad c_n = 0, \quad d_{\text{even}} = 0. \tag{74}$$

Hence the Fourier expansion of Gaits 7 and 8A can be written as

$$\begin{aligned}
\phi_1'(t) &= \sum\nolimits_{n=1,\text{odd}}^{N} a_n\cos n\omega t, \\
\phi_2'(t) &= \sum\nolimits_{n=1,\text{odd}}^{N} d_n\sin n\omega t.
\end{aligned} \tag{75}$$

In addition, since Gait 8A starts from the origin, it has to satisfy an extra condition

$$\phi_1'(0) = \sum\nolimits_{n=1,\text{odd}}^{N} a_n = 0.$$

We perform similar analyses to all the other symmetric gaits, and obtain their corresponding simplified Fourier expansions as shown in Table III. For single symmetric gaits 1–6B, either the even or odd coefficients are eliminated, or the entire sine or cosine terms. This effectively cuts the number of Fourier expansion coefficients by half. For double symmetric gaits 7–9B, the expansion expressions are even more simplified and the coefficient numbers can be brought down to a quarter. This can greatly alleviate the computational load for the optimization procedure.

D. Steering by symmetric gaits

With the simplified Fourier expansion derived above, we consider to steer the microswimmer by applying symmetric Gaits 7 and 8B. The comprehensive investigations on other symmetric gaits are left to future work.

From previous studies, we know that Gait 7 moves the microswimmer along x_0 direction. We address the question how far Gait 7 can move the microswimmer toward the negative x_0-axis on the time interval $[0, T]$. Note that when we refer to Gait 7 in what follows, it does not have to be in the same square shape as in Fig. 7(a), but rather can

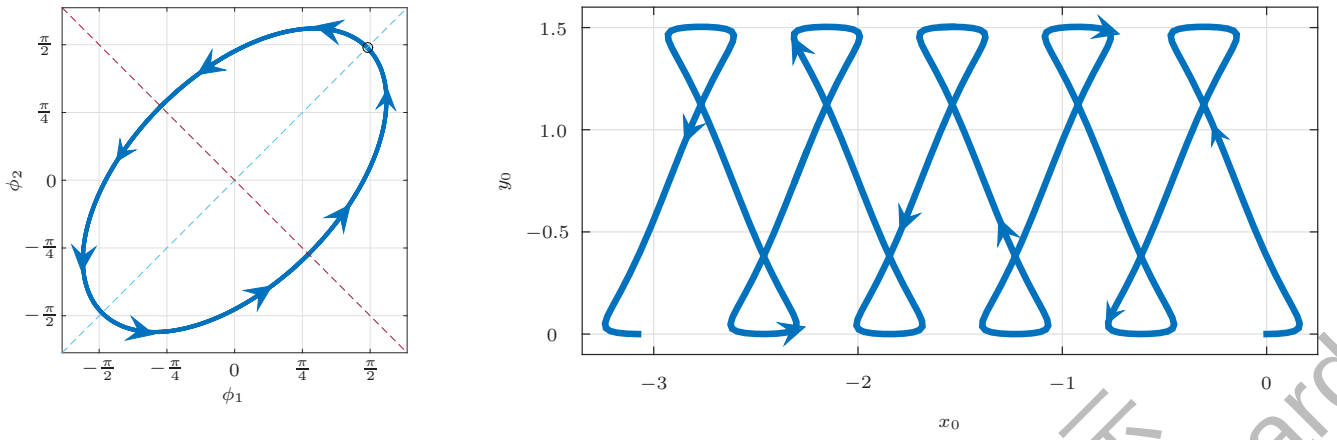
Gait	ϕ_1'	ϕ_2'	**Additional Conditions**
1	$\sum_{n=1,\text{odd}}^{N}(a_n\cos n\omega t + b_n \sin n\omega t)$	$\sum_{n=1,\text{odd}}^{N}(c_n\cos n\omega t + d_n \sin n\omega t)$	
4A	ditto	ditto	$\sum_{n=1,\text{odd}}^{N} a_n = 0$ $\sum_{n=1,\text{odd}}^{N} c_n = 0$
4B	$\sum_{n=1}^{N} b_n \sin n\omega t$	$\sum_{n=1}^{N} d_n \sin n\omega t$	
2	$a_0 + \sum_{n=1}^{N} a_n \cos n\omega t$	$\sum_{n=1}^{N} d_n \sin n\omega t$	
5A	ditto	ditto	$a_0 + \sum_{n=1}^{N} a_n = 0$
5B	$a_0 + \sum_{n=2,\text{even}}^{N}(a_n\cos n\omega t + b_n \sin n\omega t)$	$\sum_{n=1,\text{odd}}^{N}(c_n\cos n\omega t + d_n \sin n\omega t)$	$a_0 + \sum_{n=2,\text{even}}^{N} a_n = 0$ $\sum_{n=1,\text{odd}}^{N} c_n = 0$
3	$\sum_{n=1}^{N} b_n \sin n\omega t$	$c_0 + \sum_{n=1}^{N} c_n \cos n\omega t$	
6A	ditto	ditto	$c_0 + \sum_{n=1}^{N} c_n = 0$
6B	$\sum_{n=1,\text{odd}}^{N}(a_n\cos n\omega t + b_n \sin n\omega t)$	$c_0 + \sum_{n=2,\text{even}}^{N}(c_n\cos n\omega t + d_n \sin n\omega t)$	$\sum_{n=1,\text{odd}}^{N} a_n = 0$ $c_0 + \sum_{n=2,\text{even}}^{N} c_n = 0$
7	$\sum_{n=1,\text{odd}}^{N} a_n \cos n\omega t$	$\sum_{n=1,\text{odd}}^{N} d_n \sin n\omega t$	
8A	ditto	ditto	$\sum_{n=1,\text{odd}}^{N} a_n = 0$
8B	$\sum_{n=2,\text{even}}^{N} b_n \sin n\omega t$	$\sum_{n=1,\text{odd}}^{N} d_n \sin n\omega t$	
9A	$\sum_{n=1,\text{odd}}^{N} b_n \sin n\omega t$	$\sum_{n=1,\text{odd}}^{N} c_n \cos n\omega t$	$\sum_{n=1,\text{odd}}^{N} a_n = 0$
9B	$\sum_{n=1,\text{odd}}^{N} b_n \sin n\omega t$	$\sum_{n=2,\text{even}}^{N} d_n \sin n\omega t$	

