2018 Annual American Control Conference (ACC)
June 27–29, 2018. Wisconsin Center, Milwaukee, USA

Distributed Planning of Multi-Segment Soft Robotic Arms

Wenlong Zhang* and Panagiotis Polygerinos

Abstract— **This paper presents a distributed kinematic planning algorithm for the end-effector of a multi-segment soft robotic arm to reach the goal position in both 2D and 3D spaces. The planning algorithm runs sequentially from the proximal to the distal segments. For each segment, the planning algorithm only requires the information about the position of itself, the end-effector, and the goal. The 2D planning of each segment is parameterized by a bending angle and an inflation ratio, which are determined by checking the overall geometry of the residual arm and the goal position. The same concept is extended to 3D planning, where parameters include inflation ratio, azimuth angle, and elevation angle for each segment. It is demonstrated in this paper that physical limits of the arm and challenging goal position could lead to failure in goal reaching. To account for this, an iterative learning function is proposed, which allows each segment to learn from past trials and be proactive for goal reaching. The proposed distributed planning algorithm, with iterative learning, demonstrates strong scalability, low computation cost, and robust goal reaching performance. Moreover, it does not need training data to pre-learn the configuration of the arm, which makes the algorithm highly applicable to a wide range of multi-segment soft and continuum robots. Both 2D and 3D simulation results are provided to illustrate the efficacy of the proposed algorithm.**

I. INTRODUCTION

Over the past decade, soft robotics has become a popular research area as it can offer a number of advantages over rigid robotic systems, including low cost, versatile motion, and safe interaction with human users. The recent introduction of soft robotics began with simple soft actuators capable of achieving complex motions from single inputs, and then with their combinations soft continuum systems are created, which are compliant and configurable. In that regard, soft robotic actuators can be categorized as cable-driven [1], electroactive polymers [2], shape memory alloys [3], pneumatic artificial muscles (PAMs, also called McKibben actuators) [4], inflatable structures [5], [6], and fluidic elastomeric actuators (FEAs) [7], with pneumatically actuated soft systems leading the exploration.

Currently, the main challenge of soft robotics, which is also the reason limiting its adaptation rates, is that the soft structures are difficult to model and, therefore, to plan and control for real-life applications [8]–[10]. In particular the challenge in precisely modeling and controlling a soft robot originates in its infinite degrees of freedom and complex nonlinear material properties. Attempts have been made

This work was supported in part by National Science Foundation under Grant CMMI-1800940.

The authors are with the Polytechnic School, Ira A. Fulton Schools of Engineering, Arizona State University, Mesa, AZ, 85212, U.S. Email: wenlong.zhang@asu.edu, polygerinos@asu.edu.

*Address all correspondence to this author.

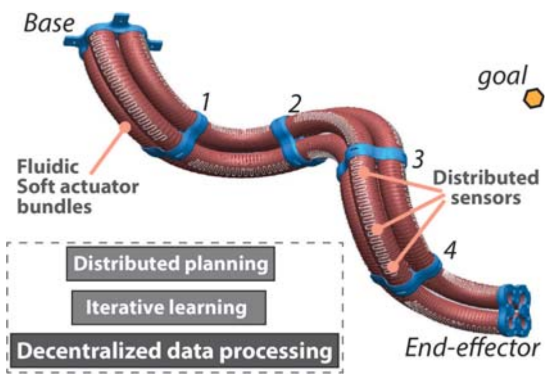


Fig. 1: Illustration of a soft robotic arm with modular fluidic soft actuator bundles and embedded distributed sensors.

through analytical and computational models to describe the behavior of such soft systems. Analytical modeling of soft actuators can be further categorized based on the assumptions of piecewise constant curvature [11] and non-constant curvature [6]. However, those analytical models all make simplifications that inevitably exclude information about the complex geometric shapes and volume deformations. As an alternative approach, finite element modeling could provide more accurately described kinematics and dynamics, as well as provide information about the local stresses and strains of soft actuators [12]. However, drawbacks of computational modeling include: 1) it cannot provide explicit differential equations to apply model-based control approaches, and 2) it requires long computation time to predict the output motion and force and thus cannot be used for real-time feedback control in real-world applications.

