Hierarchical Planning with Deep Reinforcement Learning for 3D Navigation of Microrobots in Blood Vessels

Yuguang Yang, Michael A. Bevan, and Bo Li*

Designing intelligent microrobots that can autonomously navigate and perform instructed routines in blood vessels, a crowded environment with complexities including Brownian disturbance, concentrated cells, confinement, different flow patterns, and diverse vascular geometries, can offer enormous opportunities and challenges in biomedical applications. Herein, a biological-agent mimicking a hierarchical control scheme that enables a microrobot to efficiently navigate and execute customizable routines in simplified blood vessel environments is reported. The control scheme consists of two decoupled components: a high-level controller decomposing complex navigation tasks into short-ranged, simpler subtasks and a low-level deep reinforcement learning (DRL) controller responsible for maneuvering microrobots to accomplish subtasks. The proposed DRL controller utilizes 3D convolutional neural networks and is capable of learning control policies directly from raw 3D sensory data. It is shown that such a control scheme achieves effective and robust decision-making within unseen, diverse complicated environments and offers flexibility for customizable task routines. This study provides a proof of principle for designing intelligent control systems for autonomous navigation in vascular networks for microrobots.

1. Introduction

Over the past decade, there has been significant interest in engineering a wide variety of micro-/nano-scale robots for applications in health care and biomedicine. Researchers have been actively exploring possibilities to engineer novel micro/nano-robots that have customizable dynamics, bio-compatible, and multifunctional surface properties, and the ability to exploit a variety of power sources such that they can be deployed to hard-to-reach, in vivo environments for emerging applications like drug delivery and precision surgery. These robots hold the promise of realizing the bold vision from the 1966 film Fantastic Voyage—sensing a microscopic multifunctional submarine to blood vessels to cure diseases.

Microrobots that can autonomously sense the environment, make decisions, and execute instructed routines hold the potential for carrying out specialized tasks in the blood vessels including fighting against targeted biological threats, clearing blood clots and cancer cells, and diagnosing diseases from blood-related biomarkers. While algorithms, software, and hardware empowering macro-scale autonomous robots are well-developed or even well-commercialized (e.g., vacuum cleaning robots, driverless package delivery robots), the autonomous navigation of a microrobot in vascular structures, even in vitro, remains a daunting task.

From an algorithmic control perspective, there exist considerable hurdles for controlling microrobots to reliably navigate and execute instructed tasks in vessels. The difficulties stem from the disturbance from the inherent Brownian motion and external environment flow as well as from concentrated red blood cells (RBCs, 8% to 50% in volume fraction) and diverse vessel network topology and geometry acting as obstacles and traps. Other aggravating factors include limited visibility, complex unsteady flows, and unpredictable, rich dynamics of RBCs (e.g., aggregation and deformation). While there are recent successes in developing navigation strategies for an active swimmer in microscopic environments based on algorithmic or optimization approaches, they are primarily focused on relatively simple 2D navigation tasks. Still, in face of the pronounced challenges arising from navigating microrobots in blood vessels, there lacks a formal algorithm framework that controls microrobots to conduct 3D navigation and accurately execute instructed routines in large-scale unknown and unsteady environments with diverse dead-ends and traps.

Here, we present a proof-of-principle study on constructing a hierarchical control scheme as a potential solution to the 3D efficient, programmable navigation of self-propelled microrobots in simplified blood vessel environments. While such environments lack the full-scale complexity of real-world blood vessels, they
capture several key characteristics such as varying vessel geometry and rich, unknown red blood cell configurations. The hierarchical control scheme is inspired by the hierarchical problem-solving strategy in biological agents.\cite{21} Specifically, we design a high-level controller to automatically decompose a complex navigation task into simpler subtasks, which are represented by a series of navigation subgoals leading toward the ultimate goal. The high-level controller is accompanied by a low-level deep reinforcement learning (DRL) controller to maneuver robots to accomplish these subtasks. By training a DRL controller via reinforcement learning method on extensive raw 3D sensor data,\cite{22} our data-driven approach to navigation control not only simplifies sensor and algorithm development but also enables navigation in unknown, diverse blood vessel environments. The hierarchical control design offers great flexibility to customize navigation routines in large-scale, complex environments; it ultimately provides an algorithmic route to address the navigation challenges arising from a broad range of biomedical applications, such as targeted drug delivery, blood clot clearance, precision surgery, and numerous circulatory system-based disease diagnostics and therapeutics.