TABLE III: Fourier Expansions of ϕ_1' and ϕ_2' for Gaits 1-9B.

be any gait that is symmetric with respect to both ϕ_1' and ϕ_2' axes, starts from ϕ_2' axis, and does not cross itself. Moreover, from the Fourier expansion listed in Table III, ϕ_1' of such a gait contains only the odd cosine terms and ϕ_2' only the odd sine terms.

This can be formulated as the following optimization problem

$$\begin{aligned} &\min_{\mu} x_0(T), \\ \text{subject to} \quad &\dot{p}_0 = \frac{1}{d_1} R(\theta_0) G_v \dot{\phi}, \\ &\dot{\theta}_0 = \frac{1}{d_1} G_\omega \dot{\phi}, \end{aligned} \tag{76}$$



(a) The gait looping around the ellipsoidal path for five times.

(b) Swimming trajectory moving towards negative x_0 direction.

FIG. 13: Gait to achieve a smallest possible $x_0(T)$ when $N = 5$.

where μ is the Fourier coefficient vector defined in Eq. (66). Set $N = 5$ and the initial guess as the unit circle:

$$\phi_1'(t) = \cos\omega t, \quad \phi_2'(t) = \sin\omega t.$$

The gradient descent algorithm generates the following result:

$$\begin{aligned} \phi_1'(t) &= -0.0003\cos\omega t + 0.0010\cos 3\omega t + 2.1765\cos 5\omega t, \\ \phi_2'(t) &= 0.0006\sin\omega t - 0.0008\sin 3\omega t + 1.2124\sin 5\omega t. \end{aligned} \tag{77}$$

Fig. 13(a) plots this gait, and (b) the corresponding swimming trajectory. It can be seen from Eq. (77) that the coefficients of the 1st and 3rd harmonics are much smaller than those of the 5th and can be effectively neglected. The optimal gait indeed travels along an ellipsoidal path for five times. This is because we have specified only the terminal time T and truncation order N in the formulation, and the swimming trajectory depends only on the geometric shape of the gait. Then within a given time interval, the optimal gait will be repeating a closed path for N times so as to move farther left.

