Navigation of inertial forces driven mini-robots using reinforcement learning

Piyabhum Chaysri
Computer Science and Engineering
University of Ioannina
Ioannina, Greece
pchaysri@cs.uoi.gr

Konstantinos Blekas
Computer Science and Engineering
University of Ioannina
Ioannina, Greece
kblekas@cs.uoi.gr

Kostas Vlachos
Computer Science and Engineering
University of Ioannina
Ioannina, Greece
kostaswl@cse.uoi.gr

***Abstract*—In this paper we propose a reinforcement learning (RL) framework for the autonomous navigation of a pair of mini-robots that are driven by inertial forces. The inertial forces are provided by two vibration motors on each mini-robot which are controlled by a simple and efficient low-level speed controller. The action of the RL agent is the direction of the velocity of each mini-robot, and it based on the position of each mini-robot, the distance between the mini-robots, and the sign of the distance gradient. Each mini-robot is considered as a moving obstacle to the other that must by avoided. We have introduced a suitable reward function that results into an efficient collaborative RL approach. A simulation environment is created using the ROS framework, that include the dynamic model of the mini-robot and of the vibration motors. Several application scenarios are simulated, and the presented results demonstrate the performance of the proposed framework.**

***Index Terms*—reinforcement learning, moving obstacles avoidance, mini-robots, autonomous navigation**

I. Introduction

The design of micro/mini robotic platforms has become a very active field of research, with several areas of application, such as microsurgery, micro-manufacturing, and micro-assembly. The MINIMAN micro-robot, presented in [12], is based on the stick-slip principle. The impact drive principle, a variant of stick-slip principle, is employed by the 3DOF micro-robotic platform Avalon [3]. A different motion mechanism based on piezo-tubes is utilized by the Nano Walker micro-robot [4]. MiCRoN is a micro-robot, employing piezoelectric actuators with an integrated micro-manipulator [2]. The centralized control architecture of MiCRoN is presented in [15]. Kilobot, a low-cost robot designed for testing and validating algorithms for a swarm of robots, is presented in [11]. AMiRo, a modular robot platform that can be easily extended and customized in hardware and software is presented in [6].

Although piezoelectric actuators provide the required positioning resolution and actuation response, they usually suffer from complex, expensive, and cumbersome power units. Small-scale piezoelectric drivers and amplifiers that could be accommodated on board are custom made and thus do not allow for cost effective designs [7]. A simple and autonomous mini-robot (with dimensions of a few centimeters), driven by two vibration motors that is able to perform translational and rotational sliding with micrometer positioning accuracy, is presented in [16]. In [18] the formulation and practical implementation of positioning methodologies for the same mini-robot that compensate for the nonholonomic constraints are presented.

Reinforcement Learning (RL) aims at controlling an autonomous agent in unknown stochastic environments [13]. Typically, the environment is modelled as a Markov Decision Process (MDP), where the agent receives a scalar reward signal that evaluates every transition. The objective is to maximize its long-term profit that is equivalent to maximizing the expected total discounted reward. Thus the learning process is designed on selecting actions with the optimum expected reward. Discovering optimal policy for agent is conducted with the notion of *value function* which associates every state with the expected discounted reward when starting from this state and all decisions are made following this policy. Q-learning algorithm [19] that belongs to the temporal difference familty of methods constitutes one of the most popular mechanisms for building a RL agent among other [14].

In the literature there are several works with RL application to various robotic platforms, such as a marine robotic platform presented in [1]. A recent work is presented in [9] using a pair of AMiRo mini-robots with various sensor modalities that employs a RL-based distributed sensing framework based on latent space from multi-modal deep generative models.

In this paper, we present the development of a reinforcement learning framework with the goal to simultaneously navigate a pair of mini-robots toward known targets. Each mini-robot is considered, from the other, as a moving obstacle that must by avoided. A capable state space and reward function are introduced in order to build an efficient collaborative reinforcement learning framework under an unknown dynamic environment. Simulation examples, for various scenarios, are presented that show the effectiveness of the proposed framework. The output of the reinforcement learning framework is the desired velocity for each mini-robot. The required forces for the realization of the commanded velocities are provided by two vibration motors on each mini-robot. The dynamic model of the mini-robot, including the vibration motors dynamics, is integrated into the simulation environment. In order to compensate for unknown disturbances, and improve the motion resolution and the bandwidth of the mini-robot, a simple and low-cost PI

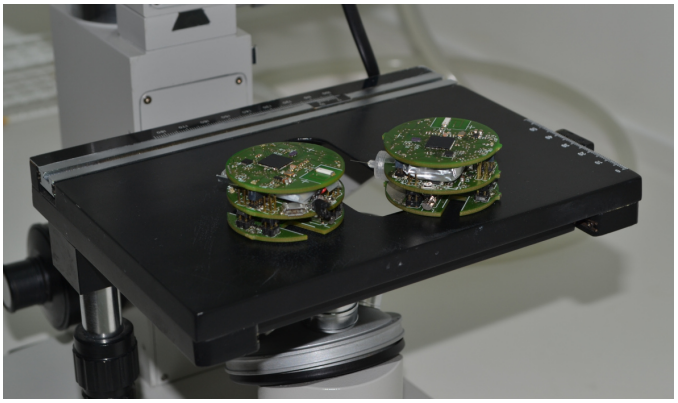


Fig. 1. Example application: Mini-robots under a microscope (photo by P. Vartholomeos)

