Physics of Fluids ARTICLE scitation.org/journal/phf

# Mechanical rotation at low Reynolds number via reinforcement learning

Cite as: Phys. Fluids **33**, 062007 (2021); doi: 10.1063/5.0053563
Submitted: 8 April 2021 · Accepted: 21 May 2021 ·
Published Online: 17 June 2021 · Corrected: 18 June 2021



Yuexin Liu,[1] Zonghao Zou,[2] Alan Cheng Hou Tsang,[3] On Shun Pak,[2] and Y.-N. Young[1,a)]

## AFFILIATIONS

[1]Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, New Jersey 07102, USA
[2]Department of Mechanical Engineering, Santa Clara University, Santa Clara, California 95053, USA
[3]Department of Mechanical Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China

[a)]Author to whom correspondence should be addressed: yyoung@njit.edu

## ABSTRACT

There is growing interest in the development of artificial microscopic machines that can perform complex maneuvers like swimming microorganisms for potential biomedical applications. At the microscopic scales, the dominance of viscous over inertial forces imposes stringent constraints on locomotion. In the absence of inertia, Purcell first proposed an elegant way to generate net translation using kinematically irreversible motions [E. M. Purcell, "Life at low Reynolds number," Am. J. Phys. **45**, 3–11 (1977)]. In addition to net translation, a more recent prototype known as Purcell's "rotator" has been proposed in Dreyfus *et al.* ["Purcell's "rotator": Mechanical rotation at low Reynolds number," Eur. Phys. J. B **47**, 161–164 (2005)] as a mechanical implementation of net rotation at low Reynolds numbers. These ingenious designs rely on knowledge of the surrounding environment and the physics of locomotion within the environment, which may be incomplete or unclear in more complex scenarios. More recently, reinforcement learning has been used as an alternative approach to enable a machine to learn effective locomotory gaits for net translation based on its interaction with the surroundings. In this work, we demonstrate the use of reinforcement learning to generate net mechanical rotation at low Reynolds numbers without requiring prior knowledge of locomotion. For a three-sphere configuration, the reinforcement learning recovers the strategy proposed by Dreyfus *et al.* As the number of spheres increases, multiple effective rotational strategies emerge from the learning process. However, given sufficiently long learning processes, all machines considered in this work converge to a single type of rotational policies that consist of traveling waves of actuation, suggesting its optimality of the strategy in generating net rotation at low Reynolds numbers.



## I. INTRODUCTION

Swimming microorganisms inhabit a world dominated by viscous force. The Reynolds number, $\mathrm{Re} = \rho U\ell/\mu$ (where $\ell$ and $U$ represent the characteristic length and speed of the swimmer and $\rho$ and $\mu$ are fluid density and dynamic viscosity, respectively), falls in the range of $10^{-4}$–$10^{-2}$ for swimming bacteria and spermatozoa.[1–3] The inertial force is therefore negligible compared with the viscous force. At such low Reynolds numbers, common swimming strategies based on inertia at the macroscopic scales become largely ineffective.[4,5] Microorganisms have evolved different strategies, including the use of flagellar rotary motors[6] or the action of molecular motors within flagella,[7] to swim effectively in their microscopic world. There is growing interest in developing artificial microscopic machines that can self-propel like their biological counterparts for potential biomedical and environmental applications.[8,9] However, without sophisticated biological molecular machines possessed by microorganisms, it remains a challenge to design micromachines for complex maneuvers in the viscously dominated flow limit.[10]

Purcell's work popularized the fundamental fluid dynamical aspects of swimming at low Reynolds numbers.[11] In particular, his scallop theorem rules out any reciprocal motion-sequence of motions with time-reversal symmetry (e.g., opening and closing the hinge of a single-hinged scallop) for self-propulsion without inertia. To escape from the constraints by the scallop theorem, Purcell designed a three-link swimmer that can perform kinematically irreversible cyclic motions for net translation.[11,12] Najafi and Golestanian[13] proposed another ingenious design consisting of three linked spheres, which can translate by modulating the distances between the spheres; the mechanism inspired a wide variety of variants.[14–22] In addition to net translation, the design of mechanisms that can produce net rotation at the microscale is important to the development of micromachines. To this end, Dreyfus *et al.* proposed a mechanism (also known as Purcell's rotator),[23] which

consists of three spheres linked like the spokes on a wheel (Fig. 1) as the rotational analog of Purcell's three-link swimmer for translation. The rotator performs a prescribed sequence of motions that exploit the hydrodynamic interaction between the spheres to produce net rotation. The mechanism of Purcell's rotator shares similarity with the conformational changes of some molecular motors undergoing ATP (adenosine triphosphate) or photochemically driven rotational movements.[23–25]

These ingenious designs rely on knowledge of the surrounding environment and the physics of locomotion within the environment, which may not be complete or clear in more complex scenarios. In particular, for biological applications, the properties of some highly complex, heterogeneous biological environments may not be known *a priori*, posing additional challenges on the design of effective self-propelled micromachines. Recent approaches have exploited the prowess of machine learning in the studies of different aspects of locomotion in fluids,[26,27] including individual and collective motions of fish[28–34] and birds,[35,36] as well as different navigation[37–45] and cloaking[46] problems of self-propelled objects. In particular, an alternative framework based on reinforcement learning has enabled a microswimmer to learn effective locomotory gaits based on its interactions with the surrounding low Reynolds number environment.[47] Without any prior knowledge of locomotion, such a "self-learning" microswimmer is able to acquire a previously known propulsion strategy by Najafi and Golestanian[13] for net translation and adapt its locomotory gaits in different media.

Similar in spirit, in this work, we employ a reinforcement learning approach to generate mechanical rotation at low Reynolds numbers. We adopt the mechanical configuration of Purcell's rotator shown in Fig. 1;[23] however, instead of prescribing the locomotory gaits of Purcell's rotator, we allow the machine to progressively learn how to exploit hydrodynamic interactions to produce net rotation via reinforcement learning on its own. We will examine the locomotion strategies acquired by the learning process and consider more complex scenarios when the number of spheres in the machine increases. The paper is organized as follows: in Sec. II, we present the geometric setup (Sec. II A) formulation of the hydrodynamic (Sec. II B) and the reinforcement learning (Sec. II C) problems used in this work. We discuss the results in Sec. III for a three-sphere rotator (Sec. III A) before extending the studies to configurations with a higher number of spheres (Sec. III B). We conclude the investigation with some remarks in Sec. IV.