To find a steering gait in one cycle, we add some additional penalty function, *e.g.*, the path length traversed by the center link:

$$J_L = \int_0^T \sqrt{\dot{x}_0^2 + \dot{y}_0^2}\, dt. \tag{78}$$

This is to say, among all the gaits that steer the microswimmer toward farther left, we want to find the one with short path length. This way the optimization algorithm will stay away from the gait that loops around a closed path for multiple times. Note that J_L is also independent of the time parameterization. To minimize $x_0(T)$ and J_L at the same time, one simple method is to combine them into a weighted sum:

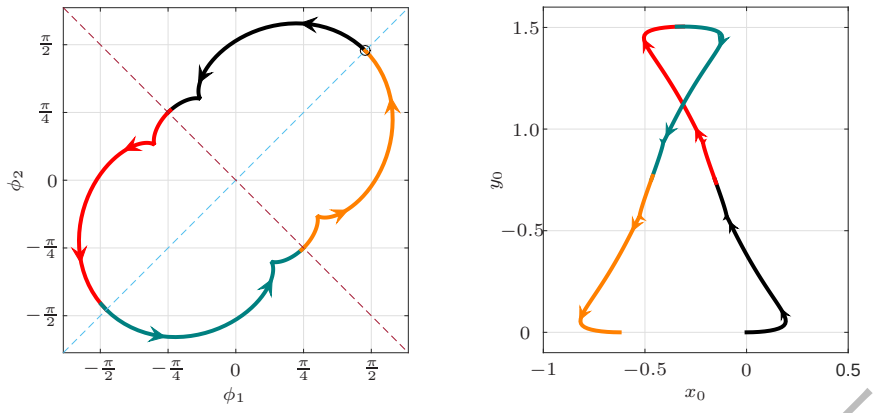
$$J = x_0(T) + \lambda J_L. \tag{79}$$

Here λ is a weighting factor that is chosen as, say, 0.01, to compromise between two penalty functions. The optimization problem can be written as

$$\begin{aligned} &\min_{\mu} \; x_0(T) + \lambda J_L, \\ \text{subject to} \quad &\dot{p}_0 = \frac{1}{d_1} R(\theta_0) G_v \dot{\phi}, \\ &\dot{\theta}_0 = \frac{1}{d_1} G_\omega \dot{\phi}. \end{aligned} \tag{80}$$

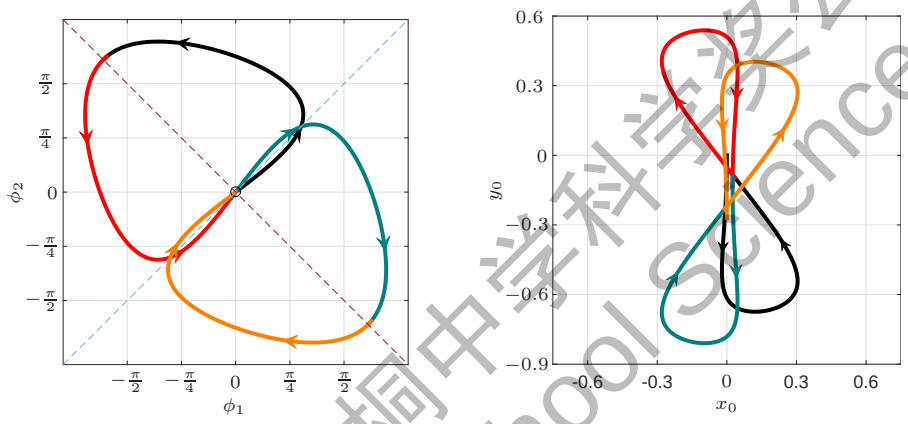
Set $N = 5$ and initial guess as the unit circle. The gradient algorithm generates the following gait:

$$\begin{aligned} \phi_1'(t) &= 1.3883\cos\omega t + 0.4562\cos 3\omega t + 0.2789\cos 5\omega t, \\ \phi_2'(t) &= 1.3108\sin\omega t + 0.4110\sin 3\omega t + 0.2108\sin 5\omega t. \end{aligned}$$



(a) Gait with path length 3.9944. (b) Swimming trajectory.