2. Experimental Section

2.1. Hierarchical Control Algorithm

A hierarchical control scheme is established to address the 3D navigation of self-propelled microrobots in blood vessels (Figure 1A). The proposed hierarchical control scheme consists of a high-level controller dynamically setting short-ranged navigation targets along a desired path (length scale $>100 \mu m$) (Figure 1B) and a low-level DRL controller responsible for navigating robots to circumvent RBC obstacles (length scale $<10 \mu m$) and moving toward the specified dynamic targets using local observation (Figure 1C,D). The choice of the DRL controller is motivated by its exceptional capability in sequential decision-making arising from various challenging situations such as games\cite{23,24} and robotics\cite{25} as well as the recent success in applying DRL to learn generalizable navigation strategies in 2D micro-structured environments.\cite{15,20}

A navigation task can be represented by a 3D preset path in the space connecting the starting point to the ultimate target point. The high-level controller selects a point in the path near the microrobot as a temporary target position. As the microrobot is steered by the low-level controller (Figure 1B) and gets closer to the temporary target, a new farther target along the path is selected. By following these guiding targets, the robot will approximately follow the designed path. Mathematically, let the path be represented by a parametric function $T(q) \in R^3$, then the sequentially generated targets are given by $T(q_1), T(q_2), ..., T(q_N)$, where $q_i < q_2 < ... < q_N$. $T(q_N)$ denotes the final target point or the desired path endpoint. The generation of new temporary targets is well paced with the progress that the micro-robot makes toward these temporary targets, as summarized by Algorithm 1.

The hierarchical control algorithm performs iterations on two levels. The high-level controller iteratively updates temporary short-ranged targets along the desired path as navigation subtasks. The low-level DRL controller iteratively updates the rotational decisions at an interval of $\bar{t}_1$ based on the microrobot

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{Hierarchical control scheme for autonomous microrobot navigation. A) Schematic representation (not to scale) of the low-level controller steering a microrobot to navigate in a blood vessel. Our deep reinforcement learning (DRL) algorithm employs deep neural networks to take 3D sensing of the microrobot’s neighborhood, microrobot’s state (position and orientation), and target (octahedron) location as inputs and to output rotational decisions. The details of the architecture are provided in the Experimental Section. B) Scheme of 3D local sensing around the microrobot. The sensation is represented by a 3D binary image with width $W$ and resolution (pixel size) $U$. The 3D binary image takes a value of 1 if the central point of the pixel is in a red blood cell (RBC) or is outside the vessel, and 0 otherwise. C) A target generator as a high-level controller to sequentially generate short-ranged targets (octahedrons) that guide the microrobot along a prescribed path. D) Local 3D sensory input, microrobot state (position and orientation), and target position are fed into a neural network, which outputs the rotational decisions to steer the microrobot towards the target. Here, the RBCs have diameters uniformly sampled between 6 and 8 $\mu m$, and the microrobot has a diameter of $2a = 1 \mu m$.}
\end{figure}
state and the local observation, with the objective to accomplish the navigation subtask. The target update is triggered only when the microrobot is making progress or getting closer to the target. The complete algorithm is illustrated in Algorithm 1.

A scalar parameter \(d_0\) is used as a threshold to update a new temporary target as the new subtask. Throughout this work, \(d_0\) is set at 20\(a\), which is slightly larger than the size of RBC (=12\(a\) to 20\(a\)), where \(a\) is the radius of the microrobot. Such a choice of \(d_0\) leads to a good balance of task decomposition and low-level controller learning. Setting \(d_0\) to be too small can make the temporary target set inside a red blood cell obstacle and further cause the robot to be trapped when the robot aims to circumvent the cell. In contrast, setting \(d_0\) to be too large makes the subtask harder due to its increasing state-action space. This creates a hurdle for the low-level controller to learn effective strategies and defeats the purpose of hierarchical decision-making.

Algorithm 1: Hierarchical control algorithm for microrobot navigation

Given a desired path represented by a parametric function \(T(q) \in \mathbb{R}^t\). Denote microrobot position by \(r\).

While True:
Select a temporary target on the path, denoted by \(t = T(q^*)\), where \(q^* = \text{argmin}_q ||T(q) - r|| > d_0\) and solved \(q^*\) is required to be monotonically increasing.