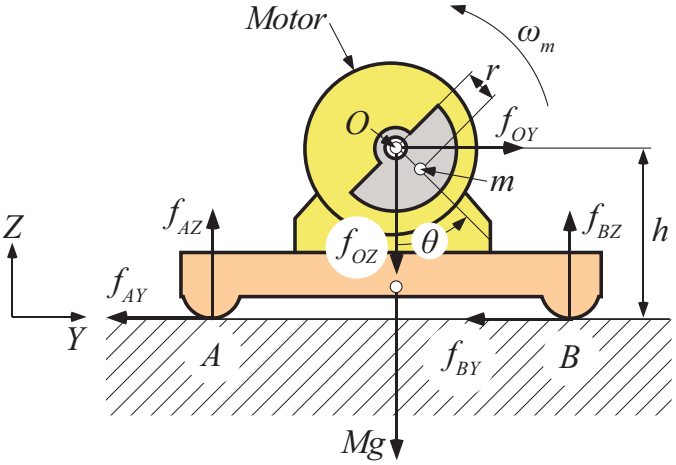


Fig. 2. Motion principle of a 1-DOF mini-robot with eccentric rotating mass (figure from [17])

speed controller for each vibration motor is implemented.

II. Application and Mini-Robot

A. Motivation and example application

Although the proposed reinforcement learning framework has a wide range of possible applications, the motivation for our work is the navigation of the mini-robot, presented in [16], capable of micrometer positioning on a plane. The developed system is a low-cost, tetheredless, fully autonomous, mini-robotic platform that can perform micro-manipulation and microassembly tasks, such as the cooperative fabrication of micro-systems or manipulation of biological specimens, in a micro scale environment, see [17]. In this paper, our application scenario would be to use a pair of these mini-robots to perform cell-manipulation activities under a microscope. To illustrate the application scenario, Fig. 1 shows a pair of such mini-robots into the workspace of a microscope. The goal is to navigate the mini-robots to predefined and known positions within the range of the microscope, avoiding a collision between the mini-robots, under the assumption of an unknown environment, see Fig. 5.

B. Brief description of the mini-robot

A brief description of the mini-robot and its motion principle is given here. For a more detailed presentation of the dynamics, design, and innovative actuation principle of the mini-robot, see [16].

1) Motion principle: The motion principle of the mini-robot is presented using the simplified 1-DOF platform, of mass M, depicted in Fig. 2. It employs a mini-motor mounted at point O with an eccentric mass m. The rotation of the eccentric mass results in gravitational and centripetal forces resolved along the Y- and Z-axes according to:

$$\begin{aligned} f_{OY} &= mr\omega_m^2 \sin\theta \\ f_{OZ} &= -mg - mr\omega_m^2 \cos\theta \end{aligned} \tag{1}$$

where θ is the rotation angle of the eccentric mass, and ω_m is the rotational velocity of the mini-motor. The acceleration of

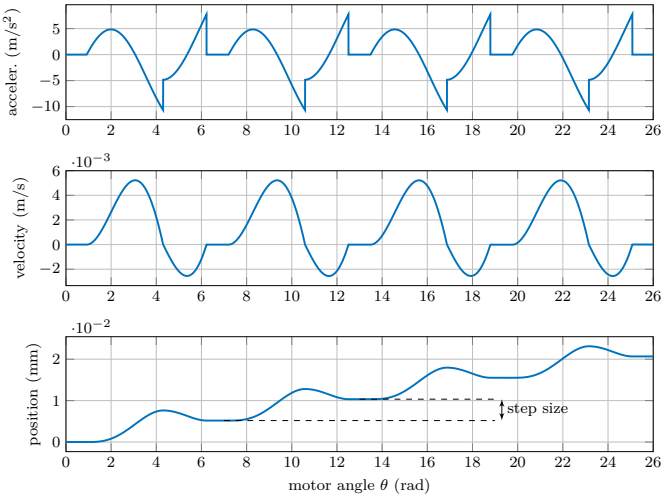


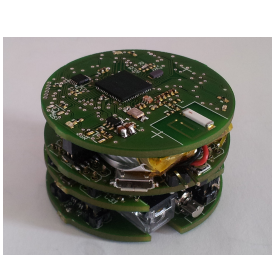
Fig. 3. Simulation results for the motion of a 1-DOF mini-robot

the gravity is denoted with g, and r is the eccentricity of the rotating mass.

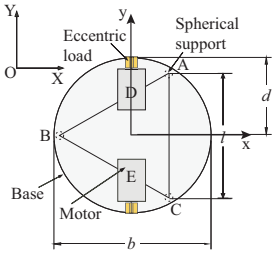
Above a critical value of the rotational velocity of the mini-motor, the actuation forces overcome the frictional forces at the contact points A and B, and the platform slides. Simulated results of the platform's trajectory are depicted in Fig. 3, where it is shown that during one cycle of operation, i.e. the eccentric mass has described an angle of 360°, the platform exhibits a net displacement in the Y-axis. The magnitude of the net displacement (step size) depends on the rotational velocity of the mini-motor [16].