FIG. 14: (Color online) Optimized Gait 7 to achieve a small $x_0(T)$ with short path length.



(a) Gait when $N = 4$. (b) Swimming trajectory.

FIG. 15: (Color online) Optimized Gait 8B to achieve a small $y_0(T)$ with short path length.

Fig. 14(a) plots the obtained gait, and (b) the corresponding swimming trajectory. The minimized value of $x_0(T)$ is -0.6245, which is smaller than the lower bound of the range attained by the original Gait 7 in Fig. 9(b).

Next we study how far Gait 8B can steer the microswimmer toward the negative y_0-axis in one cycle. This can be formulated into the following optimization problem

$$\begin{aligned} &\min_{\mu} \ y_0(T) + \lambda J_L, \\ \text{subject to} \quad &\dot{p}_0 = \frac{1}{d_1} R(\theta_0) G_v \dot{\phi}, \\ &\dot{\theta}_0 = \frac{1}{d_1} G_\omega \dot{\phi}. \end{aligned} \tag{81}$$

From Table III, we know that ϕ_1' for Gait 8B contains only the even sine terms, and ϕ_2' only the odd sine terms. Set the initial guess as

$$\phi_1'(t) = \sin 2\omega t, \quad \phi_2'(t) = \sin \omega t.$$

Different truncation order N generates the optimization results as follows.

- $N = 4$:

$$\begin{aligned} \phi_1'(t) &= 1.6055 \sin 2\omega t + 0.1546 \sin 4\omega t, \\ \phi_2'(t) &= 1.9037 \sin \omega t - 0.8099 \sin 3\omega t, \\ y_0(T) &= -0.2710, \\ J_L &= 6.3072; \end{aligned}$$

- $N = 6$:

$$\begin{aligned} \phi_1'(t) &= 1.3714\sin 2\omega t - 0.4564\sin 4\omega t - 0.1537\sin 6\omega t \\ \phi_2'(t) &= 1.3996\sin \omega t - 0.9472\sin 3\omega t + 0.3811\sin 5\omega t \\ y_0(T) &= -0.2712, \\ J_L &= 6.3420; \end{aligned}$$

- $N = 8$:

$$\begin{aligned} \phi_1'(t) &= 1.0600\sin 2\omega t - 0.6726\sin 4\omega t + 0.0978\sin 6\omega t + 0.1025\sin 8\omega t \\ \phi_2'(t) &= 1.1243\sin \omega t - 0.8930\sin 3\omega t + 0.4860\sin 5\omega t - 0.2108\sin 7\omega t \\ y_0(T) &= -0.2703, \\ J_L &= 6.1425. \end{aligned}$$

These results show that different N generates similar values of $y_0(T)$, which are all considerably smaller than the lower bound of the range achieved by the original Gait 8B in Fig. 9(f). Fig. 15 shows the optimal gait and the corresponding swimming trajectory when $N = 4$. The gaits from $N = 6$ and $N = 8$ are also in similar shapes to Fig. 15(a), except that they are more curved because of the presence of high frequency components.

VI. CONCLUSION AND FUTURE WORK

In this work we studied dynamical behaviors and steering of a three-link microswimmer in a viscous fluid by analyzing its dynamics and symmetries and by applying optimization techniques.

We derived the analytic expressions of dynamics and discussed basic properties of its time evolutions. We carefully examined the symmetries in dynamics as well as their ramifications on the swimming trajectory and net rotation. Moreover, comprehensive investigations on general symmetric gaits were conducted. Based on these results, we studied the steering problem of microswimmer, which was formulated as an optimization problem and solved by gradient descent algorithm. Various steering results from different symmetries, initial conditions, truncation orders, and two-fold objectives were discussed.