While the robot is not getting closer to the target:
Steering the microrobot towards the target \(r\) based on DRL policy.

End While

End While

In the following, the dynamics model of microrobots and the DRL algorithm used to derive the control policy used to accomplish navigation subtasks are discussed.

2.2. Microrobot Dynamics

In this work, a type of microrobot that is engaged in constant self-propulsion but allows continuous control of orientation via external stimuli or intrinsic features (e.g., electric[26] and magnetic fields,[9,27] light,[28–30] agent chirality,[7,31,32] flexible structure mechanics[13]) was considered. The dynamics of such a direction-controllable microrobot is given by

\[
\dot{r} = \xi_{f}(t) + v_{SP} p
\]

\[
\dot{p} = [I - p(t)p(t)^{T}] \cdot \xi_{p}(t) + w_{1} q_{1} + w_{2} q_{2}
\]

where \(r\) and \(p\) denote the position vector and the orientation vector (which is also the self-propulsion direction), respectively; \(t\) is time, and \(v_{SP}\) is propulsion speed taking a constant value; \(w = (w_{1}, w_{2})\), \(-w_{\text{max}} < w_{1}, w_{2} < w_{\text{max}}\) are the two bounded control inputs that change the self-propulsion direction in two orthogonal basis directions of \(q_{1}\) and \(q_{2}\), where \(q_{1} = e_{x} \times p\) (\(e_{x}\) is the unit vector in the z-direction) and \(q_{2} = p \times q_{1}\). Brownian translation and rotation are characterized by zero-mean independent multivariate Gaussian noise process \(\xi_{f}\) and \(\xi_{p}\) with covariance \(E[\xi_{f}(t)\xi_{f}^{T}(t')] = 2D_{f} \delta(t - t'), E[\xi_{p}(t)\xi_{p}^{T}(t')] = 2D_{r} \delta(t - t')\), where \(D_{f}\) is the translational diffusivity, \(D_{r}\) is the rotational diffusivity, and \(I\) denotes the unit tensor. All lengths are normalized by microrobot radius \(a\) and time is normalized by \(\tau = 1/D_{r}\). The control update time is \(\tau_{c} = 0.02\tau\), the integration time step \(\Delta t = 0.001\tau\), and \(D_{f} = 1.33a^{2} D_{r}\).

2.3. DRL Controller Design

Given a short-ranged target specified by the high-level controller (Figure 1B), the low-level DRL controller aims to steer the microrobot to the specified target within the minimum time. Based on the dynamics model of the microrobot, \(w = (w_{1}, w_{2})\) with \(-w_{\text{max}} < w_{1}, w_{2} < w_{\text{max}}\) are the two control inputs that change the self-propulsion direction \(p\) on two orthogonal bases. Here robot state \(s\) refers to its position \(r\) and orientation \(p\) and the system state \(\phi(s)\) is defined to include the microrobot’s state \(s\), the target position \(r\), and its local 3D observation (the 3D binary image of the microrobot’s neighborhood with a range of \(\approx 20\mu m\), double the size of a typical RBC).

To seek an optimal control policy \(\pi\) that maps the system state \(\phi(s)\) to rotational decisions \(w\), the expected reward collected during a navigation process is maximized \(E[\sum_{t=0}^{\infty} \gamma^{t} R(s_{n+1})]\) in the policy space,[14,15] where \(R\) is the instant reward function that encourages or penalizes the system states, \(\gamma\) is the discount factor, and \(n\) denotes the time step. In the DRL framework, the optimal Q\(\pi\) function associated with the reward collecting process is defined as

\[
Q^\pi(\phi(s), w) = E[R(s_{1}) + \gamma R(s_{2}) + \gamma^{2} R(s_{3}) + \ldots | \phi(s_{0})]
\]

\[
= \phi(s), w_{0} = w, \pi\]

which is the expected sum of rewards along the navigation process by following the optimal policy \(\pi^\star\), after observing \(\phi(s)\) and making a rotational decision of \(w\). Given \(Q^\pi\) function, the optimal policy \(\pi^\star\) is connected to \(Q^\pi\) via \(\pi^\star = \text{argmax}_{\pi} Q^\pi(\phi(s), \pi)\).