2) Dynamics of the mini-robot: An older prototype of the mini-robot is shown in Fig. 4a, see [17]. A new version of the mini-robot, based on the same actuation principle, is under construction. Some physical parameters of the mini-robot are presented in Table I.

The dynamic model of the mini-robot is described by the



(a)



(b)

Fig. 4. (a) Prototype; (b) Design concept (top view) (figures from [17])

TABLE I
PHYSICAL PARAMETERS OF THE MINI-ROBOT

Parameter	Value
Mini-robot mass	0.1 kg
Mini-robot diameter	0.05 m
Mini-robot height	0.045 m
Vibration motor body diameter	0.0044 m
Vibration motor body length	0.0102 m
Vibration motor axis height	0.003 m

following equations:

$$\begin{aligned} \mathbf{M}\dot{\mathbf{v}} &= \mathbf{R}\sum_i {}^b\mathbf{f}_i \\ {}^b\mathbf{I}\dot{\boldsymbol{\omega}}_p + {}^b\boldsymbol{\omega}_p \times {}^b\mathbf{I}{}^b\boldsymbol{\omega}_p &= \sum_i \left({}^b\mathbf{r}_i \times {}^b\mathbf{f}_i\right) + \sum_j {}^b\mathbf{n}_j \end{aligned} \tag{2}$$

where $i = \{A, B, C, D, E\}$, and $j = \{D, E\}$, see Fig. 4b. In (2), $\mathbf{M}$ denotes the mass of the mini-robot, $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}]^\top$ is the linear velocity of the center of mass of the mini-robot, and $\mathbf{R}$ is the rotation matrix from the body frame, $\{b\}$, to the inertial frame. ${}^b\mathbf{f}_i$ is a vector that includes the actuation forces generated by the two vibration motors and the friction forces at the three contact points of the mini-robot, and ${}^b\mathbf{n}_j$ includes the moments exerted by the vibration motors. The moment of inertia of the mini-robot is denoted by $\mathbf{I}$, and $\boldsymbol{\omega}_p$ is the angular velocity of the mini-robot. Finally, ${}^b\mathbf{r}_i$ is the position vector of point i expressed in the body frame. The actuation forces generated by each vibration motor when its eccentric load rotates are given by the following equations:

$$\begin{aligned} {}^b f_{jX} &= (mr\ddot{\theta}\cos\theta - mr\dot{\theta}^2\sin\theta)\sin\phi_j \\ {}^b f_{jZ} &= -mg - mr\ddot{\theta}\sin\theta - mr\dot{\theta}^2\cos\theta \end{aligned} \tag{3}$$

where $\phi_j \in \{90°, -90°\}$ is the angle between the motor axis and the X-axis of the body frame, see Fig. 4b.

III. REINFORCEMENT LEARNING FOR MINI-ROBOTS NAVIGATION

The Reinforcement Learning (RL) agent constitutes the basic building block of the proposed decision support system. The RL agent receives a state related to the position of the mini-robot and performs an action which corresponds to the direction of its velocity. Note that the desired magnitude of the mini-robots velocity is constant and not affected by the RL agent.

The RL framework can be formally described as a *Markov decision process* (MDP) given by a five-tuple $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$:

- $\mathcal{S}$ denotes the set of agent's states. In our case we have considered the mini-robot inertial coordinates. i.e. $s = (x, y)$ as the states of the agent. However, as it will be shown later the state vector will be enriched with other features in the case of having a pair of mini-robots.
- $\mathcal{A}$ is the set of possible agent's actions. We have considered eight (8) discrete values: $\mathcal{A} = \{0^o, 45^o, 90^o, 135^o, 180^o, 225^o, 270^o, 315^o\}$ that correspond to eight different directions of mini-robot velocity. It must be noted that the selected size of the action set seems to be enough in our application with satisfactory performance. However, any other set of actions can be also used.
- T denotes the state transition function where $T(s, a, s')$ specifies the probability $P(s'|s, a)$ of visiting a new state s' from state s by taking action a. In this study we consider deterministic transitions for simplicity.
- $R : \mathcal{S} \to \mathbb{R}$ is the reward function for a state-action pair, $R(s, a)$, describing the immediate reward of executing action a in state $\boldsymbol{s}$, and finally,
- γ is the discount factor range in $[0, 1]$ used for γ-discounted future rewards.

Policy $\pi : \mathcal{S} \to \mathcal{A}$ constitutes the decision mechanism of the agent. It reflects a map from state space to actions and it is designed so as to maximize the expected sum of future rewards (called return). This is denoted by the state-action *Q-value* function, $Q(\boldsymbol{s}, a)$ which describes the expected discounted reward received by starting from state s, executing the action a and following the policy π:

$$Q^\pi(\boldsymbol{s}, a) = E_\pi\{R_t|\boldsymbol{s}_t = \boldsymbol{s}, a_t = a\} = E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}\right\} \tag{4}$$

Training an RL agent can be made by minimizing the Bellman's equation error given by:

$$\mathcal{E}_t(\theta) = \frac{1}{2}E_{s\sim T, a\sim\pi}\|R(\boldsymbol{s}_t, a_t) + \gamma\max_a Q(\boldsymbol{s}_{t+1}, a) - Q(\boldsymbol{s}_t, a_t)\|^2 \tag{5}$$

The objective of RL is to estimate optimal policy π^* by choosing actions that yield the appropriate action-state value function, i.e.

$$\pi^*(\boldsymbol{s}) = \arg\max_{a\in\mathcal{A}} Q^\pi(\boldsymbol{s}, a) \tag{6}$$

The term "optimal" is used to describe the shortest path and the minimum required rotation of each mini-robot to achieve the commanded direction.