Based on different types of models, control techniques of soft robotics can also be categorized as model-based and model-free control. For model-based control, sliding mode control [13], optimal control [14], and model predictive control [6] have been applied to control various soft robots. For model-free control, bang-bang control, finite-state machine, and PID control have been popular approaches due to their easy implementation [15], [16]. Despite the great advances in the past decade, there are still many open problems in soft actuator modeling and control, including improving tracking accuracy and reducing algorithm complexity [8].

Besides modeling and low-level dynamic control of soft robots, motion planning also poses a unique challenge because soft robots do not have rigid links and segments. The most intuitive approach to tackle this challenge is to approximate a soft robot as a rigid-link robot with high number of segments, i.e., a continuum robot, and many continuum robot planning approaches can be then applied to soft robots. Sample-based planning approaches are very popular for continuum robots [17], and machine learning approaches have been extensively studied for continuum robot planning in recent years [18]. Bio-inspired planning is another popular approach for soft robots as they are usually designed to mimic different living mechanisms, such as octopus arms [19] and elephant trunks [20]. However, the limitations of the existing algorithms include: 1) it takes long time to learn the robot configurations in order to complete the tasks, 2) most planning algorithms do not consider the physical limits of soft arms, and 3) many existing algorithms are centralized and thus not scalable to large number of segments in soft/continuum robots.

This paper aims at addressing the challenges in soft robot planning and control, and a distributed planning algorithm is proposed with iterative learning. The approach is illustrated with a multi-segment soft robot arm, and it will allow the soft arm to generate their own motion rules in order to reach a desired goal. The contributions of this paper include: 1) a distributed soft arm planning algorithm is proposed to reduce the computation load and make the algorithm highly scalable. 2) An iterative learning function is introduced to allow the arm to learn from past trials and overcome its physical limits by optimizing its configuration. The algorithm is applicable in both 2D and 3D planning, and simulation results verify that it can allow the soft arm to reach the goal position after a few trials without large amount of training data.

The remainder of this paper is organized as follows. Section II provides a detailed problem statement. In Section III, the distributed planning algorithm is explained for goal reaching and the iterative learning algorithm is introduced to improve the articulation performance in 2D space. The algorithm is extended to 3D space in Section IV. Section V presents simulation results to demonstrate the ability of the soft robotic arm to reach goal locations in 2D and 3D spaces. Section VI concludes this paper and discusses future work.

II. PROBLEM STATEMENT

In Fig. 1, the chosen fluidic driven soft robotic arm design for this study is illustrated. This arm is comprised of soft linear actuators that are grouped in a parallel formation. This allows the individual actuators to work together and achieve bending motions. In 2D case, each soft actuator is capable of achieving a maximum bending angle (α_m) and a maximum inflation ratio (L_m). In 3D case, each soft actuator is capable of achieving a maximum azimuth angle (ϕ_m), a maximum elevation angle (γ_m), and a maximum inflation ratio (L_m). Furthermore, serial stacking of an N number of these bundled actuators creates a soft robotic arm capable of performing highly complex articulation motion in space. As stated in the introduction, the difficulty that the soft robotics community has faced in the past with similar types of soft arms is how to achieve effective control to reach a desired goal location.

In this paper, we set each segment of the arm to only know its own location, the location of the goal that the end-effector needs to be reached, and the current end-effector location. We aim at designing a decentralized planning algorithm for each segment of the soft arm so that the end-effector of the arm can reach the goal position. It should be noted that no prior information about the physical limits will be given and used in the algorithm, and no training data are collected about the arm. It is expected that the distributed planning algorithm will be highly scalable and can be applied in a wide range of multi-segment soft robot planning applications. However, the physical constraints as well as challenging locations of the goal can hinder the ability of the algorithm to reach the desired goal. Accounting for this, we further introduce an iterative learning function that allows the arm to investigate alternative actuation rules for each segment by learning from previous unsuccessful attempts.

In this paper, the following assumptions are made:

- Goal location is accurately known and not changing.
- Only kinematic planning is considered, without dynamic modeling or control of the arm.
- The algorithm cares only the base and tip locations of each segment, and it does not consider arm curvatures.
- The algorithm deals with position planning, and it does not consider orientation of the end-effector surface.
- Without loss of generality, the original length of each segment is assumed to be 1.
- Each segment can be bent, extended, and compressed.
- The soft arm is assumed to be with zero gravity.

III. DISTRIBUTED PLANNING IN 2D SPACE

In this paper, a distributed planning algorithm is proposed to plan each segment of the soft arm for goal reaching. Each segment will only need to sense position of itself and the last segment of the arm. The algorithm is first illustrated using an arm with four series-connected soft actuators in 2D space.