The navigation policy \(\pi\) is optimized through the deep deterministic gradient descent algorithm,[16] which simultaneously trains a deep neural network, called Critic network, to approximate the optimal Q\(\pi\) function, and another deep neural network, called Actor-network, to approximate the policy \(\pi^\star\) (Supplementary Materials, Figure S1, Supporting Information).

The discount factor \(\gamma\) is set to 0.99 to encourage the microrobot to seek rewards in the long run and \(R\) is set equal to 1 for all states that are within a threshold distance 1 to the target, and 0 otherwise. Both neural networks employ 3D convolution neural layers to process 3D local sensory input and a fully connected layer to process the system state. The neural network is trained extensively to estimate \(Q^\pi\) through multiple episodes of navigation in different blood environments (see Supplementary Materials for details; Figure S2 and S3, Supporting Information) to learn robust and generalizable navigation strategies in various scenarios (different RBC configurations, vessel sizes, and target locations). The code with training instructions [https://github.com/kyungyutu/DeepReinforcementLearning-PyTorch] is released.

3. Results and Discussion

3.1. Free Space Navigation

We first examine the free space navigation strategies learned by the DRL controller. Figure 2 shows the rotational speeds (normalized by the maximum allowed rotation \(w_{\text{max}}\)) parameterized by target locations. For clarity, we place the microrobot at the
origin and align its self-propulsion direction with the lab coordinate x-axis. The in-plane rotation changes the self-propulsion direction in the xy plane while out-of-plane rotation changes the self-propulsion direction in the xz plane. Analogous to typical steering, to reach the target, the microrobot constantly adjusts its propulsion direction according to the relative position of the target throughout the navigation process. Considering targets are in the xy plane, the key aspects of the navigation strategy are summarized as follows (Figure 2A,B): i) When the target is in the front, propulsion direction adjustment is achieved mainly through in-plane rotation in proportion to the angle deviation; ii) If the target locates behind the microrobot, both in-plane and out-of-plane rotation are engaged at nearly the maximum value to quickly reorient the propulsion direction.

The resulting controlled trajectories of the microrobot navigating to targets at different locations are shown in Figure 2C and Movie S1, Supporting Information, where we arrange targets on a lattice surrounding the microrobot for comprehensive testing (see the 3D scheme). For targets lying in front (i.e., r' = (30, 0, 0)), the microrobot directly navigates toward the target. For other

Figure 2. Free space navigation. A) Learned rotational decision for in-plane rotation speed \( w_1 \). B) Learned rotational decision for out-of-plane rotation speed \( w_2 \). \( w_1 \) and \( w_2 \) are normalized by the maximum rotation speed \( w_{\text{max}} \). In presenting the control policies, we fix the microrobot at the origin with orientation pointing along the +x direction. The target locations vary in the xy plane. C) Representative controlled trajectories (200 control steps or 20 \( \tau \)) of the microrobot (initially located at the origin) navigating towards different target locations. The targets are arranged like lattice, with locations of \((-30, -30, 0), (-30, 30, 0), (-30, -30, 30), \ldots, (30, 30, 30)\) (upper panel); the distance versus time between the microrobot and the target is showed at the lower panel. E,F) Representative microrobot trajectories navigating towards the different target locations with external flow speeds \( v_f = 0.5 v_{\text{sp}} \) (E) and \( v_f = 0.8 v_{\text{sp}} \) (F). The setup is the same as (D) except for a steady external flow in the x direction.
target locations that the microrobot does not initially point to, rotational actions are first engaged to quickly reorient the microrobot towards the target and thereafter used to maintain the direction against Brownian motion. In either situation, nearly straight-line trajectories are produced, suggesting the optimality of the navigation strategy. Rotational actions are engaged to correct stochastic Brownian disturbances to maintain trajectories moving toward the target.

The learned control policy enables not only rapid navigation toward the target but also stable localization around the targets upon arrival (Figure 2D). Because the propulsion is constantly engaged, after arriving at the target, the microrobot still needs to constantly adjust its orientation to maintain positions in the vicinity of the target. As the microrobot carefully hovers around the target, the microrobot orbits periodically to trace out circular trajectory patterns (radius = \( v_{SP}/w_{max} \)).

