A Multiagent Reinforcement Learning Approach for Inverse Kinematics of High Dimensional Manipulators with Precision Positioning

Yasmin Ansari, Egidio Falotico, Yoan Mollard, Baptiste Busch, Matteo Cianchetti, Cecilia Laschi

A Multiagent Reinforcement Learning Approach for Inverse Kinematics of High Dimensional Manipulators with Precision Positioning

Yasmin Ansari, Egidio Falotico, *Member, IEEE*, Yoan Mollard, Baptiste Busch, Matteo Cianchetti, *Member, IEEE* Cecilia Laschi, Senior *Member, IEEE*

***Abstract*— Flexible manipulators based on soft robotic technologies demonstrate compliance and dexterous maneuverability with virtually infinite degrees-of-freedom. Such systems have great potential in assistive and surgical fields where safe human-robot interaction is a prime concern. However, in order to enable practical application in these environments, intelligent control frameworks are required that can automate low-level sensorimotor skills to reach targets with high precision. We designed a novel motor learning algorithm based on cooperative Multi-Agent Reinforcement Learning that enables high-dimensional manipulators to exploit an abstracted state-space through a reward-guided mechanism to find solutions that have a guaranteed precision. We test our algorithm on a simulated planar 6-DOF with a discrete action-set and show that the all the points reached by the manipulator average an accuracy of 0.0056m (± 0.002). The algorithm was found to be repeatable. We further validated our concept on the Baxter robotic arm to generate solutions up to 0.008m, exceptions being the joint angle accuracy and calibration of the robot.**

I. INTRODUCTION

Minimally invasive surgery (MIS) [1] is an advanced surgical procedure that uses a limited number of ports either natural or through a small incision to access internal organs. The use of manipulators based on traditional design (i.e. cables, pulleys, gears) for single-port intervention is known to be limited in its distal maneuverability due to few degrees-of-freedom (DOFs). A more innovative approach involves the application of Soft Robotics [2] to design whole arm flexible manipulators with virtually infinite DOF. These flexible manipulators take inspiration from boneless structures found in nature such as the octopus tentacles, elephant trunks, etc. that exhibit advanced manipulation capabilities due to their muscular arrangement. STIFF-FLOP [3] is an example of a soft surgical manipulator that is modular where each module comprises of radially arranged flexible fluidic actuators (FFA) [4] encapsulated within an elastomeric outer body. A combination of various simple feedforward actuation sequences produces elongation and omnidirectional bending. Combining three modules in series results in many DOFs with a highly dexterous workspace with no kinematic singularities. Additionally, they are safe to interact with due to the compliance of the soft actuators. However, the successful application of such soft continuum manipulators in these demanding environments is dependent upon its ability to automate low-level reaching skills with precision.

Marchese et. al. [5] applied a closed-loop controller on a 3D soft-arm to position the end effector to reach a ball with a diameter of 0.04m. Giorelli et al. [6] used a Jacobian based approach to reach an average tip accuracy of 6% the total manipulator length. These traditional methods are limited by modelling assumptions, computational expense, and most importantly, precision that needs to be further reduced for technological advancements in soft robotics. Learning mechanisms [7] provide a more promising approach by encoding correlations between sensorimotor data through internal models [8]. Malakzadeh et. al [9] applied imitation learning to a high-dimensional soft manipulator though this implies that the robot can only be as good as the provided information. Interaction with the environment through exploration is essential for a robot to learn optimal behavior [10] [11]. However, this is a non-trivial task for high-dimensional systems that can generate a large amount of redundant data [12]. Morphological Computation [13] has potential as a control paradigm to exploit high-dimensional structures as a computational resource through exploration, however, current applications [14] are limited to learn dynamical behavior without taking precision into account. Goal-directed motor exploration [15] is the most optimal framework, so far, to learn inverse kinematics with precision but it requires the controller to define linear paths in state-space. We propose to address this task through Multi-Agent Reinforcement Learning (MARL) [16] by viewing a

This work is supported by People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement number 608022

Y. Ansari, E. Falotico, M. Cianchetti, C. Laschi are with the Soft Robotics Lab, The BioRobotics Institute, Scuola Superiore Sant'Anna, Pisa, 56025, Italy.(email: y.ansari, e.falotico, m.cianchetti , c.laschi)@sssup.it

Y. Mollard, B. Busch, are with the FLOWERS Lab, INRIA (email: yoan.mollard, baptiste.busch)@inria.fr

manipulator as a group of independent agents that share an environment where they must coordinate their behavior through autonomous exploration to reach a target.