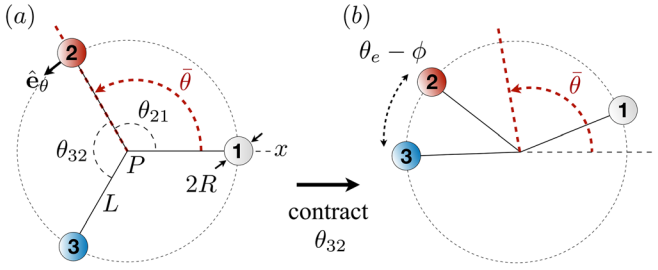


**FIG. 1.** Schematic diagram and notations of a mechanical setup based on Purcell's rotator by Dreyfus *et al.*[23] The machine consists of three spheres of radius $R$ connected to the center $P$ with connecting rods of length $L$. The spheres are connected to the center of the circle $P$ with connecting rods. (a) In its initial configuration, the three spheres have equal angular spacing, $\theta_e = 2\pi/3$. There exist active elements that can contract the angle $\theta_{21}$ or $\theta_{32}$ by an amount $\phi$ or expand by the same amount to return to the value $\theta_e$. In (b), we illustrate the configuration of the machine after it contracts the angle $\theta_{32}$, which results in an overall change of the angular centroid of the machine, $\bar{\theta}$ (indicated by the red dashed lines).

## II. FORMULATION

### A. Geometric setup

We first illustrate the geometric setup using a three-sphere configuration similar to Purcell's rotator [Fig. 1(a)] before considering systems with an increased number of spheres. We place three spheres of radius $R$ on an imaginary circle of radius $L$. The spheres are individually connected to the center of the circle $P$ with connecting rods. Figure 1 shows an initial configuration with equal angular spacing ($\theta_e = 2\pi/3$) between the spheres, where the angle between spheres 2 and 1 ($\theta_{21}$) and the angle between spheres 3 and 2 ($\theta_{32}$) attain their fully extended values ($\theta_{21} = \theta_{32} = \theta_e$). There exist two internal active elements that can contract $\theta_{21}$ or $\theta_{32}$ (referred to as active angles here) by an amount $\phi$ [Fig. 1(b)] or expand an angle back to its fully extended value $\theta_e$. The remaining angle between spheres 3 and 1 ($\theta_{13}$) only reacts passively to the contraction and expansion. To measure the net rotation of the machine, we define the angular centroid $\bar{\theta} = \sum_1^3 \theta_i/3$, which is the average of the angles of all spheres $\theta_i$ measured from the $x$-axis. The angular centroid of the initial configuration shown in Fig. 1(a) is given by $\bar{\theta} = 2\pi/3$, as indicated by the red dashed line. Actuating (contracting or expanding) any of the active angles will alter the angular centroid of the machine as illustrated in Fig. 1(b). The goal of the machine is to generate net rotation (i.e., a net increase in the angular centroid $\bar{\theta}$) in the anticlockwise direction by choosing different actions of the active elements. Without requiring prior knowledge of low Reynolds number locomotion, we will demonstrate a reinforcement learning approach in achieving this goal. We next present the formulation of the hydrodynamic problem in Sec. II B and its integration with a reinforcement learning algorithm in Sec. II C.

### B. Low Reynolds number hydrodynamics

We consider the hydrodynamics governed by the Stokes equation ($\nabla p = \mu\nabla^2\mathbf{u}$, $\nabla\cdot\mathbf{u} = 0$) in the low Reynolds number regime, where $p$ and $\mathbf{u}$ represent, respectively, the pressure and velocity fields. Here we neglect the hydrodynamic influence of the connecting rods and account for the leading-order hydrodynamic interaction between the spheres in the fluid via the Oseen tensor[13,23,48] in the limit $R/L \ll 1$. The forces $\mathbf{F}_i$ and velocities $\mathbf{V}_i$ of the spheres ($i = 1, 2, 3$) are related as

$$\mathbf{F}_i = \sum_{j=1}^{3} \mathbf{H}_{ij}\mathbf{V}_j, \tag{1}$$

where

$$\mathbf{H}_{ij} = \begin{cases} -6\pi\mu R\,\mathbf{I}, & \text{if } i = j, \\ 6\pi\mu R\dfrac{3R}{4R_{ij}}\left(\mathbf{I} + \hat{\mathbf{R}}_{ij}\hat{\mathbf{R}}_{ij}\right), & \text{if } i \neq j, \end{cases} \tag{2}$$

and $R_{ij} = ||\mathbf{r}_i - \mathbf{r}_j||$, $\mathbf{r}_i$ is the position of sphere $i$ from the center $P$, $\hat{\mathbf{R}}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)/R_{ij}$, and $\mathbf{I}$ is the identity matrix. The torque about the origin in the laboratory frame is given by $\mathbf{\Gamma}_i = \mathbf{D}_i \times \mathbf{F}_i = \mathbf{D}_i \times \sum_{j=1}^3 \mathbf{H}_{ij}\mathbf{V}_j$, where $\mathbf{D}_i$ is the position vector of each spheres in the laboratory frame. Here we focus on pure rotation of the machine and

thus fix its center $P$ to the origin in the laboratory frame. If the center is not kept fixed, the machine can undergo both translation and rotation.[23] The velocity of the spheres $\mathbf{V}_i = L\dot{\theta}_i\,\hat{\mathbf{e}}_\theta$ is therefore purely tangential to the imaginary circle, where $\hat{\mathbf{e}}_\theta$ is the unit vector tangent to the circle. In the absence of an external torque, the system is torque-free,

$$\sum_{i=1}^{3} \mathbf{\Gamma}_i = \mathbf{0}. \tag{3}$$

The machine is allowed to actuate any one of the active elements in each step to contract or expand the angle at a rate $\omega$. For instance, in Fig. 1 from (a) to (b), the machine contracts the angle $\theta_{32}$ by an amount $\phi$ (i.e., $\dot{\theta}_3 - \dot{\theta}_2 = -\omega$) while maintaining the angle $\theta_{21}$ fixed (i.e., $\dot{\theta}_2 = \dot{\theta}_1$). Such action results in an overall change of the angular centroid of the machine, $\bar{\theta}$ (indicated by the red dashed lines in Fig. 1). These kinematic constraints close the system of equations, which can be numerically solved to determine the rotational dynamics of the machine for each action taken. We remark that the linearity and time-independence of the Stokes equation lead to the property of rate independence:[5,11] any translational or rotational displacement of the machine resulting from its configurational changes (contraction/expansion of active angles) does not depend on the rate of configurational changes but only on the sequence of the changes. We, therefore, follow Dreyfus *et al.*[23] and assume a uniform rate of expansion and contraction $\omega$ in this work. We also consider small actuation angles $\phi$ in order for the far-field hydrodynamic description to be valid.