So far we have demonstrated the learned control policy under one hyperparameter setting (i.e., \( v_{SP}, w_{max} \)). Control policies under other hyperparameter settings can be obtained via a simple arithmetic transformation (Supplementary Materials, Equation (S3) and Figure S4, Supporting Information) without retraining the model. Moreover, the control policy under external flow fields can be derived accordingly by treating the system as if a microrobot navigating in absence of flow but with a change in its hyperparameter (Supplementary Materials, Equation (S4), Supporting Information). We apply a flow field in the x-direction and verified the derived control policies (Figure 2E,F, Supplementary Materials, Figure S5, Supporting Information). Despite the adversarial impact of external fluid flow, the microrobot still eventually reaches prescribed targets located at different locations. The external fluid flow asymmetrically affects the microrobot motion: it speeds up the microrobot when the microrobot travels along the flow direction but slows down the microrobot if the microrobot travels against it. Therefore, the controlled navigation trajectories no longer resemble straight lines but are bent toward the flow direction, which is also predicted by the theoretical optimal trajectory of micro-swimmers in simple flows. Particularly when the magnitude of the flow increases to \( v_{f} = 0.8v_{SP} \), the trajectories are strongly bent as the microrobot is struggling toward the target. The presence of flow fields also causes delayed arrivals when microrobots travel against the flow as well as additional disturbance to the localization process (Supplementary Materials, Figure S5, Supporting Information). The radii of the hovering trajectories are significantly larger than the ones when fluid flow is absent. It is important to note that when \( v_{f} \) is greater than the propulsion speed \( v_{SP} \), the microrobot is no longer controllable.

### 3.2. Navigation in Blood Vessels

Navigation in blood vessels meets additional challenges as biconcave RBCs and vessel walls can act as traps and barriers. As a first step to evaluate the learned navigation strategy, we consider steering microrobots in a simple blood vessel environment with a few RBCs (Figure 3A,B). We arrange targets at different locations as in the free navigation test in Figure 2C and examine if the steered microrobot can circumvent RBC obstacles blocking its way. As shown in the representative trajectories in Figure 3A, when there is no RBC blockage in the way, the microrobot follows nearly the ideal straight-line path to the target as in the free-space navigation; In contrast, when an RBC is blocking the direct path, the microrobot will adjust its propulsion direction to get around the RBC. After its arrival, the microrobot employs similar localization strategies around the target as in free space navigation. To investigate the impact of the vessel wall confinement on navigation, we perform a similar evaluation near the vessel. As shown in Figure 3B, the microrobot successfully arrive at all the targets near a curved vessel wall. Particularly, when a near-wall RBC is blocking the path to the target, the microrobot will adjust its propulsion direction to circumvent the RBC and simultaneously avoid colliding with the vessel wall.

Now we evaluate the robustness, generalization, and efficiency of navigation strategies in more realistic blood environments in Figure 3C–K, which have typical sizes of arteries or veins and different RBC volume fractions. The major assumption for these blood model environments is that the blood flow is spatially uniform rather than turbulent, such that all objects in the blood flow have similar drifting speeds and appear to be still relative to each other. We also assume that self-propelled robots move much faster than RBCs and that RBCs appear effectively static.

We randomly place RBCs with different configurations (position and orientation) and sizes (uniformly sampled between diameters 6 to 8 \( \mu m \)) to create an unseen blood environment to test the generalization of the learned strategies. The high-level controller sequentially generates temporary targets to guide the microrobot to follow a straight path at the axis of vessels extending from the bottom to the top (Algorithm 1). Robots can navigate through the vessels by circumventing all RBCs in the way (Movie S2 and S3, Supporting Information). Since RBC configurations are randomly generated and are unseen in the training stage of the neural network, this test suggests that the neural network have learned a generalizable navigation strategy.

We further quantify the navigation performance in blood vessels by calculating the mean travel distance \( (L) \) versus the mean time \( t \) when we set the target at the end of the vessel in Figure 3C. As a benchmark, in a deterministic limit, the theoretical optimal deterministic performance is given by \( (L) = v_{SP}t \). A rough linear relationship indicates that the microrobot can navigate through a different portion of the vessel with a similar speed while the configuration of RBC varies. Particularly, in the free space navigation case, the navigation speed achieves \( \approx 90\% \) of the optimal deterministic speed. In general, microrobots transport faster in vessels with a larger radius and fewer RBCs. When the sizes of vessels are the same, more RBCs lead to a frequent adjustment of orientation and therefore slow down navigation. At similar RBCs concentrations, more confinement in small vessels produces additional difficulties for microrobots to get around RBC obstacles, and therefore leads to slower navigation.