MARL has gained rapid success in multi-robot systems [17] due to its inherent nature to decentralize complex problems that accounts for a speed-up in learning. Despite these potential benefits, these applications are limited to a few DOFs [18] [19]. This is due to the 'curse-of-dimensionality' faced as the number of agents increase. An additional challenge is to find solutions in 3D continuous state-space with precision. The novelty in this work lies in the design of a reward-guiding mechanism that enables the manipulator to learn optimally increasing actions over time in abstracted state-space to reach a global optimum with precision. We test our algorithm in simulation on a 6-DOF planar robotic arm with a discrete action set. It is able to reach 129 points in its workspace with a mean accuracy of 0.0056m (±0.002). We then validate this concept on the Baxter arm in 3D Cartesian space. We show that the algorithm can reach the goals within 0.008m precision with exceptions from joint limits.

Section II discusses the development of the learning framework. Section III presents the model selection methodology followed by experimental analysis on the simulated arm and Baxter anthropomorphic platform in Section IV. Section V provides a conclusion of the overall work with future research goals.

II. Inverse Kinematics Learning Model

RL [20] is an adaptive optimization technique where a single agent uses trial-and-error to learn an optimal behavior. Model-free robot control policies can be learnt through an action-value function (also called Q-function) by maximizing the expected cumulative discounted reward after executing an action ($a_t = a$) in a current state ($s_t = s$) and following a given policy π (ϵ-greedy, etc.) thereafter.

$$Q^{\pi} = E\{\sum_{j=0}^{\tau} \gamma^j r_{t+j+1} \mid s_t = s, a_t = a, \pi\} \quad (1)$$

Where, γ is the discount rate and $\gamma \in [0, 1]$; t denotes the time-steps to when the episode terminates; r is the reward received at each time-step. This is scaled to multiple-agents by associating an independent Q-function [21] to each agent as shown in Equation 2. A control policy can be obtained only if all concurrent policies reach a global optimum.

$$Q_i^{\pi}(s, a) = E_i^{\pi}\{\sum_{j=0}^{\tau} \gamma^j r_{(j+t+1)} \mid s_t = s, a_t = a, \pi\} \quad (2)$$

where, $i = 1 \ldots n$, where n is total number of agents. However, the presence of multiple agents within the same environment renders it non-stationary and partially-observable from the point of view of a single agent. The model-free Sarsa(λ) Temporal Difference (TD) with eligibility traces control approach is applied that is an online on-policy Q-function estimator with memory, allowing real-time adaptive control. Mathematically,

$$Q_{t+1}(s, a) = Q_t(s, a) + \alpha * \delta_t * e_t(s, a) \quad (3)$$

$$\delta_t = r_{t+1} + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_{t+1}(s_t, a_t) \quad (4)$$

$$e_t(s, a) = \begin{cases} \gamma\lambda e_{t-1}(s, a) + 1 & if\ s = s_t,\ a = a_t \\ \gamma\lambda e_{t-1}(s, a) & otherwise \end{cases} \quad (5)$$

where, $Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)$ is the temporal-difference error; α is the step-size parameter (also known as learning rate in some cases); e_t is the eligibility trace from the backward view of the temporal difference learning where , λ is the accumulating trace-decay error.

A. Abstraction of 3D Cartesian State-Space

Positioning the robotic arm in 3D Cartesian state-space formulates a continuous-valued domain. We propose to solve the task of reaching by abstracting the state-space such that the euclidean millimeter distance can be locally generalized to the same actuator input. This can be achieved through function approximation where a smaller number of features are used to represent the infinite-sized region. Tile coding [20] is differentiable, stable, piecewise-constant feature-based approximator that allows to linearly approximate solutions for a non-linear system. This is achieved by partitioning the state space into multiple layers called tilings. Each element of a tiling is called a tile that allows for local generalization of state-space dependent upon the shape of the tile. These characteristics formulate the underlying motivation to employ it as our function approximator. Only one tile per tiling is activated if and only if the given state falls in the region delineated by that tile (Fig 1). The Q-function is then simply represented by a sum of the indexes of these activated tiles as,

$$Q(\mathrm{s}, \mathrm{a}) = \sum_{j=0}^{k} \Theta_j(s) w_j \quad (6)$$

where $j = 1 \ldots k$, where k is the total number of tilings; $\Theta_j(s)$ is the value (1 or 0) of the *jth* tile given state s; w_j is the weight of that tile.