### C. Reinforcement learning

In this work, we define a stroke as an action between two configuration states, and a cycle as a sequence of strokes that starts and ends with the same configuration state. The goal of the machine is to generate net rotation by performing an effective sequence of strokes. Instead of prescribing the sequence of strokes in the conventional approach, here we use a simple reinforcement learning algorithm to enable the machine to acquire effective locomotion strategies by itself. Such an approach does not rely on prior knowledge of locomotion but allows the machine to learn and adapt its locomotion strategies based on its experience interacting with the surroundings. Here we implement the $Q$-learning algorithm for its simplicity and expressiveness compared with other reinforcement learning algorithms.[49]

In a given learning step in $Q$-learning (for example, the $n$th step in Fig. 2) the machine performs an action ($a_n$, contracting or expanding one of the active angles) taking the machine from the current configuration state ($s_n$) to the next state ($s_{n+1}$). The "success" of action $a_n$ is measured by reward $r_n$, which is defined as the resulting difference of the angular centroid (i.e., $r_n = \bar{\theta}_{n+1} - \bar{\theta}_n$). The expected long-term reward for taking the action $a_n$ given the state $s_n$ is quantified by the $Q$-matrix, $Q(s_n, a_n)$, which is an action-value function that encodes the adaptive decision-making intelligence of the machine. After each learning step, the $Q$-matrix evolves based on the experience gained by the machine,

$$Q(s_n, a_n) \leftarrow Q(s_n, a_n) + \alpha\left[r_n + \gamma \max_{a_{n+1}} Q(s_{n+1}, a_{n+1}) - Q(s_n, a_n)\right], \tag{4}$$

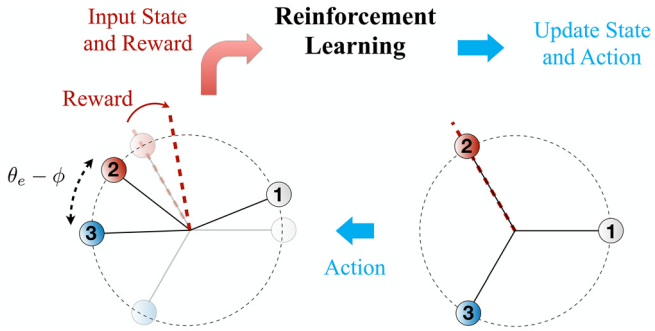


**FIG. 2.** Mechanical rotation at low Reynolds numbers via reinforcement learning. The goal of the machine is to generate net rotation by performing different configurational changes. Instead of designing a sequence of locomotory gaits in advance, here we leverage a simple reinforcement learning algorithm ($Q$-learning) to enable the machine to acquire an effective locomotion strategy based on its interaction with the surroundings. In each learning step, the machine performs an action $a_n$ (contracting or expanding on the active angles) to transform from one configuration state $s_n$ to the next $s_{n+1}$. The reward $r_n$, defined as the resulting difference of the angular centroid ($\bar{\theta}_{n+1} - \bar{\theta}_n$) measures the success of each action. The reinforcement learning process progressively updates the $Q$-matrix, which encodes the adaptive decision-making intelligence of the machine.[49]

where $\alpha$ is the learning rate ($0 \le \alpha \le 1$) that determines to what extent new information overrides old information and therefore controls the learning speed of the machine. Here we fixed $\alpha = 1$ to maximize the learning speed. The discount factor $\gamma$ ($0 < \gamma < 1$) determines the trade-off between immediate reward $r_n$ and maximum future reward at the next state $\max_{a_{n+1}} Q(s_{n+1}, a_{n+1})$. When $\gamma$ is small, the machine is shortsighted and tends to maximize the immediate reward; when $\gamma$ is large, the swimmer is farsighted and takes actions that maximize the long-term reward. In order to avoid the machine from being trapped in locally optimal policies, we implemented an $\varepsilon$-greedy selection scheme: In each learning step, the machine chooses the best action recommended by the $Q$-matrix with a probability $1 - \varepsilon$ or takes a random action with a small probability $\varepsilon$, which allows the machine to explore new solutions.

As a remark, the configuration states considered here correspond to the shape space in the literature, which contains all possible shapes of the machine without considering the positions and orientations of the rotator.

## III. RESULTS AND DISCUSSION

### A. Three-sphere rotator

We first consider a three-sphere configuration in this section. Instead of prescribing any sequence of strokes, we allow the rotator to take an action based on the $Q$-matrix (Sec. II C) and use the resulting reward to update the $Q$-matrix, informing the next action. We measure the net rotation of the machine $\Delta\bar{\theta} = \bar{\theta}_n - \bar{\theta}_0$ by comparing the angular centroid at the $n$-th learning step ($\bar{\theta}_n$) with the initial angular centroid ($\bar{\theta}_0$). Figure 3(a) shows a typical learning process of a three-sphere rotator: the rotator takes the initial steps to explore the viscous environment [Fig. 3(b)] without forming an effective rotational strategy yet. As the machine learns from its interaction with the environment progressively, it eventually repeats the same sequence of cyclic motions that produce net rotation in the anticlockwise direction