In the future, we plan to explore many further interesting problems. One is to study the steering features of symmetric gaits other than 7 and 8B. Another is to combine other optimal functions such as power, energy, and efficiency. The ideal goal, of course, will be to build real multi-link microswimmers that realize the various engineering applications.

APPENDIX A: MATLAB CODE TO COMPUTE G_v AND G_ω

```
clear all;
syms l0 theta0 phi1 phi2 real;
a0=theta0; a1=theta0-phi1; a2=theta0+phi2;

D1=[1 0  l0*(sin(a0)+sin(a1));
   0 1 -l0*(cos(a0)+cos(a1));
   0 0  1];
D2=[1 0 -l0*(sin(a0)+sin(a2));
    0 1  l0*(cos(a0)+cos(a2));
    0 0  1];
E1=[-l0*sin(a1) 0;
     l0*cos(a1) 0;
     -1 0];
E2=[0 -l0*sin(a2);
    0  l0*cos(a2);
    0  1];

P0=[1+(sin(a0))^2    -sin(a0)*cos(a0) 0;
    -sin(a0)*cos(a0)  1+(cos(a0))^2   0;
    0 0 l0^2/6];
P1=[1+(sin(a1))^2    -sin(a1)*cos(a1) 0;
    -sin(a1)*cos(a1) 1+(cos(a1))^2    0;
```

```
    0 0 l0^2/6];
P2=[1+(sin(a2))^2   -sin(a2)*cos(a2) 0;
    -sin(a2)*cos(a2) 1+(cos(a2))^2   0;
    0 0 l0^2/6];

Z1=simplify( P0+ D1'*P1*D1 + D2'*P2*D2);
S2=simplify( D1'*P1*E1+D2'*P2*E2 );
R=[cos(theta0) -sin(theta0);
   sin(theta0)  cos(theta0)];

d1=det(Z1);
S1=simplify( -inv(Z1)*d1 );
g0=S1*S2;
G1=simplify([R'*g0(1:2,:); g0(3,:)]);
```