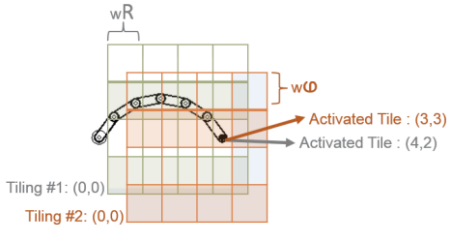

Fig 1. 2D Cartesian plane is abstracted into 2-layered rectangular tilings in R and φ dimensions. There are 4 tiles with a width of wR and wφ in each dimension, respectively. The origin of the tilings are offset w.r.t. each other. The position of the soft manipulator end effector activates one tile per tiling.

This equation is applied to all the actuators i.e. for $i = 1 \ldots n$, where n is total number of actuators. Equation 6 shows that the computational complexity of tile coding is linearly dependent on the number of tilings. The resolution of a tile is given as,

$$R = \frac{w}{T} \tag{7}$$

where, w represents the width of the tile; T represents the total number of tilings. This equation is calculated for all dimensions of the state-space i.e., for a 3D Cartesian state-space is in the x-y-z dimensions. The shape of tiles is usually problem specific as each manipulator will occupy a different reachable workspace. This information has to be provided to a programmer who will then then create appropriate shaped tiles to fit for the given workspace. However, the authors argue that the problem can be generalized by creating tiles in spherical co-ordinates (R, ω, θ). The underlying principle is to parameterize the tilings such that the range in each dimension is defined as: (i) $R = [0 \max(R)]$; (ii) $\omega = [0^{\circ}\ 360^{\circ}]$; and (iii) $\theta = [0^{\circ} \max(\theta)^{\circ}]$; where max(R) refers to the length of the manipulator in a fully extended state; ω refers to the azimuth which for omnidirectional bending will always be 360^0; and θ refers to the zenith dependent upon the contraction capability of the manipulator (Note: θ represents a zenith angle measured from the vertical axis.). This defines a volumetric space within which the manipulator is free to reach any point. Knowing these three quantities, rectangular shaped tiles can be created in each dimension, hence, limiting the tiling space to the reachable workspace that can be scaled to any manipulator. Controlling the precision is then directly proportional to controlling the width of the rectangular tile in each dimension. This will be discussed in more detail in Section III *A*.

B. Reward-Guided Actor-Critic Architecture

We combine the episodic Sarsa(**λ**) TD with the abstracted state-space in a model-free actor-critic architecture [20] to with a discrete action-set. A generalized policy iteration (GPI) is applied where an episode starts with the arm in a resting position proceeded by each agent following the ϵ-greedy policy (**actor**) to select either optimal actions (move the arm towards the goal) or sub-optimal actions (move the arm away from the goal). The update in the Q-function (**critic)** by an agent are based solely on a global environmental feedback without being affected by the behavior of any other agent. We design the scalar reward dependent upon the Euclidean distance from the goal such that it has a high negative value for distances further away from the goal that progressively but discretely decreases towards target, where it receives a reward of 0. The motivation behind this reward structure is to enable the robotic arm to make goal-directed attempts [22]. The episode will end either when the goal is reached or the maximum number of trials per episode are reached.

C. Dealing with High-Dimensionality

It has been heuristically proven that decreasing the exit probabilities out of tiles with non-optimal actions improves the performance of tile coding [23]. We take advantage of our reward structure to apply this concept in our algorithm. Every time it encounters an action that is rewarded with a scalar value higher than previously encountered ones, that action will be the first one taken by the system from the rest position from the next episode onwards. This process will continue until the reward before 0 is obtained. Additionally, every time a better action is selected from the rest position, the exploration rate is reduced by half until the goal is reached a certain number of times where it is made completely greedy. In the following section, we discuss the application of this algorithm in simulation and hardware.