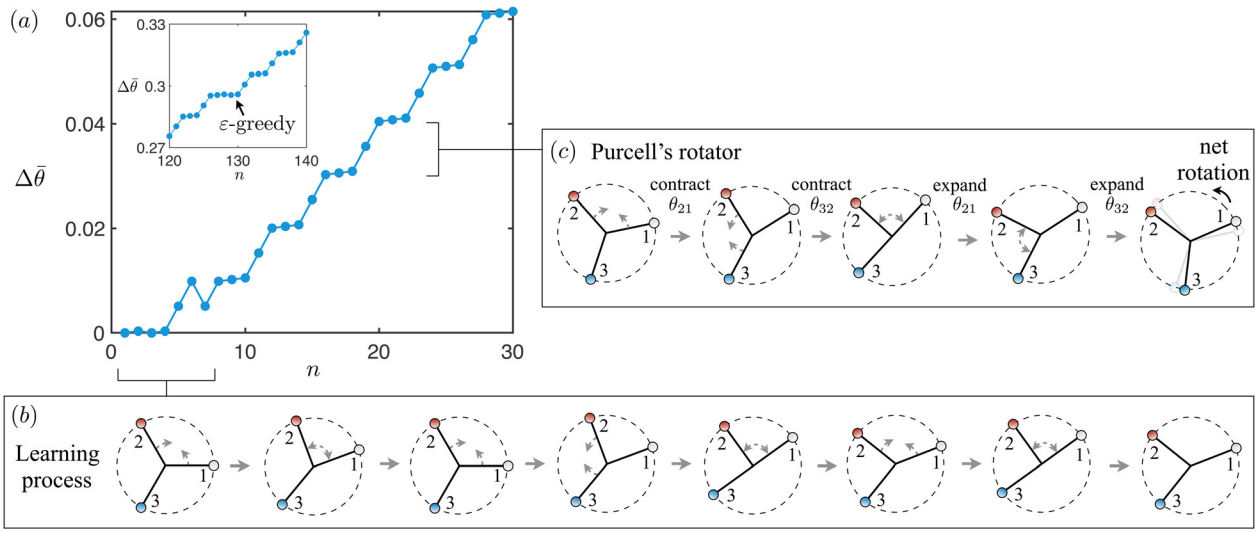


**FIG. 3.** Reinforcement learning of a three-sphere ($N = 3$) rotator. (a) The net rotation of the machine, measured by the change of angular centroid $\Delta\bar{\theta}$, generated by a series of actions at different learning steps $n$. (b) The rotator undergoes an initial learning stage by performing different actions to interact with the surrounding environment and learn from the resulting rewards. (c) Via reinforcement learning, the machine eventually repeats a sequence of cyclic motions that produce net rotation in the anticlockwise direction. The strategy acquired through reinforcement learning here coincides with that used for Purcell's rotator by Dreyfus *et al.*[23] Inset in (a): the $\varepsilon$-greedy scheme allows a small probability $\varepsilon$ for the machine to act against the $Q$-matrix and perform a random action for exploration. Here we set $\phi = \pi/6$, $\gamma = 0.9$, $\varepsilon = 0.05$, and $R/L = 0.1$. The rigid body rotation illustrated in panels (b) and (c) is magnified by twenty times for better visualization of the rotational motion.

[Fig. 3(c)]. We note that the policy harvested by reinforcement learning here coincides with the mechanism proposed by Dreyfus *et al.* for Purcell's rotator.[23] As the analog of the self-learning swimmer that produces net translation,[47] our example here demonstrates the first use of reinforcement learning to generate net mechanical rotation in a low Reynolds number environment, without requiring prior knowledge of locomotion.

As a remark, even when the machine is informed by the $Q$-matrix to repeat the same sequence of strokes after sufficient learning steps [Fig. 3(a), inset], the use of the $\varepsilon$-greedy selection scheme allows a small but nonzero probability $\varepsilon$ for the machine to act against the $Q$-matrix and perform a random action for exploration. The sequence of strokes is therefore sometimes interrupted with random actions as shown in the inset. Such a mechanism avoids being trapped around only locally optimal policies. For the three-sphere configuration, the machine eventually returns to Purcell's rotator sequence after the random actions. We will examine the effect of the magnitude of $\varepsilon$ with more complex examples in Sec. III B.

## B. *N*-sphere rotator

We next extend the analysis beyond the three-sphere configuration. For a configuration with $N$ spheres, the description of the hydrodynamic force and velocity via the Oseen tensor on sphere $i$ can be readily extended from Eq. (1) as $\mathbf{F}_i = \sum_{j=1}^{N} \mathbf{H}_{ij}\mathbf{V}_j$. Similarly, the torque-free condition now reads $\sum_{i=1}^{N} \mathbf{\Gamma}_i = \mathbf{0}$, where $\mathbf{\Gamma}_i = \mathbf{D}_i \times \sum_{j=1}^{N} \mathbf{H}_{ij}\mathbf{V}_j$. Similar to the case of three spheres, there are $N - 1$ active elements that can contract or expand any one of the angles between two neighboring spheres by an amount $\phi$, except for the angle $\theta_{1N}$, which only reacts passively to the contraction and expansion of other angles. At each step, the $Q$-learning algorithm informs one pair of neighboring spheres (e.g., the $i$ and $i+1$ spheres) to extend or contract their angle at a uniform rate $\omega$: $\dot{\theta}_{i+1} - \dot{\theta}_i = \pm\omega$, while keeping other angles fixed (i.e., $\dot{\theta}_j = \dot{\theta}_i$ for $j = 1, 2, \ldots, i-1$ and $\dot{\theta}_j = \dot{\theta}_{i+1}$ for $j = i+2, i+3, \ldots, N$). The goal is to learn effective strategies to generate net rotation based on the machine's interaction with the viscous environment.

We remark that as the number of sphere $N$ in the machine increases, the angle between the spheres in its initial (equally spaced) configuration reduces accordingly as $\theta_e = 2\pi/N$. This also limits the angle of contraction ($\phi$) allowed as the number of spheres increases in the machine. In order for $\phi$ to not exceed the maximum angle between the spheres ($\theta$), we set $\phi = \theta_e/4 = \pi/(2N)$ in our simulations for a $N$-sphere system. In other words, the machine uses a fixed portion of $\theta_e$ for contraction. The machine, hence, has a smaller angle of contraction as the number of sphere increases. We note that only a small portion (1/4) of $\theta_e$ is used for contraction here to ensure that the spheres are sufficiently far apart for the hydrodynamic description via the Oseen tensor to be valid (see Sec. II B).

When we have a larger number of spheres $N$ in the machine, the increased degree of freedom allows multiple effective strategies to emerge. The policy identified by reinforcement learning largely depends on different learning parameters, including the discount factor, the number of learning steps, and the value of $\varepsilon$ in the $\varepsilon$-greedy scheme. We illustrate some general characteristics using a four-sphere ($N = 4$) configuration. Figure 4(a) shows that, for a fixed number of learning steps, a four-sphere machine evolves different rotational policies depending on the value of $\varepsilon$ in the $\varepsilon$-greedy scheme. We can measure the performance of different policies by the angular displacement per cycle ($\Delta\bar{\theta}_C$) or the displacement per cycle per stroke ($\Delta\bar{\theta}_S = \Delta\bar{\theta}_C/N_S$); the latter measure divides the angular displacement per cycle by the number of strokes involved in the cycle, $N_S$, to account for the difference in the number of strokes in individual policies.