APPENDIX B: MATLAB CODE TO COMPUTE THE GRADIENT

```
function [xT, dxTda] = x0T_grad(a)

x3=dlode45(@dx0dt_dlarray, [0 1], dlarray([0;0;0;0], 'SS'), a);
xT=x3(1,1,end);
dxTda = dlgradient(xT,a);

end


function dx0dt = dx0dt_dlarray(t, x, a)

l0=1; w=2*pi; a0=a; rotangle=0; theta0=x(3);

M=length(a0)/2; phi1=a0(1); phi2=a0(M+1); phi1dot=0; phi2dot=0;
for k=1:(M-1)/2
    phi1=phi1+a0(2*k)*cos(k*w*t)+a0(2*k+1)*sin(k*w*t);
    phi2=phi2+a0(2*k+M)*cos(k*w*t) +a0(2*k+1+M)*sin(k*w*t);
    phi1dot=phi1dot-a0(2*k)*k*w*sin(k*w*t)+a0(2*k+1)*k*w*cos(k*w*t);
    phi2dot=phi2dot-a0(2*k+M)*k*w*sin(k*w*t)+a0(2*k+1+M)*k*w*cos(k*w*t);
end

z=[cos(rotangle) -sin(rotangle); sin(rotangle) cos(rotangle)]*[phi1 phi1dot; phi2 phi2dot];
phi1=z(1,1); phi2=z(2,1); phi1dot=z(1,2); phi2dot=z(2,2);

d1=l0^2*(64*cos(phi1)+64*cos(phi2)+24*cos(phi1)*cos(phi2)-32*sin(phi1)*sin(phi2)+12*cos(phi1)^2 ...
   +12*cos(phi2)^2+8*cos(phi1)*cos(phi2)^2+8*cos(phi1)^2*cos(phi2)+7*cos(phi1)^2*cos(phi2)^2 ...
    -16*cos(phi1)*sin(phi1)*sin(phi2)-16*cos(phi2)*sin(phi1)*sin(phi2) ...
    -11*cos(phi1)*cos(phi2)*sin(phi1)*sin(phi2)+98);

G1=dlarray(zeros(3,2));
G1(1,1)=-(l0^3*(288*sin(phi1-phi2)+48*sin(phi1-2*phi2)+4*sin(phi1+2*phi2)+22*sin(2*phi1+phi2) ...
        +55*sin(2*phi1)-41*sin(2*phi2)+11*sin(2*phi1+2*phi2)+96*sin(phi1+phi2)+468*sin(phi1)-186*sin(phi2)))/12;
G1(1,2)=(l0^3*(22*sin(phi1+2*phi2)-288*sin(phi1-phi2)+4*sin(2*phi1+phi2)-41*sin(2*phi1)+55*sin(2*phi2) ...
        -48*sin(2*phi1-phi2)+11*sin(2*phi1+2*phi2)+96*sin(phi1+phi2)-186*sin(phi1)+468*sin(phi2)))/12;
G1(2,1)=-(l0^3*(80*cos(phi1)+28*cos(phi2)+108*cos(phi1)*cos(phi2)+22*cos(phi1)^2+14*cos(phi2)^2 ...
        +34*cos(phi1)*cos(phi2)^2+11*cos(phi1)^2*cos(phi2)-11*cos(phi1)*sin(phi1)*sin(phi2) ...
        +2*cos(phi2)*sin(phi1)*sin(phi2)))/3;
G1(2,2)=-(l0^3*(28*cos(phi1)+80*cos(phi2)+108*cos(phi1)*cos(phi2)+14*cos(phi1)^2+22*cos(phi2)^2 ...
        +11*cos(phi1)*cos(phi2)^2+34*cos(phi1)^2*cos(phi2)+2*cos(phi1)*sin(phi1)*sin(phi2) ...
        -11*cos(phi2)*sin(phi1)*sin(phi2)))/3;
G1(3,1)=(l0^2*(96*cos(phi1)+36*cos(phi1)*cos(phi2)-48*sin(phi1)*sin(phi2)-12*cos(phi2)^2 ...
        +12*cos(phi1)*cos(phi2)^2+11*cos(phi1)^2*cos(phi2)^2-24*cos(phi2)*sin(phi1)*sin(phi2) ...
        -11*cos(phi1)*cos(phi2)*sin(phi1)*sin(phi2)+82))/3;
G1(3,2)=-(l0^2*(96*cos(phi2)+36*cos(phi1)*cos(phi2)-48*sin(phi1)*sin(phi2)-12*cos(phi1)^2 ...
        +12*cos(phi1)^2*cos(phi2)+11*cos(phi1)^2*cos(phi2)^2-24*cos(phi1)*sin(phi1)*sin(phi2) ...
        -11*cos(phi1)*cos(phi2)*sin(phi1)*sin(phi2)+82))/3;

z1=G1(1,1)*phi1dot+G1(1,2)*phi2dot;
z2=G1(2,1)*phi1dot+G1(2,2)*phi2dot;

dx0dt=dlarray(zeros(4,1), 'SS');
```

```
dx0dt(1)=cos(theta0)*z1-sin(theta0)*z2;
dx0dt(2)=sin(theta0)*z1+cos(theta0)*z2;
dx0dt(3)=G1(3,1)*phi1dot+G1(3,2)*phi2dot;

dx0dt(1:3)=dx0dt(1:3)/d1;

end
```

[1] J. E. Avron and O. Raz, *A geometric theory of swimming: Purcell's swimmer and its symmetrized cousin*, New Journal of Physics (2008).

[2] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, *Automatic differentiation in machine learning: a survey*, Journal of Machine Learning Research **18** (2018), no. 1, 1–43.

[3] Frank M Callier and Charles A Desoer, *Linear System Theory*, Springer New York, New York, NY, 1991.

[4] R. G. Cox, *The motion of long slender bodies in a viscous fluid part 1. general theory*, Journal of Fluid Mechanics **44** (1970), 791–810.