As a further test of robustness and generalization of learned navigation strategies, we also examine the navigation in curved vessels with varying diameters (Figure 3J–L, Movie S3, Supporting Information) from bottom to top. Surprisingly, while microrobots are only trained in cylindrical blood vessels, the generalization of DRL controllers enables successful navigation in
curved vessels. Here we note that while we have achieved impressive performance across different blood environments using a single neural network, additional performance gains can be expected if the neural network is further fine-tuned to a specific blood environment, which is a topic of future studies.

The aforementioned results assume that the RBCs and the microrobots are experiencing the same ambient flows, and therefore the RBCs appear stationary with respect to the microrobot. An extra robustness test is to allow the microrobots to experience an additional external flow field with speed $v_f$. We find that microrobots are capable of arriving at targets when the external flow speed $v_f$ is small ($v_f \leq 0.5 v_{sp}$) and RBCs are dilute (e.g., 5%) via a simple control policy remapping (Supplementary Materials, Figure S6, Equations (S3) and (S4), Supporting Information).

### 3.3. Exhaustive Spatial Survey in Blood Vessels

We have demonstrated that the present hierarchical DRL controller can steer the microrobot towards specified targets in both RBC-absent and RBC-present environments. To further illustrate that our hierarchical control scheme allows controlled navigation according to a preset routine, we consider the problem of steering a microrobot to exhaustively survey a blood vessel, analogous to a vacuum robot cleaning a room. The capability to quickly and completely survey a blood vessel is crucial for applications such as deploying robots to search hard-to-reach regions and clean sparse hidden biological threats (e.g., cancer cells, toxins, etc.), or to rapidly release and mix drugs in complex environments.

Here we consider steering the microrobot to closely follow a predefined path $T: (x(q), y(q), z(q))$ given by a parametric function

---

**Figure 3.** Navigation trajectories of a controlled microrobot in blood vessels. A,B) Representative controlled trajectories of the microrobot navigating towards different targets when the microrobot is initially placed at the center of the vessel (A) or near the boundary of the vessel (B). The vessel has a diameter of 50 μm. C–I) Testing results microrobot navigation performance in blood vessels with different vessel diameters $D$ and RBC volume fractions $f$. Microrobots are navigating from the bottom to the top, spanning 500μm (250 μm). Specifically, D) free space baseline. E) $D = 12$ μm, $f = 10%$; F) $D = 25$ μm, $f = 10%$; G) $D = 50$ μm, $f = 5%$; H) $D = 50$ μm, $f = 10%$. I) shows the magnified view of the trajectory in (H). Mean traveled distances versus time at navigation scenarios (D–H) are collected in (C). J,K) Example navigation of microrobot in curved blood vessels with varying cross-section diameter. L) is the magnified view of the trajectory in (K).
\[
\begin{align*}
    x &= R_0 \cos(k_2 q) \cos(k_3 q) \\
    y &= R_0 \cos(k_2 q) \sin(k_3 q) \\
    z &= k_1 q
\end{align*}
\]  
(Equation (3)). In all blood environments, we observe a rough linear relationship in elevation versus time, indicating microrobots are making constant progress in this task. With 0%, 5%, and 10% RBCs in the vessels, the elevation speeds are 85.9%, 73.9%, and 64.2%, respectively, of the optimal speed, as a result of slowdowns caused by more RBC blockages.

Another quantification of the routine execution quality is measured by the distance between the 3D preset path \( \mathbf{T} \) (Equation (3)) and the actually executed path \( \mathbf{r} \) after appropriate alignment. Particularly, we can define point-wise deviation between the two paths at an arbitrary point \( q \) by

\[
\Delta(q) = \| \mathbf{r}(q) - \mathbf{T}(q') \| 
\]  
(Equation (4)).

where \( q' \) is the corresponding optimal alignment in \( \mathbf{T} \) computed using the dynamic time warping algorithm in MATLAB. The mean deviation between the two paths is given by averaging enough sample points within the paths (Supplementary Materials). As shown in Figure 4F, the pointwise deviation \( \Delta \) at different elevations for all cases are fluctuating around increasing mean deviation of 4.8a, 6.0a, and 7.2a in blood environments with 0%, 5%, and 10% RBCs. With increasing RBCs, occasional spikes in \( \Delta \) are more frequent since microrobots are going the extra way to get around RBCs. Overall, our control scheme can enable microrobots to execute preset surveying routines within different microstructured environments with high fidelity. Moreover, by modifying the preset routine path defined by