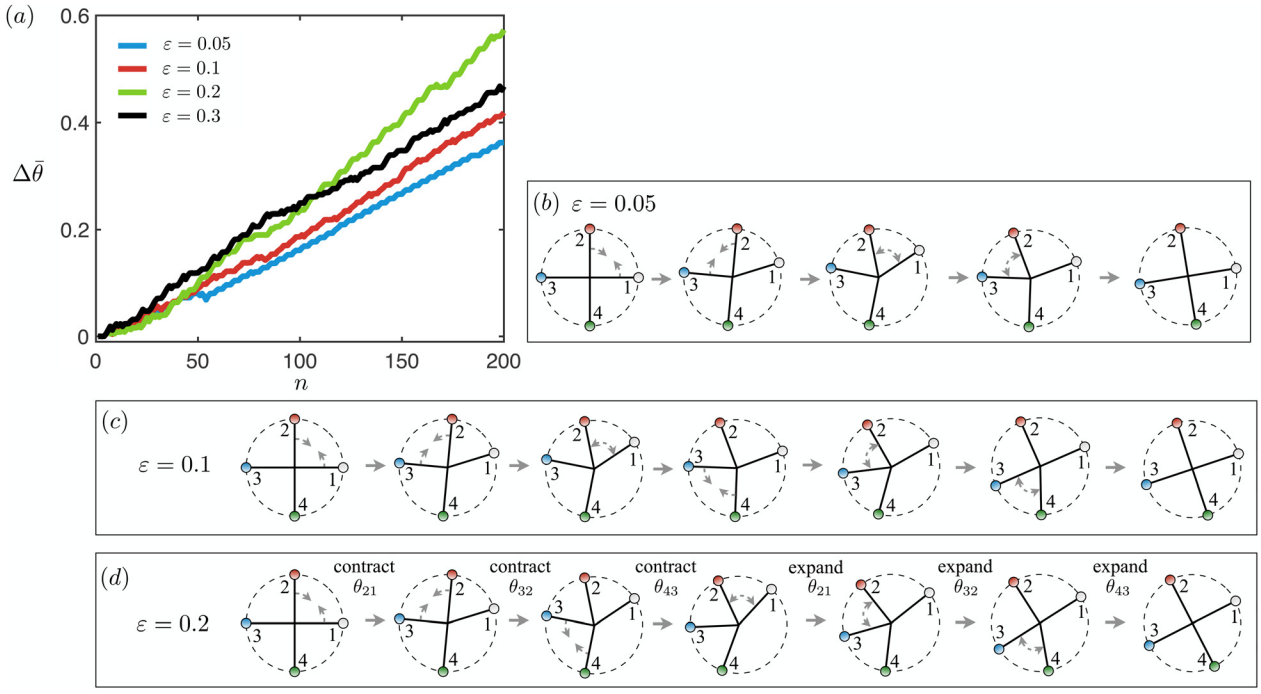


**FIG. 4.** Reinforcement learning of a four-sphere ($N=4$) rotator. (a) The net rotation of the machine, measured by the change of angular centroid $\Delta\bar{\theta}$, generated by a series of actions at different learning steps $n$. The value of $\varepsilon$ in the $\varepsilon$-greedy scheme affects the policies acquired by the machine at the end of the learning process. (b) With $\varepsilon = 0.05$, the machine has learned four-stroke cyclic motion same as that in the three-sphere rotator [Fig. 3(b)] without utilizing the active angle $\theta_{43}$. The angular displacement per cycle $\Delta\bar{\theta}_C = 0.008$; the angular displacement per cycle per stroke $\Delta\bar{\theta}_S = 0.002$. (c) With $\varepsilon = 0.1$, the machine has learned an improved but suboptimal six-stroke cyclic motion with $\Delta\bar{\theta}_C = 0.0161$ and $\Delta\bar{\theta}_S = 0.0027$. (d) With $\varepsilon = 0.2$, the machine further improves the performance with another six-stroke cyclic motion with $\Delta\bar{\theta}_C = 0.0238$ and $\Delta\bar{\theta}_S = 0.004$. The motion involves a sequential contraction of all active angles $\theta_{i+1,i}$ from $i=1$ to $i=3$, followed by a sequential expansion of all active angles $\theta_{i+1,i}$ from $i=1$ to $i=3$. This policy, which consists of traveling waves of actuation propagating in the anticlockwise direction, represents an extension of the strategy in Purcell's rotator to the case four spheres with all active angles utilized in the sequence. As a remark, the policy obtained with $\varepsilon = 0.3$ is the same as that with $\varepsilon = 0.2$; yet the more frequent interruptions by the random actions with $\varepsilon = 0.3$ leads to a smaller net rotation overall compared with the case with $\varepsilon = 0.2$ as shown in (a). In these simulations, $\phi = \pi/8$, $\gamma = 0.9$, and $R/L = 0.1$. The rigid body rotation illustrated in panels (b)–(d) is magnified by twenty times for better visualization of the rotational motion.

Similar to the case for translation,[47] the value of $\varepsilon$ in the $\varepsilon$-greedy scheme plays an important role in the learning process. When there is not any exploration scheme ($\varepsilon = 0$) the machine frequently gets trapped going back and forth between two states, resulting in reciprocal motion that does not yield net rotation.[47] With a small $\varepsilon = 0.05$ [blue line in Fig. 4(a)], the machine is able to identify an effective but suboptimal policy for net rotation [Fig. 4(b)]; indeed the four-stroke policy follows the same sequence of strokes as a three-sphere Purcell's rotator in Fig. 3(c), with the angle $\theta_{43}$ not participating in the gait at all (sphere 4 thus acts essentially like a passive cargo). The angular displacement per cycle for this policy is given by $\Delta\bar{\theta}_C = 0.008$ and $\Delta\bar{\theta}_S = \Delta\bar{\theta}_C/4 = 0.002$ on a per stroke basis. As we increase the exploration rate [$\varepsilon = 0.1$, red line in Fig. 4(a)], the machine learns an improved six-stroke policy [Fig. 4(c)] with larger $\Delta\bar{\theta}_C = 0.0161$ and $\Delta\bar{\theta}_S = \Delta\bar{\theta}_C/6 = 0.0027$. For $\varepsilon = 0.2$ [green line in Fig. 4(a)], the machine acquires another six-stroke policy as shown in Fig. 4(d) with further improved $\Delta\bar{\theta}_C = 0.0238$ and $\Delta\bar{\theta}_S = \Delta\bar{\theta}_C/6 = 0.004$. This policy here consists of contraction of all active angles in a sequential manner starting from $\theta_{21}$ in the anticlockwise direction, followed by expansion of all active angles again in a sequential manner starting from $\theta_{21}$. More generally, we define such type of policies as traveling wave policies, which consist of a sequential contraction of angles $\theta_{i+1,i}$ from $i=1$ to $i = N-1$ followed by a sequential expansion of angles $\theta_{i+1,i}$ from $i=1$ to $i = N-1$, because the sequence of action corresponds to propagation of traveling wave of actuation in the anticlockwise direction. The sequential actuation of a $N$-sphere system with a traveling wave policy is illustrated below,

$$\theta_{21} \rightarrow \theta_{32} \rightarrow \cdots \rightarrow \theta_{i+1,i} \rightarrow \cdots \rightarrow \theta_{N,N-1} \tag{5}$$

These traveling wave policies, therefore, consist of $2(N-1)$ strokes; indeed, the sequence of strokes in Purcell's rotator ($N=3$) in Fig. 3(c) and the $N=4$ policy in Fig. 4(d) are both traveling wave policies. As a remark, with an even higher exploration rate ($\varepsilon = 0.3$), the machine also learns the traveling wave policy [black line, Fig. 4(a)]; yet, the overall displacement of the angular centroid is less compared with the case with $\varepsilon = 0.2$ (green line) due to frequent interruptions by the random actions at the higher value of $\varepsilon$.