[5] Laetitia Giraldi, Pierre Martinon, and Marta Zoppello, *Optimal design of Purcell's three-link swimmer*, Physical Review E **91** (2015), 023012.

[6] Google TensorFlow, *Introduction to gradients and automatic differentiation*, (2022), https://www.tensorflow.org/guide/autodiff.

[7] Emiliya Gutman and Yizhar Or, *Symmetries and gaits for Purcell's three-link microswimmer model*, IEEE Transactions on Robotics **32** (2016), no. 1, 53–69.

[8] Ross L. Hatton and Howie Choset, *Connection vector fields and optimized coordinates for swimming systems at low and high reynolds numbers*, Proceedings of the ASME 2010 Dynamic Systems and Control Conference (Cambridge, Massachusetts, USA), September 2010.

[9] B. Jang, E. Gutman, N. Stucki, B. Seitz, P. Garcia-Wendel, T. Newton, J. Pokki, O. Ergeneman, S. Pané, Y. Or, and B. J. Nelson, *Undulatory locomotion of magnetic multi-link nanoswimmers*, Nano Letters **15** (2015), no. 7, 4829–4833.

[10] Eric Lauga and Thomas R Powers, *The hydrodynamics of swimming microorganisms*, Reports on Progress in Physics **72** (2009), no. 9, 096601.

[11] Cristina Nuevo-Gallardo, José Emilio Traver, Inés Tejado, and Blas M. Vinagre, *Purcell's three-link swimmer: Assessment of geometry and gaits for optimal displacement and efficiency*, Mathematics **2021** (2021), no. 9, 1088.

[12] E. M. Purcell, *Life at low reynolds number*, American Journal of Physics **45** (1977), no. 1, 3–11.

[13] Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, *Robot modeling and control*, second ed., John Wiley & Sons, 2020.

[14] Daniel Tam and A. E. Hosoi, *Optimal stroke patterns for Purcell's three-link swimmer*, Physical Review Letters **98** (2007), no. 6, 068105.

[15] A. Walther and A. Griewank, *Getting started with ADOL-C*, Combinatorial Scientific Computing (U. Naumann and O. Schenk, eds.), Chapman-Hall CRC Computational Science, 2012.

[16] F. W. Warner, *Foundations of differentiable manifolds and Lie groups*, Springer-Verlag, New York, 1983.

[17] Oren Wiezel and Yizhar Or, *Using optimal control to obtain maximum displacement gait for Purcell's three-link swimmer*, IEEE 55th Conference on Decision and Control (Las Vegas, USA), December 2016.

ACKNOWLEDGMENTS

This report summarizes a year-long project in which I learned to apply many useful mathematical techniques to study an interesting problem in fluid mechanics. Upon completion, I am deeply grateful to Professor Mian Li at Shanghai Jiao Tong University for his wonderful supervision and great patience. He guided me from the fundamentals of mathematical analyses, derivations, and optimizations to Matlab programming and technical writing. This project will not be possible without his guidance. I am also indebted to Professor ZiJie Qu at SJTU for introducing me basic properties of microswimmers in a low Reynolds number world. Moreover, I really appreciate various assistances from Mr. Hui Liao at my high school. Last but not the least, I want to thank my parents and older bother for their constant supports to overcome numerous obstacles throughout this project.

Brief biographical information

Aine Zhang is a G11 student at Shanghai High School International Division. She has kept keen interests in mathematics, physics, and linguistics. She has won awards in prestigious competitions including International Linguistics Olympiad China and Math League Contests. Ms. Zhang is a native speaker in both English and Chinese, and she reads Latin at intermediate level.

Dr. Mian Li is currently a tenured full professor and Assistant Dean of Global Institute of Future Technologies at Shanghai Jiao Tong University. He received the BS and MS degrees from Tsinghua University, and the PhD degree from the University of Maryland at College Park with the Best Dissertation Award, in 1999, 2001, and 2007, respectively. Dr. Li is a Fellow of the American Society of Mechanical Engineers (ASME) and serves on the editorial boards of several prestigious international journals. His current research interests include mathematical control and multidisciplinary optimization.