III. MODEL SELECTION

The robotic platform under consideration is a planar 6-DOF simulated robotic arm [24]. All the joints under consideration are revolute. This gives us a mapping from an $\mathbb{R}^6$ motor space to an $\mathbb{R}^2$ cartesian space. The total length of the robotic arm is 1m with each link equal in length. The base is centered at the origin of the Cartesian plane. The discrete action set can decrease/increase the angle or keep it unchanged within a range of $[-180^0, 180^0]$. The robotic arm, its initial position, and target points of the robotic arm is illustrated in Fig 2.

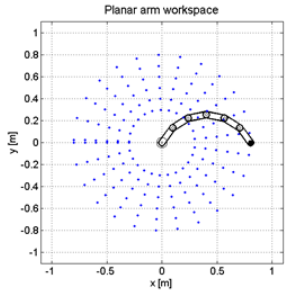

Fig 2. The workspace of the 6-DOF robotic arm

A. State-Action Parameter Selection

The resolution ($wR, w\omega, w\theta, T$), step-size (α) exploration (ϵ), and eligibility traces (λ) all need to be tuned for an optimum trade-off between precision and learning time. This is a non-trivial task that in most works is done empirically [25] [26]. We also follow an empirical approach, the generalized steps of which are: For a given ϵ and λ, we first define the reachable workspace of the manipulator as described in Section II A. Then, a minimum number of tiles

nR, $n\omega$, and $n\theta$ are heuristically initialized in the R, ω, and θ dimensions, respectively such that the resulting precision R/nR, $\omega/n\omega$, and $\theta/n\theta$ is in a centimeter range that can be improved to millimeter range through more tiling layers. Finally, a grid search is performed over nT tilings and $n\alpha$ step-sizes (forming an nT x $n\alpha$ matrix) to reach 20 random goals. The role of the discrete action set is to ensure as much reachability as possible in the defined workspace. In this work, this is found through trial-and-error.

For planar movement, we consider only x-y cartesian and corresponding R-φ spherical coordinates. The reachable workspace of the manipulator is: $R = [0\ 1\text{m}]$ and $\varphi = [0^0\ 360^0]$. We heuristically selected a discrete action set of 7 actions (see Table 1). Thus, there exists a total of 7e6 = 823543 possible input combinations. For ϵ=0.1 and λ=0.9, standard values found in text, we heuristically initialize 40 and 24 tiles such that the planar tiling layer of 960 tiles has a reaching precision of 0.025m and 0.261m in the two dimensions, respectively. We then perform a grid search for tilings between 2 to 32 in powers of 2 as recommended by [27] and 5 step-sizes of heuristically selected values, forming a 5x5 matrix. The target radius is set to 0.008m.

Table 1 A summary of experimental results on 6 DOF planar arm

REACHABLE WORKSPACE	R	0-1m	nR	40	wR	0.05
	ω	0°-360°	nω	24	wω	0.26
STATE-SPACE PARAMETER SELECTION	ϵ	0.1				
	ƛ	0.9				
	ɣ	1				
	T	Range Tested		2 – 32 (Powers of 2)		
		Optimal Value		4		
	α	Values Tested		[0.25 0.16 0.08 0.05 0.025]		
		Optimal Value		0.16		
ACTION SET	[-0.087 -.0349 -0.0175 0 .0175 0.0349 0.087] (m)					
MEAN REACHING ERROR FOR 20 GOALS				0.0055m ± 0.00186m		
MEAN EPISODES REQUIRED FOR CONVERGENCE						56

The algorithm runs for a total of 300 episodes with 100 trials per episode which means that in one experiment, the algorithm is given a total of 3e4 trials to converge to an optimal solution for two performance criteria: (i) mean reaching error (ii) mean episodes needed for convergence. It is worth noticing that this number is 27 times less than the number of input combinations that the manipulator can select from. The optimal settings are provided in Table 1. α was found to be 0.16 and T was 4. The value of the step-size parameter implies that the arm moves roughly six-tenth of the way across a tile to the target in one update. A smaller number of steps per tile reduces the time spent on a tile in turn decreasing the risk of error propagation if the arm takes a sub-optimal action in it. With these parameter settings, the algorithm was able to reach all 20 goals with a mean reaching precision of 0.0055m ($\pm$0.00186) with an average of 56 episodes required for convergence. The state space now comprises of a total 3840 tiles (4 layers * 960 tiles/layer) where each layer is offset with respect to one another.