Next, we further increase the number of spheres in the system up to $N=9$. Similar to the case of translation,[47] the learning parameters $\alpha$, $\gamma$, and the number of learning steps affect the policy eventually adopted by the machine. For a system with a large number of spheres (e.g., $N=9$), a sufficiently large discount factor $\gamma$ and maximized

learning rate $\alpha$ is generally required for effective performance (see the supplementary material). Unless otherwise stated, we set $\alpha = 1$ and $\gamma = 0.9$ in the simulations. We examine the number of different policies obtained by reinforcement learning for systems with different numbers of spheres, $N$. The policy eventually adopted by the machine largely depends on the number of learning steps allowed. In Fig. 5(a), we show the number of different policies adopted by the machine when its rotation has reached a certain target angular displacement, $\Delta\bar{\theta}_T$, in different sample runs. For instance, when the machine is allowed to learn up to a target angular displacement of $\Delta\bar{\theta}_T = 2\pi$ [top panel, Fig. 4(a)], all trials for $N = 3$ and $N = 4$ machines converge to a single policy—the traveling wave policy. However, increasingly more policies emerge in the trials for machines with a larger number of spheres. When more learning is allowed by increasing the target angular displacement to $\Delta\bar{\theta}_T = 50\pi$ [middle panel in Fig. 5(a)] more configurations converge to the traveling wave policies ($N = 3$ to $N = 6$) with lower number of policies appearing in the trials for $N > 7$. Finally, when sufficient amount of learning is allowed [e.g., $\Delta\bar{\theta}_T = 350\pi$, bottom panel in Fig. 5(a)], all configurations considered converge to a single policy, namely, the traveling wave policy. These results demonstrate that the larger the target angular displacement, the more chance the machine can learn to converge to the traveling wave policy, suggesting its optimality in generating net rotation at low Reynolds number. We also note that the same trend applies to swimmers consisting of linear chains of spheres for net translation:[47] given sufficient amount of learning, the swimmers with different numbers of spheres all converge to the same type of traveling wave policy via reinforcement learning.

In Fig. 5(b), we quantify the performance of the traveling wave policy for different values of $N$ in terms of the angular displacement per cycle $\Delta\bar{\theta}_C$ and the angular displacement per cycle per stroke $\Delta\bar{\theta}_S$ (inset). As the number of sphere $N$ increases, the traveling wave policy generates more displacement per cycle $\Delta\bar{\theta}_C$. Even though the number of strokes in the traveling wave policy also increases as $2(N-1)$ machines with a higher number of spheres still produce a larger displacement per cycle per stroke, $\Delta\bar{\theta}_S = \Delta\bar{\theta}_C/2(N-1)$, as shown in the inset.

## IV. CONCLUDING REMARKS

In this work, we demonstrate the first use of reinforcement learning to generate mechanical rotation at low Reynolds numbers. This alternative approach diverges from the conventional way of prescribing a predefined sequence of strokes based on knowledge of locomotion; instead, we exploit a simple reinforcement learning algorithm ($Q$-learning) to enable a machine to identify effective rotational policies based on its interaction with the surroundings, without requiring prior knowledge of locomotion. When the machine has the minimum degrees of freedom for net rotation ($N = 3$), it recovers the strategy identified by Dreyfus *et al.* for Purcell's rotator, which shares similarity with the conformational changes of some molecular motors undergoing ATP or photochemically driven rotational movements.[23–25] For an increased number of spheres ($N > 4$), the machine is capable of identifying multiple effective policies for net rotation, depending on different learning parameters in the system. However, when sufficient learning steps are allowed, the machine eventually evolves to a single policy—the traveling wave policy. The traveling wave policy enables the machine to generate net rotation by a sequential contraction (and then expansion) of active angles in the machine. The sequence of strokes in Purcell's rotator is a special case of this family of traveling wave policies. As a remark, in this work, only one degree of freedom is allowed to change in each learning step. More effective locomotion strategies may emerge if multiple degrees of freedom are allowed to change in

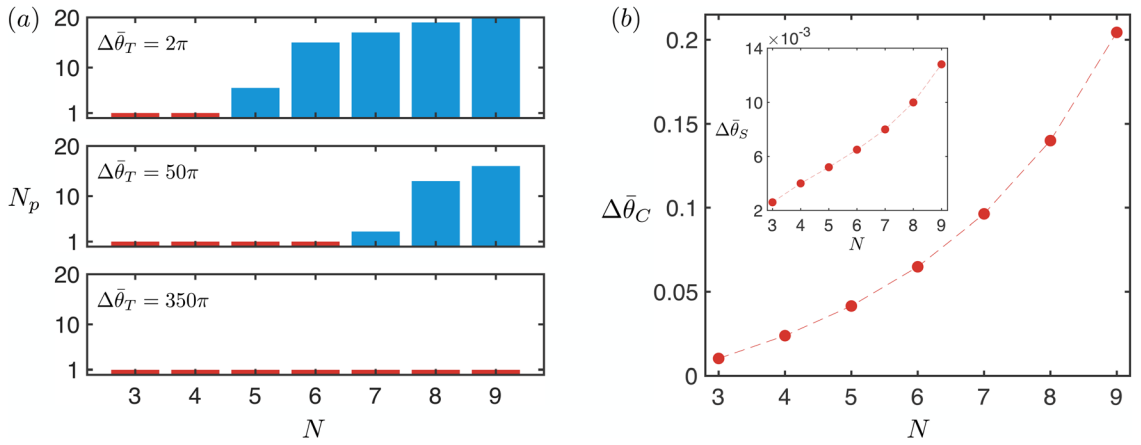


**FIG. 5.** Mechanical rotation of a $N$-sphere rotator via reinforcement learning. (a) The number of different policies $N_p$ adopted by a $N$-sphere rotator when the learning process is stopped at different values of target angular displacement, $\Delta\bar{\theta}_T$, in 20 sample runs. For each run, the machine continues to learn until the net angular rotation $\Delta\bar{\theta}_n$ reaches $\Delta\bar{\theta}_T$. With a relatively short learning process ($\Delta\bar{\theta}_T = 2\pi$; top panel), the three-sphere and four-sphere rotators converge to a single policy in all runs (red bars), which correspond to the traveling wave policies. For $N > 4$, the machine adopts a wider variety of different policies as $N$ increases (blue bars). With a longer training process ($\Delta\bar{\theta}_T = 50\pi$; middle panel), more rotators converge to the traveling wave policies at the end of the learning process (red bars) with a reduced number of policies for $N \geq 7$. With a sufficiently long learning process ($\bar{\theta}_T = 350\pi$; bottom panel) all rotators converge to the traveling wave policies. (b) Characterization of the performance of the traveling wave policies of individual $N$-sphere rotators by the net angular displacement per cycle $\Delta\bar{\theta}_C$ and the net angular displacement per cycle per stroke (inset) $\Delta\bar{\theta}_S = \Delta\bar{\theta}_C/2(N-1)$, where $2(N-1)$ is the number of strokes in the traveling wave policies. Both $\Delta\bar{\theta}_C$ and $\Delta\bar{\theta}_S$ increase with $N$. In these simulations, $\phi = \pi/(2N)$, $\gamma = 0.9$, $\varepsilon = 0.15$, and $R/L = 0.1$.