Q-learning [19] is a convenient model-free methodology used for learning RL agents that offers easy off-policy temporal different control. It employs the following policy update rule:

$$Q(\boldsymbol{s}, a) \leftarrow Q(\boldsymbol{s}, a) + \eta\left[r + \gamma\max_{a'} Q^\pi(\boldsymbol{s}', a') - Q(\boldsymbol{s}, a)\right] \tag{7}$$

where η is the learning rate. A typical value of this parameter used in all experiments is $\eta = 0.001$.

The reward signal, r, for a single mini-robot is defined as following:

$$r = \begin{cases} +L & , \text{ if it reaches goal} \\ -L & , \text{ if it is found out of the board} \\ -1 & , \text{ otherwise} \end{cases} \quad (8)$$

where L is a constant term (a typical value is $L = 100$).

The learning procedure is episodic. Every episode typically starts at a random position on the border of the board. Then, the agent performs an exploration of the environment by visiting a sequence of states (making a sequence of transitions) and interacting with the environment. An important issue in reinforcement learning is how to manage the trade-off between exploration and exploitation since it may have significant impact to the quality of learned policy. A common choice is to employ the ϵ-greedy exploration scheme, where at each time step t an action is selected greedily, based on the estimated action-value function with probability $1 - \epsilon_t$, while a random action is chosen with probability ϵ_t, ($\epsilon_t \in [0, 1]$).

Every episode terminates either when the agent reaches the goal state, or when the mini-robot collides with the workspace limits. Note that in this application we consider that there is no physical obstacle in the workspace (board) so the agent is free to explore through the entire board without any obstruction. However, extending our study in environments with obstacles remains one of our future plans.

A. Navigation of a pair of mini-robots

Our main goal is to develop a reinforcement learning framework of more than one mini-robots which are jointly interacting in a collaborative environment. In the current work we have considered a pair of them. Each mini-robot must learn a sub-optimal policy in order to reach its own goal position under the presence of the other. As a consequence of this

- the goal position of the second mini-robot is operated as an additional *static* obstacle, and
- any mini-robot is considered as a *dynamic* obstacle for the other which must be avoided during the navigation process.

The workspace is assumed to be a rectangular flat surface divided into cells. Both agents have no prior knowledge of it and their starting position differs at each episode. During our experimental study we assume different scenarios that vary according to the targets' position and the starting positions of both mini-robots. The episode ends when both mini-robots reach their goals, or at least one of them fails, i.e. it goes out of board or collides with an (static or dynamic) obstacle. When a mini-robot reaches its goal position, it remains there until the other terminates. Various exemplary scenarios are presented in Fig. 5. The robot targets are located in the central region of the board having enough space among them so as robots are able to navigate without collisions.

As in the case of a single mini-robot, the state space of every mini-robot $i \in \{1, 2\}$ consists of its inertial coordinates (x_i, y_i). In addition it includes the next two features:

- the distance d between both robots (center of each mini-robot) which has been discretized into three (3) values:

$$d = \begin{cases} \text{“near”} & \text{if } 5\ cm < distance \leq 6\ cm \\ \text{“medium”} & \text{if } 6\ cm < distance \leq 10\ cm \\ \text{“large”} & \text{if } distance > 10\ cm \end{cases} \quad (9)$$

  The minimum limit of the “near” value in (9) is justified by the fact that the diameter of each mini-robot is equal to 5 cm, see Table I.
- the sign of the distance gradient, g, that indicates whether the distance between the mini-robots is increased or decreased:

$$g = \text{sign}(distance^{(t)} - distance^{(t-1)}) \quad (10)$$

  in each time step, t.

The proposed set of mini-robot's features: $s_i = (x_i, y_i, d, g)$ allows the agent to decision taking under consideration the presence of moving or static obstacles in its neighborhood, and build a policy so as to avoid them. The last two features, (d, g), play a significant role to the calculation of the reward and must specify the strategy that both mini-robots must follow so as to avoid collision. The following reward function has been considered:

$$r = -b\left(1 - \frac{\min(distance, D)}{D}\right) - 1.0 \quad (11)$$

where

- D is the maximum value of *distance*. Above this value the distance between two mini-robots has not influence on the reward function (typical value of this threshold is $D = 10$).
- $$b = \begin{cases} c & \text{if } g < 0 \\ 1.0 & \text{if } g \geq 0 \end{cases} \quad (12)$$

  This coefficient penalizes more ($c \geq 1$) the situation when two mini-robots coming closer.

Finally, as in the case of a single mini-robot, the reward is constantly positively large ($+L$) when the mini-robot reaches its goal, and negatively large ($-L$) when it finds an obstacle or is located outside the grid border.