IV. EXPERIMENTS

We experimented on 142 points in the reachable workspace. We found the algorithm to converge for 129 points with an average reaching accuracy of 0.0059m ($\pm$0.002). From these results, it is deduced that as long as we ensure millimeter accuracy in one dimension, we can meet the overall precision requirements through this algorithm. This is useful to avoid too much tiling in the ω dimension which is much larger as compared to the first. The solution for all the 142 points is found to be limited due to the discrete action set, and will be taken up as a future work using continuous action spaces.

Fig 3 depicts an example of the robotic arm reaching a target, its policy development, and the total accumulated reward. This policy (Fig 3 center) illustrates the goal-directed behavior mentioned in Section II B. Depending upon the distance from the goal, the manipulator initially takes actions with a lower scaler reward. As it repeatedly encounters actions with a higher scalar reward, it tends to exploit them more. Hence, over time it learns the ability to choose optimally increasing actions. As the probability of selecting better actions increases, the trials required to reach the goal decreases, ensuring convergence. This is particularly beneficial as it guides the robotic arm to perform exploration towards regions of interest without prior knowledge of the environment, but only of the system's perception of the goal in relation to its current position. This example converges at 63 episodes after testing a total of 5742 actions (Fig 3 Bottom). It is worthy that this number of actions is much less in comparison to the available 3e4 in one complete episode. The trend for the reward accumulation (Fig 3 Bottom) increases until convergence, which is as expected. We refer the readers to the complementary video for further reference.

B. Repeatability and Robustness

We repeated the tests on the workspace 5 times found it to reach the same targets with an average of 0.0061m ($\pm$0.0019).

C. Effects of taking Optimal Actions in Resting Position

In order to demonstrate why the method proposed in Section II D is essential to deal with high-dimensions, we tested 20 random goal points with and without exploiting the reward structure in the resting position. Table 2 illustrates a summary of the obtained results. For the former approach, all

goals are reached, however, for the latter only 9 are. After, reiterating the experiment 5 times, it was found that the repeatability was guaranteed for the former approach whereas not for the latter. Fig. 4 plots the accumulated absolute value of the reward in trying to reach each goal irrespective of whether it was successful or not. Fig 4. (Top) illustrates the rewards accumulated by the former approach and the axes has been scaled to the size of Fig. 4. (Bottom), representing the latter approach, in order to draw a relative comparison amongst both. It highlights that the maximum reward accumulated by the former approach (-5500000) is approximately equal to the minimum reward accumulated by the latter approach (-56000000). High negative reward implies that the latter method explores the state-space by taking more suboptimal actions. As a result, more error is propagated throughout the state-space which has a direct impact on the estimated Q-functions from which control policies are learned. This is why policies learned by the latter method Fig 5 (Top) require a much larger learning time in comparison to that in Fig 5 (Bottom) as well as the uncertainty of convergence in the former approach. However, it is worth mentioning that in the case the manipulator cannot deal with deal high dimensions, whenever it does find solutions, they are still high-precision. This fact is reiterated in Fig 5 which compares the policy development for the two approaches.

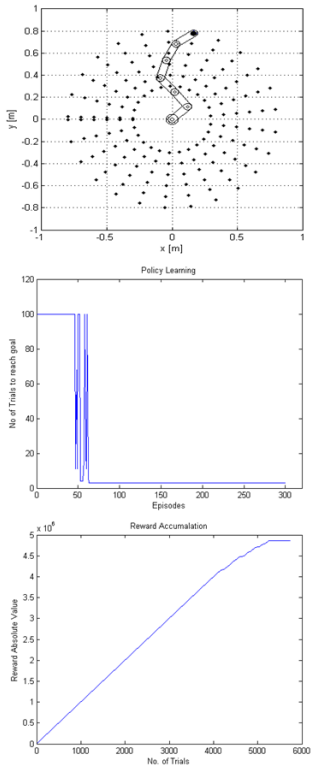

Fig 3. (Top) An example of the 6-DoF robotic planar converging to an optimal solution within 42.5 secs with a reaching accuracy of 3.2mm (Center) Policy Development (Bottom) Absolute Value of total reward accumulated

Fig 5 (Top) superimposes the policy development plots for the 20 learnt goals in the former method, whereas, Fig 5 (Bottom) superimposes the 9 learnt goals in the latter approach. All policy development plots observe the goal-directed behavior mentioned previously. The learning behavior in the former approach is concentrated within the first 120 episodes where the learning time increases with the increase in distance from the resting position. This is not true for the latter method where the learning time exhibits no stable identifiable trend.