**Figure 4.** Exhaustive spatial survey of blood vessel. A) The predefined path that aims to exhaustively patrol a vessel. B–D) Controlled trajectories following the preset path to exhaustively patrol the vessels. B) zero RBCs, C) \( \approx \)5% RBCs, D) \( \approx \)10% RBCs. Upper panel: 3D view; Lower panel: Top view. E) Elevation \( z \) versus time of patrolling trajectories in different blood environments. F) Pointwise path distance \( \Delta \) versus elevation \( z \) in different blood environments (Equation (4)).
Equation (3), different surveying strategies such as adaptive exploration in vessels of varying sizes can be implemented.

3.4. Model Analysis

We now analyze what has been learned in the decision-making module enabled by DRL to understand the navigation performance in the aforementioned tasks. In a toy blood environment (Figure 5A), we apply the $t$-distributed stochastic neighbor embedding ($t$-SNE) algorithm\textsuperscript{[38]} to embed the learned representations of randomly sampled states into a 2D plane and color each point by the state value given by

$$V(s) = \max_w Q^*(\phi(s), w) = E \left[ \sum_{n=0}^{\infty} \gamma^n R(s_{n+1}) | \phi(s_0) = \phi(s), \pi \right]$$

(5)

The state value provides information on if one state $s$ is favorable to another state; a higher $V$ indicates the controlled micro-robot can arrive sooner than states with a lower $V$.

We consider five configurations (I)–(V) in Figure 5 to examine how the network perceives different situations. As shown in Figure 5A, high dimensional system states are embedded in the 2D plane apparently based on the shortest path distance to the target location, with closer states on the right. For example, configuration (I) with the closest distance to the target has its embedding on the right. Similarly, in configurations (II) and (IV), where the microrobot in (II) gets blocked by the RBC and has to reorient to get to the target and the microrobot in (IV) without RBC blockage in the way, the two configurations get assigned similar value. Additionally, in configurations (III) and (V), a microrobot in (V) gets blocked by two RBCs in the way but the configuration is evaluated to have a similar state value to (V), where the microrobot is not blocked by any RBCs. We hypothesize that the neural network can implicitly estimate the shortest paths based on local sensor information and the target position, and uses this estimate to guide the rotation decision to follow these shortest paths. To validate this hypothesis, we use the Dijkstra algorithm to estimate the shortest path distance from each state to the target. Under this hypothesis, the shortest path distance can provide state value estimation in Equation (5) via

$$V_{est}(s) = \gamma L_s / v_{SP}$$

(6)

where $\gamma$ is the discount factor used in Equation (5), $L_s$ is the shortest path length from the micro-robot’s position to the specified target, and $L_s / v_{SP}$ is the number of control steps needed to move along the shortest path. The similarity in the learned state value function and the estimated one (Figure 5B) suggests the micro-robot has acquired nearly optimal navigation strategies; this is,
making rotation decisions to follow approximate shortest paths. Although we never explicitly consider this information in the development of our model, the orientation rotating to follow the shortest path emerges after deep reinforced learning of extensive navigation data.

4. Conclusions

We have presented a proof-of-principle study on designing a hierarchical control scheme to solve the complex microrobot navigation problem in blood vessels. The integration of a low-level DRL controller and a high-level controller enables the customization of navigation routines beyond simple navigation scenarios considered previously.\textsuperscript{15,20} While we do not attempt a fully modeling of the realistic complexity of the blood environment (e.g., nonsteady blood flow, RBC deformation,\textsuperscript{19,40} and hemodynamics\textsuperscript{[41,42]}), we aim to emphasize the key idea of local sensing, hierarchical decision-making, and data-driven learning.

We show that a 3D sensing of the local environment together with DRL-based control can learn robust and efficient navigation strategies in blood vessels with diverse RBCs and varying vessel geometries. We further demonstrate that the hierarchical control scheme can steer robots to efficiently and reliably accomplish preset spatial survey routines in blood vessels. Finally, we illustrate that the neural network can learn effective representations of observations that underpin successful navigation performance. Our results not only demonstrate a general data-driven control scheme to enable navigation in human blood vessels, but also lay the foundation for achieving more sophisticated nano/microrobot autonomy in an ample spectrum of complex environments, either in vivo or in vitro.