IV. Vibration motor model and speed controller

Each vibration motor is a DC motor with permanent magnet and a shaft coupled with an eccentric mass. The dynamics, in the presence of an unknown disturbance torque d at the system, is described by the following differential equation:

$$\dot{\omega} = \frac{K_T}{JR}V_s - \frac{bR + K_T^2}{JR}\omega - \frac{c\,\text{sgn}(\omega)}{J} - \frac{mgr}{J}\sin(\theta) + \frac{d}{J} \quad (13)$$

where ω is the rotational speed of the vibration motor shaft, θ is the angular position of the eccentric mass, g is the gravitational acceleration, J is the eccentric load's moment of inertia, b is the viscous friction, the term $c\,\text{sgn}(\omega)$ is the Coulomb friction at the vibration motors axis, and r is the distance between the center of the eccentric mass m, and the shaft of the motor. The motors electrical resistance is denoted

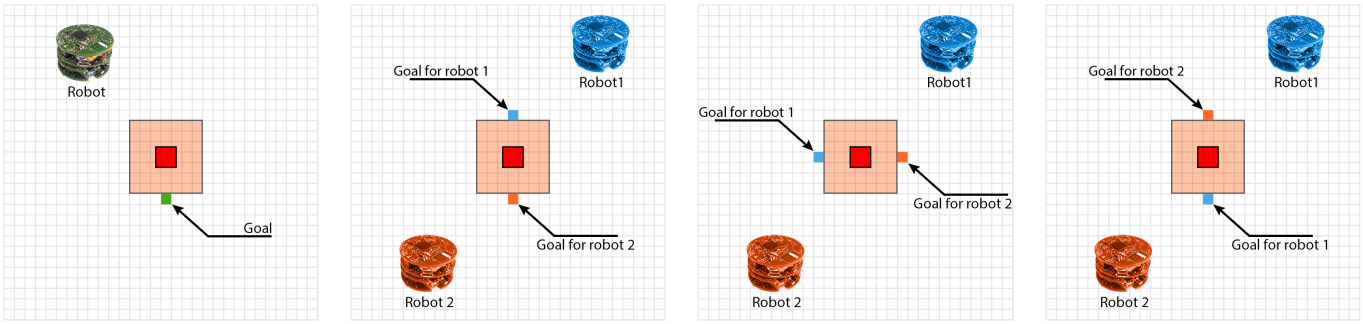


Fig. 5. Scenarios description used during our simulations with one (1) or two mini-robots (2-4)

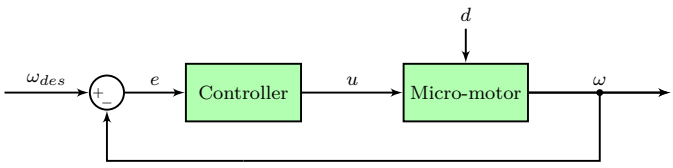


Fig. 6. Rotational speed closed-loop control for each vibration motor

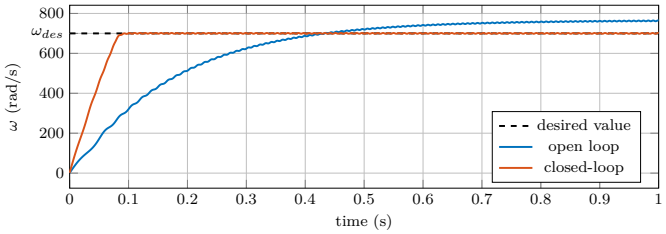


Fig. 7. Open loop and closed-loop simulation results of the vibration motor

by R, K_T is the torque constant of the motor, the voltage V_s is the input to the motor, and the windings inductance is neglected.

In order to compensate for the unknown disturbances, and improve the motion resolution and the bandwidth of the mini-robot, a simple and low-cost PI speed controller for each vibration motor is implemented. The block diagram of the resulted closed-loop system is shown in Fig. 6. The controller block is described by the following equation:

$$u = K_p e + K_i \int e \, \mathrm{d}t \tag{14}$$

where e is the error between the desired, ω_{des}, and the measured, ω, rotational speed of each vibration motor shaft, u is the control input, and the constants K_p and K_i are the controller gains. The control input, i.e. the voltage V_s to the motors, is provided by on-board batteries, and a restriction is imposed so that the control input, u, can not exceed $V_{s,max}$.

The time responses of the open loop and the closed-loop systems in test simulation runs are depicted in Fig. 7. Table II gives the values of the parameters used in the simulation runs. We see that in the case of the closed-loop system, the desired rotational speed of the vibration motor, ω_{des}, is achieved, despite the disturbance torque and the power restrictions. In addition, the response time of the vibration motor is significantly reduced compared to the open loop system.