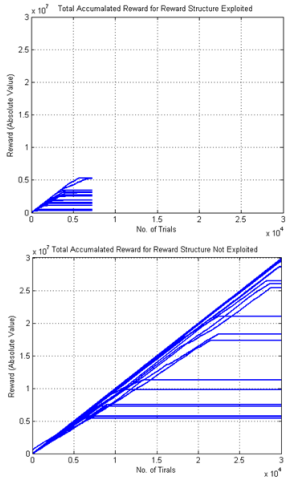

Fig 4. (Top) Rewards accumulated when taking progressively optimal actions from resting position for 20 targets. Steady-state straight lines show convergence. (Bottom) Rewards accumulated without taking progressively optimal actions from resting position for 20 target points. Linear lines indicate i.e. no learning.

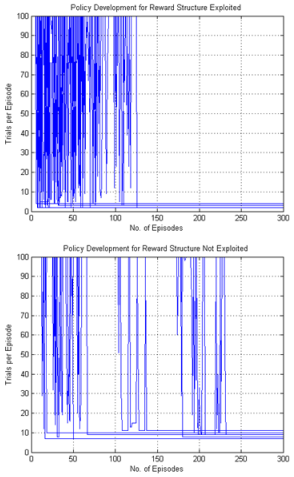

Fig 5. Results of reaching 20 goals with and without exploiting the reward structure (Top) 20 super-imposed learned policies when taking progressively optimal actions from resting position (Bottom) 9 superimposed policies when taking random actions from the resting position goals. These goals exhibit a learning time much slower in comparison to the first approach.

Table 2 Taking optimal Actions in resting position vs. not taking optimal actions in resting position. Test results on 20 random goals

EXPLOITING THE REWARD	YES	NO
NO. OF GOALS REACHED	20	9
REPEATABILITY (5 TRIALS)	Yes	No
REACHING ACCURACY	0.00632m ± 0.0017m	0.00598m ± 0.0021m

V. HARDWARE

We further validated the concept presented in Section III A on the left arm of the Baxter anthropomorphic robot developed by Rethink Robotics [28]. It has 7 DOFs: two rotational joints in the shoulder, two in the elbow, and three in the wrist. This gives us a mapping from an R^6 motor space to an R^3 cartesian space. The reachable workspace of the manipulator is identified as R = [0m 1m], $\omega = [0^0\ 360^0]$, and $\theta = [0^0\ 46^0]$. For the model selection and experiments, the forward kinematics (provided by the manufacturer) of the robot in simulation was used in conjunction with the algorithm and a procedure similar to Section III A was followed. The optimal parameters obtained are provided in Table 3.

A total of 150 target points in 3D Cartesian state-space were created and tested. For the given discrete action-set, the algorithm was able to generate a total of 15 solutions within a state-space of approximately (0.7m x 1m x 0.80m) region. Each solution from the set was then fed sequentially to the robot for a direct comparison of the prediction to the actual outcome. This was done by obtaining the 3D cartesian coordinates from an OptiTrack Motion Capture [29] vision feedback. Within the simulator, the solutions generated had a mean accuracy of 0.006m (±0.002). Practically, it was observed that 60% of the given solutions reached the goal with a mean reaching accuracy of 0.008m (± 0.001). The rest were off the target position with a mean reaching accuracy of 0.0138m (± 0.012). This offset could be credited to calibration issues and joint-accuracies limitations. Two such examples have been provided in Fig 6 and 7 below.