each learning step. Our preliminary studies with the three-sphere model have shown that the rotator still evolves to the same traveling wave policy even if both angles are allowed to change in one learning step, suggesting the optimality of the policy at least for the three-sphere system. A more extensive study toward this direction would be an interesting extension of this work. Finally, we also remark that the change in the angular centroid is used as the reward in reinforcement learning here based on the goal to maximize net rotation of the machine. Rewards accounting for energy consumption due to different actions may also be considered in future work for optimization based on energetic considerations. Recent works have also suggested traveling wavelike deformations to be energy-optimal strokes for locomotion.[15,50–52]

The alternative approach in this work is particularly desirable when a machine explores an environment with unknown properties or when the knowledge of locomotion remains incomplete in more complex environments. The approach based on reinforcement learning bypasses the challenging of designing locomotory gaits in advance in these situations. As a proof of concept, we adopt a standard $Q$-learning algorithm for its simplicity and expressiveness. There exists a vast potential in the use of more advanced machine learning approaches[53–59] to improve the learning performance. Taken together, this work combines with previous work on translation to lay the foundation for the use of machine learning techniques in generating more complex, three-dimensional maneuvers in future works.

## SUPPLEMENTARY MATERIAL

See the supplementary material for illustrating the effect of other learning parameters. Specifically, we illustrate that the effect of learning rate ($\alpha$) and discount factor ($\gamma$) become more apparent for systems with an increased number of spheres. Based on these findings, we set $\alpha = 1$ and $\gamma = 0.9$ in the simulations presented in the main text.

## AUTHORS' CONTRIBUTIONS

Y.L. and Z.Z. contributed equally to this work.

## ACKNOWLEDGMENTS

Funding support by the National Science Foundation (Grant No. EFMA-1830958 to O.S.P. and Grant Nos. 1614863 and 1951600 to Y.-N.Y.) is gratefully acknowledged. Y.-N.Y. acknowledges support from Flatiron Institute, part of Simons Foundation. Z.Z. and O.S.P. also acknowledge the computational resources from the WAVE computing facility (enabled by the E. L. Wiegand Foundation) at Santa Clara University.

## DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## REFERENCES

[1]L. J. Fauci and R. Dillon, "Biofluidmechanics of reproduction," Annu. Rev. Fluid Mech. **38**, 371–394 (2006).
[2]E. Lauga, "Bacterial hydrodynamics," Annu. Rev. Fluid Mech. **48**, 105–130 (2016).
[3]J. E. Simons and S. D. Olson, "Sperm motility: Models for dynamic behavior in complex environments," in *Cell Movement: Modeling and Applications*, edited by M. Stolarska and N. Tarfulea (Springer International Publishing, Cham, 2018), pp. 169–209.
[4]J. M. Yeomans, D. O. Pushkin, and H. Shum, "An introduction to the hydrodynamics of swimming microorganisms," Eur. Phys. J.: Spec. Top. **223**, 1771–1785 (2014).
[5]E. Lauga and T. R. Powers, "The hydrodynamics of swimming microorganisms," Rep. Prog. Phys. **72**, 096601 (2009).
[6]H. C. Berg, "The rotary motor of bacterial flagella," Annu. Rev. Biochem. **72**, 19–54 (2003).
[7]P. Sartori, F. V. Geyer, A. Scholich, F. Jülicher, and J. Howard, "Dynamic curvature regulation accounts for the symmetric and asymmetric beats of Chlamydomonas flagella," eLife **5**, e13258 (2016).
[8]B. J. Nelson, I. K. Kaliakatsos, and J. J. Abbott, "Microrobots for minimally invasive medicine," Annu. Rev. Biomed. Eng. **12**, 55–85 (2010).
[9]W. Gao and J. Wang, "The environmental impact of micro/nanomachines: A review," ACS Nano **8**, 3170–3180 (2014).
[10]S. J. Ebbens and J. R. Howse, "In pursuit of propulsion at the nanoscale," Soft Matter **6**, 726–738 (2010).
[11]E. M. Purcell, "Life at low Reynolds number," Am. J. Phys. **45**, 3–11 (1977).
[12]L. E. Becker, S. A. Koehler, and H. A. Stone, "On self-propulsion of micromachines at low Reynolds number: Purcell's three-link swimmer," J. Fluid Mech. **490**, 15–35 (2003).
[13]A. Najafi and R. Golestanian, "Simple swimmer at low Reynolds number: Three linked spheres," Phys. Rev. E **69**, 062901 (2004).
[14]J. E. Avron, O. Kenneth, and D. H. Oaknin, "Pushmepullyou: An efficient micro-swimmer," New J. Phys. **7**, 234 (2005).
[15]D. J. Earl, C. M. Pooley, J. F. Ryder, I. Bredberg, and J. M. Yeomans, "Modeling microscopic swimmers at low Reynolds number," J. Chem. Phys. **126**, 064703 (2007).
[16]R. Golestanian and A. Ajdari, "Stochastic low Reynolds number swimmers," J. Phys.: Condens. Matter **21**, 204104 (2009).
[17]F. Alouges, A. DeSimone, and A. Lefebvre, "Optimal strokes for low Reynolds number swimmers: An example," J. Nonlinear Sci. **18**, 277–302 (2008).
[18]F. Alouges, A. DeSimone, L. Heltai, A. Lefebvre-Lepot, and B. Merlet, "Optimally swimming Stokesian robots," Discrete Contin. Dyn. Syst. Ser. B **18**, 1189 (2013).
[19]Q. Wang and H. G. Othmer, "Analysis of a model microswimmer with applications to blebbing cells and mini-robots," J. Math. Biol **76**, 1699–1763 (2018).
[20]Q. Wang, "Optimal strokes of low Reynolds number linked-sphere swimmers," Appl. Sci. **9**, 4023 (2019).
[21]B. Nasouri, A. Vilfan, and R. Golestanian, "Efficiency limits of the three-sphere swimmer," Phys. Rev. Fluids **4**, 073101 (2019).
[22]O. Silverberg, E. Demir, G. Mishler, B. Hosoume, N. Trivedi, C. Tisch, D. Plascencia, O. S. Pak, and I. E. Araci, "Realization of a push-me-pull-you swimmer at low Reynolds numbers," Bioinspiration Biomimetics **15**, 064001 (2020).
[23]R. Dreyfus, J. Baudry, and H. A. Stone, "Purcell's "rotator": Mechanical rotation at low Reynolds number," Eur. Phys. J. B **47**, 161–164 (2005).
[24]N. Koumura, R. W. J. Zijlstra, R. A. van Delden, N. Harada, and B. L. Feringa, "Light-driven monodirectional molecular rotor," Nature **401**, 152–155 (1999).
[25]K. Kinosita, R. Yasuda, H. Noji, and K. Adachi, "A rotary molecular motor that can work at near 100% efficiency," Philos. Trans. R. Soc. London, Ser. B **355**, 473–489 (2000).
[26]F. Cichos, K. Gustavsson, B. Mehlig, and G. Volpe, "Machine learning for active matter," Nat. Mach. Intell. **2**, 94–103 (2020).
[27]A. C. H. Tsang, E. Demir, Y. Ding, and O. S. Pak, "Roads to smart artificial microswimmers," Adv. Intell. Syst. **2**, 1900137 (2020).
[28]M. Gazzola, B. Hejazialhosseini, and P. Koumoutsakos, "Reinforcement learning and wavelet adapted vortex methods for simulations of self-propelled swimmers," SIAM J. Sci. Comput. **36**, B622–B639 (2014).
[29]M. Gazzola, A. A. Tchieu, D. Alexeev, A. de Brauer, and P. Koumoutsakos, "Learning to school in the presence of hydrodynamic interactions," J. Fluid Mech. **789**, 726–749 (2016).
[30]G. Novati, S. Verma, D. Alexeev, D. Rossinelli, W. M. van Rees, and P. Koumoutsakos, "Synchronisation through learning for two self-propelled swimmers," Bioinspiration Biomimetics **12**, 036001 (2017).
[31]S. Verma, G. Novati, and P. Koumoutsakos, "Efficient collective swimming by harnessing vortices through deep reinforcement learning," Proc. Natl. Acad. Sci. U. S. A. **115**, 5849–5854 (2018).