TABLE II
PARAMETERS FOR THE SIMULATION OF THE CLOSED-LOOP VIBRATION MOTOR SYSTEM

Parameter description	Symbol	Value
Motor eccentric mass	m	0.00021 kg
Eccentricity of the rotating mass	r	0.00177 m
Torque constant	K_T	3.64×10^{-4} Nm/A
Eccentric load's moment of inertia	J	2.67×10^{-9} kg m^2
Motors electrical resistance	R	10.7
Motors viscous friction	b	2.94×10^{-9} Ns/m
Motors Coulomb friction	c	1.34×10^{-5} Nm
Gravitational acceleration	g	9.81 m/s^2
Desired rotational speed	ω_{des}	700 rad/s
Disturbance torque	d	10^{-6} Nm
Proportional term gain	K_p	0.015
Integral term gain	K_i	0.025
Maximum input voltage	$V_{s,max}$	3.0 V

V. Simulation Results

We have studied the performance of the proposed method by conducting several simulated experimental scenarios. The simulation environment has been implemented using the ROS (Robot Operating System) framework. It includes the kinematic and dynamic model of the mini-robotic platforms, information about the proximity between the robots and white noise added to each mini-robot rotational speed that is equivalent to $\pm 5\%$ of the nominal rotational speed. In all simulation runs the integration time step was set to $dt = 10^{-5}$. The RL agent step is equal to 2×10^5 integration time steps, equivalent to 2 seconds. The workspace is 30×30 cm, where a rectangular restricted zone of size 7×7 cm is located at its center. It is assumed that every target position of each mini-robot occupies a single cell in the border of this central zone.

During executing the Q-learning framework for training the agents we have considered a discount rate of $\gamma = 0.99$, a learning rate parameter of $\eta = 0.01$, as well as an ε-greedy

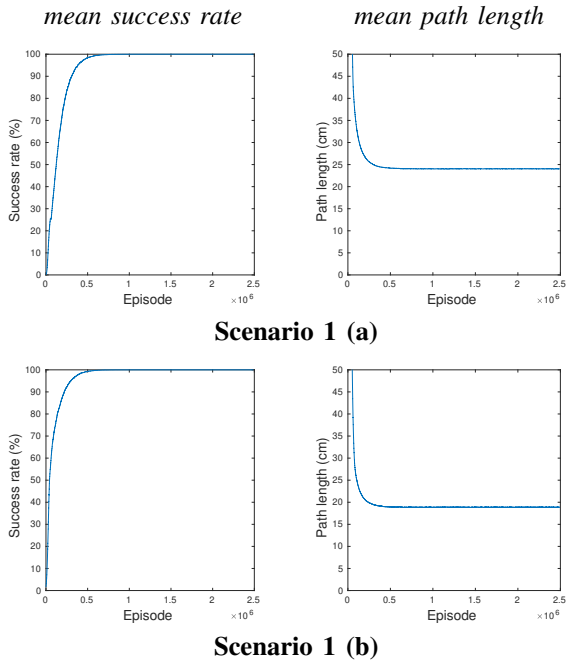


Fig. 8. Simulated results of single mini-robot using two alternations of the Scenario 1 in Fig. 5. The curves represent the mean success rate and the mean path length needed to reach the target.

exploration scheme with an initial probability of $\varepsilon = 0.9$ and a linear reduction scheme using a coefficient of 0.999 at every 500 episodes. In total 5×10^6 episodes (execution time about 10 min) were required for training the agents, where the initial 60% of them were used for the exploration phase. We validate all the policies generated by the proposed method based on a) the percentage of successful episodes and b) the required average distance to reach the mini-robots' targets when starting from a random position of the work space border. In all scenarios we calculated the mean value of each performance metric after executing 20 independent simulation runs.

A. Scenario description

The scenarios during the simulated experiments are depicted in Fig. 5.

- In the case of a single mini-robot (Fig. 5.1) we have considered two scenarios that differs in the starting position of the mini-robot: (a) randomly from the upper side of the board ($x^{init} \in [0, 30]$ cm and $y^{init} \in [29, 30]$ cm) and (b) randomly from a zone of width equal to one cell on every side of the board (see Fig. 9).
- In the case of two mini-robots we have applied three scenarios with different target position for each mini-robot, as shown in Fig. 5.2, 5.3 and 5.4, respectively. During all scenarios the start position of each mini-robot is chosen randomly within the upper side and the lower side of the workspace, respectively.

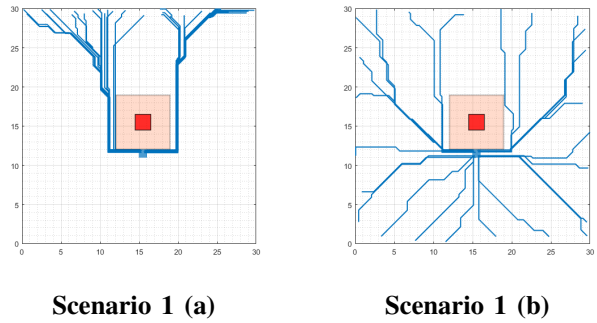


Fig. 9. Exemplar mini-robot trajectories obtained after finishing the learning procedure in the case of two alternations of the Scenario 1 in Fig. 5.

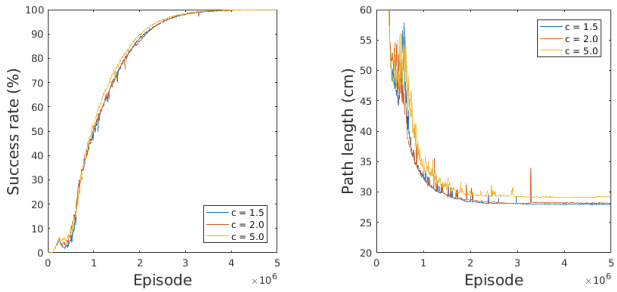


Fig. 10. Impact of the parameter c of the reward function Eqs. 11, 12. The three curves of mean success rate and path length correspond to three selected values of this parameter.