VI. CONCLUSION

Soft Robotics applied as an underlying key technology in the design of assistive and surgical tools can provide high dexterity in the instrument. However, their practical applicability is dependent upon the development of a new framework of intelligent control strategies that can automate accurate and repeatable low-level sensorimotor skills while taking into account the non-linearity and high dimensionality of these systems. We have designed an algorithm using the

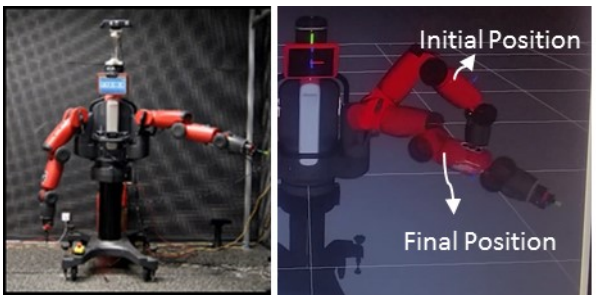

Fig 6. Target: [R = 0.76m $\omega = 63^o$ $\omega = 7^o$]; **(Left)** Real Reaching Accuracy: 0.019m **(Right)** Simulated Reaching Accuracy: 0.0052m generated in 52s

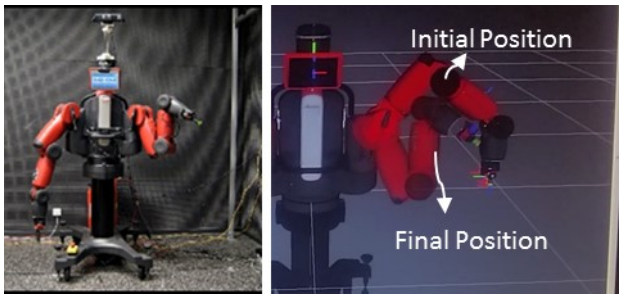

Fig 7. Target: [R = 0.425m $\omega = 41.5^o$ $\omega = 17.5^o$]; **(Left):** Real reaching Accuracy: 0.007m; (**Right):** Simulated reaching Accuracy: 0.003m generated in 27s.

Table 3 A summary of experimental Results on Baxter

REACHABLE WORKSPACE	R	0-1m	nR	30	wR	0.03
	ω	0°-360°	nω	6	wω	1.04
	θ	0°-46°	nθ	20	wθ	0.04
STATE-SPACE PARAMETER SELECTION	T	Range Tested		2 – 32 (Powers of 2)		
		Optimal Value		4		
	α	Values Tested		0.25; 0.16; 0.08; 0.05; 0.025		
		Optimal Value		0.16		
	ϵ	0.1				
	λ	0.9				
	γ	1				
ACTION SET	[-0.175 -0.0349 -0.0175 -0.00873 0 0.00873 0.0175 0.0349 0.175]m					
REACHING ERROR IN SIMULATION	0.006m (±0.0022)					
TOTAL TARGETS REACHED WITH PRECISION	9/15					

MARL framework that allows open-ended autonomous exploration limited to a manipulator's reachable workspace guided by the motivation to move towards the goal as quickly as possible. We tested this algorithm on 142 points for a planar 6-DOF arm with a discrete action action-set. With a

discrete action set of 7 actions, it was able to reach 129 points with a 0.0056m mean reaching accuracy. The algorithm also has been tested to be repeatable with robustness. We further validated the results on the Baxter Robotic arm. We found it capable of reaching points with within 0.008m accuracy. Future works will take into account continuous action spaces and follow trajectories. The algorithm will also be applied to soft robotic platforms.

ACKNOWLEDGMENT

This work is supported by People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement number 608022.

REFERENCES

[1] Ponsky TA, Khosla A, Ponsky JL. Minimally Invasive Surgery. Textbook of Clinical Gastroenterology and Hepatology, 2nd ed. Oxford: Wiley-Blackwell, 2012.

[2] Kim, Sangbae, Cecilia Laschi, and Barry Trimmer. "Soft robotics: a bioinspired evolution in robotics." Trends in biotechnology 31.5 (2013): 287-294.

[3] T. Ranzani, M. Cianchetti, G. Gerboni, I. D. Falco and A. Menciassi, "A Soft Modular Manipulator for Minimally Invasive Surgery: Design and Characterization of a Single Module," in IEEE Transactions on Robotics, vol. 32, no. 1, pp. 187-200, Feb. 2016.

[4] De Greef, Aline, Pierre Lambert, and Alain Delchambre. "Towards flexible medical instruments: Review of flexible fluidic actuators." Precision engineering 33.4 (2009): 311-321.