Our control framework can be applied in experimental settings\textsuperscript{[21]} as well as extended in other computational studies. The data-driven nature of our approach lowers the hurdle of sensing and control algorithm development, offering an end-to-end approach that maps raw sensor data to decisions. The highly decoupled nature of our control scheme also allows the modification of different low-level control modules for different purposes, including adapting the controller for specific robots and motors and accommodating experimental measurement errors using additional state estimation components (e.g., Kalman filter). The proposed algorithm can be combined with the high-fidelity blood physics simulator of the true blood to learn control strategies in realistic blood environments. Similarly, our control scheme also applies to a broad class of micro-robots in other navigation scenarios like the urinary tracts or the eyes in the human body or 3D porous media for environmental applications. The high-level controller can also be extended from the rule-based one considered here to more generic learning-based ones,\textsuperscript{[42]} which enable data-driven task decomposition and opportunities for joint optimization with low-level controllers. A further extension could include controlling a swarm of microrobots via single-agent control paradigm\textsuperscript{[43]} or multiagent stochastic control paradigm\textsuperscript{[44,46]} to achieve swarm intelligence for more complicated tasks such as capturing circulating tumor cells in the blood.\textsuperscript{[47,48]}

Supporting Information

Supporting Information is available from the Wiley Online Library or from the author.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 11961131005, 11922207, and 11921002). The Interactive Supporting Information of this article can be found as DOI: 10.22541/au.165925038.87647919/v1.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

B.L.: supervised the project; Y.Y.: performed theoretical modeling and simulations; Y.Y., M.A.B., and B.L.: discussed the results; Y.Y., M.A.B., and B.L.: wrote the manuscript.

Data Availability Statement

The data that support the findings of this study are available in the supplementary material of this article.

Keywords

artificial intelligence, autonomous navigation, deep reinforcement learning, microrobot

Received: June 18, 2022
Revised: July 30, 2022
Published online: September 26, 2022
[32] Q.-L. Lei, M. P. Ciamarra, R. Ni, Sci. Adv. 2019, 5, eaay7679.
[33] T. Li, J. Li, K. I. Morozov, Z. Wu, T. Xu, I. Rozen, A. M. Leshansky, L. Li, J. Wang, Nano Lett. 2017, 17, 5092.
[34] Y. Yang, M. A. Bevan, ACS Nano 2018, 12, 10712.
[35] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, USA 1998.
[36] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, arXiv preprint arXiv:150902971, 2015.
[37] MATLAB and Statistics Toolbox, The MathWorks, Inc., Natick, MA 2018.
[38] L. van der Maaten, G. Hinton, J. Mach. Learn. Res. 2008, 9, 2579.
[39] J. L. Mcwhirter, H. Noguchi, G. Gompper, Proc. Natl. Acad. Sci. U.S.A. 2009, 106, 6039.
[40] R. Skalak, P. I. Brønmark, Science 1969, 164, 717.
[41] Y. C. Fung, Biomechanics: Circulation, Springer Science & Business Media, Berlin 2013.
[42] T. D. Kulkarni, K. R. Narasimhan, A. Saeedi, J. B. Tenenbaum, 30th Conf. on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain 2016.
[43] L. Yang, J. Jiang, X. Gao, Q. Wang, Q. Dou, L. Zhang, Nat. Mach. Intell. 2022, 4, 480.
[44] Y. Yang, Stochastic modeling and optimal control for colloidal organization, navigation, and machines, Thesis, Johns Hopkins University, 2017.
[45] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, et al., Nature 2019, 575, 350.
[46] M. A. Grover, D. J. Griffin, X. Tang, Y. Kim, R. W. Rousseau, J. Process Control 2020, 88, 32.
[47] T. H. Kim, Y. Wang, C. R. Oliver, D. H. Thamm, L. Cooling, C. Paolelli, K. J. Smith, S. Nagrath, D. F. Hayes, Nat. Commun. 2019, 10, 1478.
[48] Y. Yang, M. A. Bevan, Sci. Adv. 2020, 6, eaay7679.