B. Experiments with single mini-robot

At first we examined the performance of the proposed methodology to the simulated environment in the case of a single mini-robot. In Fig. 8 the relevant results are shown, i.e. the mean success rate (percentage of reaching the target) and the mean required path length (in cm) of the last 100 episodes. As shown, in both scenarios the optimal policy is achieved quickly using the proposed methodology. In addition, several paths are illustrated in Fig. 9. Based on the results shown in Fig. 9 it is worth mentioning that the proposed framework has the ability to estimate sub-optimal paths from any random starting point of a relative large grid workspace.

C. Experiments with a pair of mini-robots

The performance of the proposed method was also evaluated in cases of having a pair of mini-robots. Initially, we made an experimental study about the impact of the c coefficient in (12) of the proposed reward function (11). As indicated previously, this coefficient is used to penalize the case of two mini-robots getting closer. Three different values of the c parameter, $c = \{1.5, 2.0, 5.0\}$ are tested. The results are shown in Fig. 10 in terms of two evaluation metrics: mean success rate and mean estimated path length. According to Fig. 10, better results are obtained when $c = 1.5$ and therefore, we have adopted this value in all simulated runs.

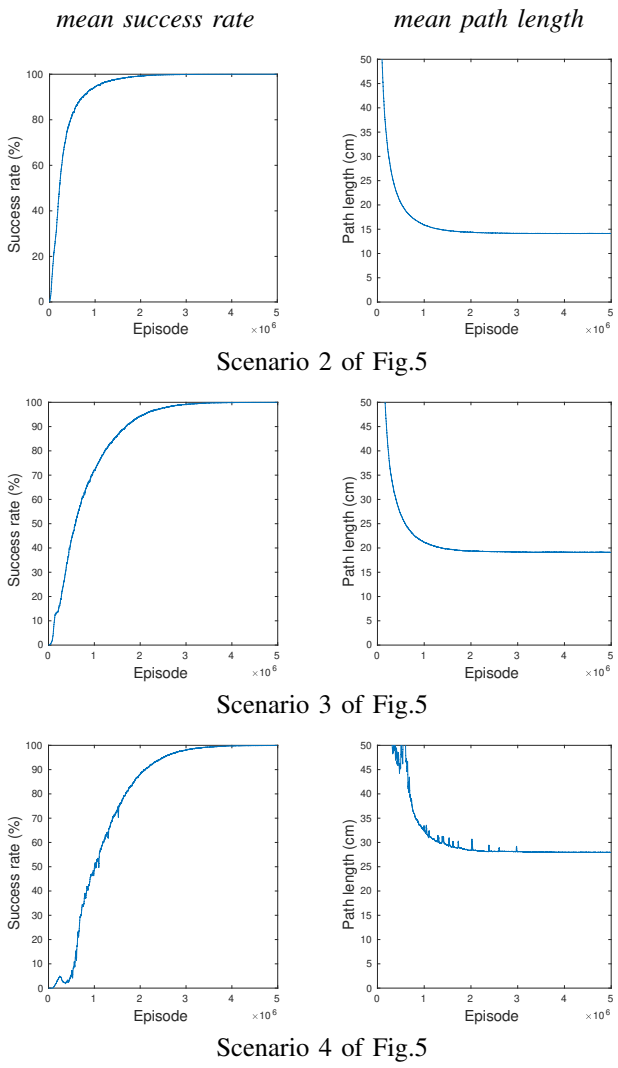


Fig. 11. Simulated results of the two mini-robots using three Scenarios 2, 3 and 4, presented in Fig. 5. The curves represent the mean success rate and the mean path length needed to reach both targets.

Next, we performed experiments using two mini-robots in three different simulated scenarios with different level of difficulty as presented in Fig. 5.2, 5.3 and 5.4, respectively. In the first scenario the target of each mini-robot is just opposite to its starting position and thus the probability of a collision is very low. The level of difficulty is progressively increased in the next two scenarios 5.3 and 5.4, and the environment becomes more complicated in terms of the position of both targets. In these cases, each mini-robot not only need to reach the target but also to avoid the collision with the other.

In Fig. 11 we illustrate the learning curves of the proposed RL methodology calculated by the mean values of 20 independent simulated runs. Two quantities are shown: the mean success rate and the mean path length of both mini-robots. It must be noted that once the first robot reaches its target, it remains there until the other mini-robot terminates its motion (either successfully, or not). An episode is considered as successful only when both mini-robots manage to discover their targets. Several successful paths for various scenarios are shown in Fig. 12. Seemingly, the proposed framework has the ability to estimate sub-optimal paths of both mini-robots from a random starting position in a relative large workspace, while a collision between them is avoided.

VI. Conclusions

In this study we have presented a complete reinforcement learning framework for the autonomous navigation of a pair of mini-robots where their actuation forces are provided by two vibration motors. The key aspect of the proposed scheme lies on the efficient design of input state space of mini-robots that allows the creation of a collaborative environment between two agents, as well as the application of a low-level simple and fast PI velocity controller for each vibration motor. Initial simulation results showed good performance. It is our intention to further pursue and develop the proposed method in three directions:

- Extend our method to a multi-agent reinforcement learning framework [8] by considering more than two mini-robots working together in a collaborative environment.
- Recast the presented framework as deep reinforcement learning scheme [5], [10].
- Validate the proposed framework in a more complex environment with greater degree of uncertainty, as well as in biomedical applications using real mini-robots that are currently under construction.