[5] Marchese, Andrew D., and Daniela Rus. "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator." The International Journal of Robotics Research (2015): 0278364915587925.

[6] Giorelli, Michele, et al. "A two dimensional inverse kinetics model of a cable driven manipulator inspired by the octopus arm." Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012.

[7] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," Cognitive Processing, vol. 12, no. 4 "Special Corner: Cognitive Robotics", 2011.

[8] D. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," Trends in Cog. Sci., vol. 2, no. 9, 1998

[9] Malekzadeh, Milad S., et al. "Learning by imitation with the STIFF-FLOP surgical robot: a biomimetic approach inspired by octopus movements." Robotics and Biomimetics 1.1 (2014): 1-15.

[10] Vannucci, L., Cauli, N., Falotico, E., Bernardino, A., Laschi, C.: Adaptive visual pursuit involving eye-head coordination and prediction of the target motion. In: IEEE-RAS International Conference on Humanoid Robots. (2014) 541–546

[11] Vannucci, L., Falotico, E., Di Lecce, N., Dario, P., Laschi, C.: Integrating feedback and predictive control in a bio-inspired model of visual pursuit implemented on a humanoid robot. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9222 (2015) 256–267

[12] Magill, Richard A., and David Anderson. Motor learning and control: Concepts and applications. Vol. 11. New York: McGraw-Hill, 2007.

[13] Zambrano D, Cianchetti M, Laschi C (2014) "The Morphological Computation Principles as a New Paradigm for Robotic Design" in Opinions and Outlooks on Morphological Computation, H. Hauser, R. M. Füchslin, R. Pfeifer (Ed.s), pp. 214-225.

[14] Hauser, Helmut, et al. "Towards a theoretical foundation for morphological computation with compliant bodies." Biological cybernetics 105.5-6 (2011): 355-370.

[15] Rolf, Matthias, and Jochen Jakob Steil. "Efficient exploratory learning of inverse kinematics on a bionic elephant trunk." Neural Networks and Learning Systems, IEEE Transactions on 25.6 (2014): 1147-1160.

[16] Yang, Erfu, and Dongbing Gu. Multiagent reinforcement learning for multi-robot systems: A survey. tech. rep, 2004.

[17] Busoniu, L.; Babuska, R.; De Schutter, B., "Multi-Agent Reinforcement Learning: A Survey," in Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on , vol., no., pp.1-6, 5-8 Dec. 2006

[18] Martin, J. A., & De Lope, H. (2007). A distributed reinforcement learning architecture for multi-link robots. In 4th International Conference on Informatics in Control, Automation and Robotics (pp. 192-197).

[19] Schuitema, Erik. Reinforcement learning on autonomous humanoid robots. TU Delft, Delft University of Technology, 2012.Watkins, C.J.C.H. (1989). Learning from delayed rewards. PhD Thesis, University of Cambridge, England

[20] Sutton, R. S., Barto, A. G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA. 1998

[21] Claus C, Boutilier C. The dynamics of reinforcement learning in cooperative multiagent systems. In: AAAI/IAAI 1998 Jul 26 (pp. 746-752).

[22] Rolf, Matthias. "Goal babbling with unknown ranges: A direction-sampling approach." Development and Learning and Epigenetic Robotics (ICDL), 2013 IEEE Third Joint International Conference on. IEEE, 2013.

[23] Scopes, Peter, and Daniel Kudenko. "Theoretical Properties and Heuristics for Tile Coding." ALA Workshop, AAMAS. Vol. 2014. 2014.

[24] Rolf, M., and J. J. Steil, "Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant", IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS), Vilamoura, Portugal, IEEE, pp. 3440-3446, 10/2012

[25] Whiteson, Shimon, Matthew E. Taylor, and Peter Stone. Adaptive tile coding for value function approximation. Computer Science Department, University of Texas at Austin, 2007.

[26] Sherstov, Alexander A., and Peter Stone. "Function approximation via tile coding: Automating parameter choice." Abstraction, Reformulation and Approximation. Springer Berlin Heidelberg, 2005. 194-205.

[27] Lin, Chun-Shin, and Hyongsuk Kim. "CMAC-based adaptive critic self-learning control." Neural Networks, IEEE Transactions on 2.5 (1991): 530-533.

[28] http://www.rethinkrobotics.com/baxter/

[29] http://www.optitrack.com/