[32]L. Biferale, F. Bonaccorso, M. Buzzicotti, P. Clark Di Leoni, and K. Gustavsson, "Zermelo's problem: Optimal point-to-point navigation in 2D turbulent flows using reinforcement learning," Chaos **29**, 103138 (2019).
[33]L. Yan, X. Chang, R. Tian, N. Wang, L. Zhang, and W. Liu, "A numerical simulation method for bionic fish self-propelled swimming under control based on deep reinforcement learning," Proc. Inst. Mech. Eng., Part C **234**, 3397–3415 (2020).
[34]Y. Jiao, F. Ling, S. Heydari, N. Heess, J. Merel, and E. Kanso, "Learning to swim in potential flow," Phys. Rev. Fluids **6**, 050505 (2021).
[35]G. Reddy, A. Celani, T. J. Sejnowski, and M. Vergassola, "Learning to soar in turbulent environments," Proc. Natl. Acad. Sci. U. S. A. **113**, E4877–E4884 (2016).
[36]G. Reddy, J. Wong-Ng, A. Celani, T. J. Sejnowski, and M. Vergassola, "Glider soaring via reinforcement learning in the field," Nature **562**, 236–239 (2018).
[37]S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, "Flow navigation by smart microswimmers via reinforcement learning," Phys. Rev. Lett. **118**, 158004 (2017).
[38]K. Gustavsson, L. Biferale, A. Celani, and S. Colabrese, "Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning," Eur. Phys. J. E **40**, 110 (2017).
[39]S. Colabrese, K. Gustavsson, A. Celani, and L. Biferale, "Smart inertial particles," Phys. Rev. Fluids **3**, 084301 (2018).
[40]E. Schneider and H. Stark, "Optimal steering of a smart active particle," EPL **127**, 64003 (2019).
[41]J. K. Alageshan, A. K. Verma, J. Bec, and R. Pandit, "Machine learning strategies for path-planning microswimmers in turbulent flows," Phys. Rev. E **101**, 043110 (2020).
[42]Y. Yang, M. A. Bevan, and B. Li, "Micro/nano motor navigation and localization via deep reinforcement learning," Adv. Theory Simul. **3**, 2000034 (2020).
[43]S. Muiños-Landin, A. Fischer, V. Holubec, and F. Cichos, "Reinforcement learning with artificial microswimmers," Sci. Robot. **6**, eabd9285 (2021).
[44]J. Qiu, W. Huang, C. Xu, and L. Zhao, "Swimming strategy of settling elongated micro-swimmers by reinforcement learning," Sci. China Phys., Mech. Astron. **63**, 284711 (2020).
[45]B. Hartl, M. Hübl, G. Kahl, and A. Zöttl, "Microswimmers learning chemotaxis with genetic algorithms," Proc. Natl. Acad. Sci. U. S. A. **118**, e2019683118 (2021).
[46]M. Mirzakhanloo, S. Esmaeilzadeh, and M.-R. Alam, "Active cloaking in Stokes flows via reinforcement learning," J. Fluid Mech. **903**, A34 (2020).
[47]A. C. H. Tsang, P. W. Tong, S. Nallan, and O. S. Pak, "Self-learning how to swim at low Reynolds number," Phys. Rev. Fluids **5**, 074101 (2020).
[48]J. Happel and H. Brenner, *Low Reynolds Number Hydrodynamics: With Special Applications to Particulate Media* (Noordhoff International Publishing, 1973).
[49]C. Watkins and P. Dayan, "Q-learning," Mach. Learn. **8**, 279–292 (1992).
[50]D. Agostinelli, F. Alouges, and A. DeSimone, "Peristaltic waves as optimal gaits in metameric bio-inspired robots," Front. Rob. AI **5**, 99 (2018).
[51]F. Alouges, A. DeSimone, L. Giraldi, Y. Or, and O. Wiezel, "Energy-optimal strokes for multi-link microswimmers: Purcell's loops and Taylor's waves reconciled," New J. Phys. **21**, 043050 (2019).
[52]E. Lauga, "Traveling waves are hydrodynamically optimal for long-wavelength flagella," Phys. Rev. Fluids **5**, 123101 (2020).
[53]M. G. Azar, R. Munos, M. Ghavamzadaeh, and H. J. Kappen, "Speedy Q-learning," in *Proceedings of the Advances in Neural Information Processing Systems* (Curran Associates, Inc., 2011), pp. 2411–2419.
[54]V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature. **518**, 529 (2015).
[55]R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in Neural Information Processing Systems 12* (MIT Press, 2000), pp. 1057–1063.
[56]J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," presented at the 4th International Conference on Learning Representation, ICLR 2016, San Juan, PR, 2–4 May 2016.
[57]J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz, "Trust region policy optimization," in *ICML* (PMLR, 2015), Vol. 37, pp. 1889–1897.
[58]J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv:1707.06347 (2017).
[59]D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *ICML* (PMLR, 2014), Vol. 32, pp. 387–395.