References

[1] K. Blekas and K. Vlachos. Rl-based path planning for an over-actuated floating vehicle under disturbances. *Robotics and Autonomous Systems*, 101:93–102, 2018.

[2] J. Brufau and M. Puig-Vidal et. al. MICRON: Small Autonomous Robot for Cell Manipulation Applications. In *Proc. of the IEEE International Conference on Robotics & Automation*, 2005.

[3] R. Buchi, W. Zesch, and A. Codourey. Inertial Drives for Micro- and Nanorobots: Analytical Study. In *SPIE Photonics East '95: Proc. Microrobotics and Micromachanical Systems Symposium*, volume 2593, 1995.

[4] M. Sylvain et al. Three-Legged Wireless Miniature Robots for Mass-scale Operations at the Sub-atomic Scale. In *IEEE International Conference on Robotics & Automation*, pages 3423–3428, 2001.

[5] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT Press, 2016.

[6] S. Herbrechtsmeier, T. Korthals, T. Schopping, and U. Rckert. Amiro: A modular customizable open-source mini robot platform. In *2016 20th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 687–692, 2016.

[7] M. Karpelson, G.Y. Wei, and R.J. Wood. Driving high voltage piezoelectric actuators in microrobotic applications. *Sensors and Actuators A: Physical*, 2011.

[8] J. R. Kok and N. Vlassis. Collaborative multiagent reinforcement learning by payoff propagation. *Journal of Machine Learning Research*, 17:1789 – 1828, 2006.

[9] T. Korthals, J. Leitner, and U. Rckert. Coordinated heterogeneous distributed perception based on latent space representation. *CoRR*, abs/1809.04558, 2018.

[10] V. Mnih, D. Kavukcuoglu, and D. Silver et. al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529 – 533, 2015.

[11] M. Rubenstein, C. Ahler, and R. Nagpal. Kilobot: A low cost scalable robot system for collective behaviors. In *2012 IEEE International Conference on Robotics and Automation*, pages 3293–3298, May 2012.

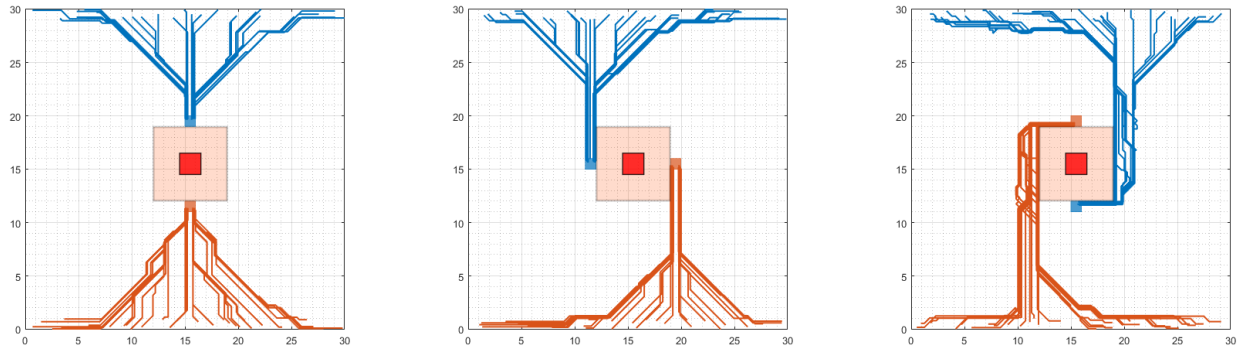


Fig. 12. Exemplar trajectories of two mini-robots obtained after finishing the learning procedure in the case three Scenarios 2, 3 and 4 in Fig. 5.

[12] F. Schmoeckel and S. Fatikow. Smart flexible microrobots for scanning electron microscope (sem) applications. *Journal of Intelligent Material Systems and Structures*, (3):191–198, 2000.

[13] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press Cambridge, USA, 1998.

[14] C. Szepesvari. *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2009.

[15] P. Vartholomeos, S. Loizou, M. Thiel, K. Kyriakopoulos, and E. Papadopoulos. Control of the multi agent micro-robotic platform micron. In *IEEE International Conference on Control Applications*, pages 1414–1419, 2006.

[16] P. Vartholomeos and E. Papadopoulos. Dynamics, design and simulation of a novel microrobotic platform employing vibration microactuators. *Journal of Dynamic Systems, Measurement and Control*, 28(1):122133, 2006.

[17] P. Vartholomeos, K. Vlachos, and E. Papadopoulos. Analysis and motion control of a centrifugal-force microrobotic platform. 10(3):545–553, 2013.

[18] K. Vlachos, D. Papadimitriou, and E. Papadopoulos. Vibration-driven microrobot positioning methodologies for nonholonomic constraint compensation. *Engineering*, 1(1):066 – 072, 2015.

[19] C. Watkins and P. Dayan. Q-learning. pages 279–292, 1992.