A. Distributed Planning for Goal Reaching

As mentioned in the problem statement, the goal of kinematic planning for the soft arm is to allow the tip to reach an arbitrary goal position in 2D space. The proposed algorithm will start from the base segment, and the next segments will be planned in series till the last segment. For each segment, the planning algorithm is to design its bending angle and inflation ratio. The algorithm is illustrated in Fig. 2. The node 0 indicates the base of the soft arm, and the node i indicates the end of the i^{th} segment. Fig. 2(a) shows the initial location of the soft arm without any actuation. Fig. 2(b) shows the planning of the first segment. The upper figure shows the bending angle planning. First, a line is drawn between node 0 and the goal, and the angle θ between this line and the current first segment is calculated. Given θ is the total bending angle that the arm should have in order to reach the goal, the bending angle of the first segment is determined as $\alpha_1 = \theta/4$ assuming the other three segments have the bending angle. The lower figure shows the inflation ratio planning. Since the goal is to let node 4 reach the goal, the inflation ratio is calculated by comparing the distance from the base to the goal (d_{g1}) and the distance from the base to the tip of the arm (d_{t1}), i.e., $L_1 = d_{g1}/d_{t1}$. Till now both the bending angle and inflation ratio have been determined, and the first segment of the arm will be actuated, as shown in Fig. 2(c). Fig. 2(d) demonstrates the planning for the second segment, which shares the same concept as the first segment except that now the new base is node 1. This process continues till the 4^{th} segment. Its bending angle is just the remaining angle is needs to bend to to point to the goal, and the inflation is determined such that the distance form node 3 to node 4 is the same as that from node 3 to the goal. This concludes the actuation of all segments, and the final arm kinematic configuration is shown in Fig. 2(f).

Let us revisit the algorithm, and it is clear that for the planning of each segment, only information about the locations of the current base and goal position is needed, which can be measured using either a motion capture system or embedded soft sensors. The location of the soft arm end-effector can be either broadcast or transmitted to the segment that needs it for its planning, which makes the algorithm distributed as it does not need information of any other segments in the soft arm. The computation load for the planning is very low with simple algebraic calculations, which makes the algorithm scalable for many real-world soft robotics applications. Algorithm 1 summarizes the distributed planning idea for a general n-segment soft robot arm.

B. Iterative Learning for Improved Performance

The distributed planning algorithm will work well if the goal position is not in a challenging position which requires very large extension/compression or bending angles. However, for safety concerns and protection of the soft actuators, physical limits of bending angles and inflation ratios will be applied in actual soft robot control systems. It should be noted the inflation ratio and bending angle for each segment are planned assuming the remaining segments will be the same, but it will not be the same as the segment is actuated and new observation of the actuation tip location is observed. This could result in extreme bending angles and inflation ratios for the last several segments, which results in failure of reaching the goal. To solve this problem, an iterative learning algorithm is proposed in this paper.

Algorithm 1 Distributed planning of a n-segment soft robotic arm for goal reaching in 2D space

Input: $(x_g, y_g), (x_0, y_0), (x_n, y_n)$

Output: $L_{\{1:n\}}, \alpha_{\{1:n\}}$

```
i = 1
while i ≤ n do
    // Determine the bending angle α_i.
    overall bending angle θ_i ← ∠(v_{i−1,n}, v_{i−1,g})
    α_i = θ_i/(n − i + 1)
    // Determine the inflation ratio L_i.
    L_i = ||v_{i−1,g}|| / ||v_{i−1,n}||
    Actuate the i^th segment based on α_i and L_i.
    Measure the tip location of the i^th segment (x_i, y_i) and
    the last segment of the arm (x_n, y_n)
    i ← i + 1
end while
```

The key idea of the iterative learning algorithm is to allow the soft arm to learn from the errors of the last trial, which is similar to iterative learning control. As the number of iteration increases, the arm will be proactive and actuate all segments towards the goal, rather than relying heavily on the last several segments. The iterative learning algorithm is summarized using a four-segment arm in Fig. 3. Besides the 2D space for the arm, the iteration domain is added so that the arm can learn from past experience. For each iteration, the following information is recorded for each segment: 1) calculated bending angle α_i, 2) actual bending angle α_i^*, 3) calculated inflation ratio L_i, and 4) actual inflation ratio L_i^*.

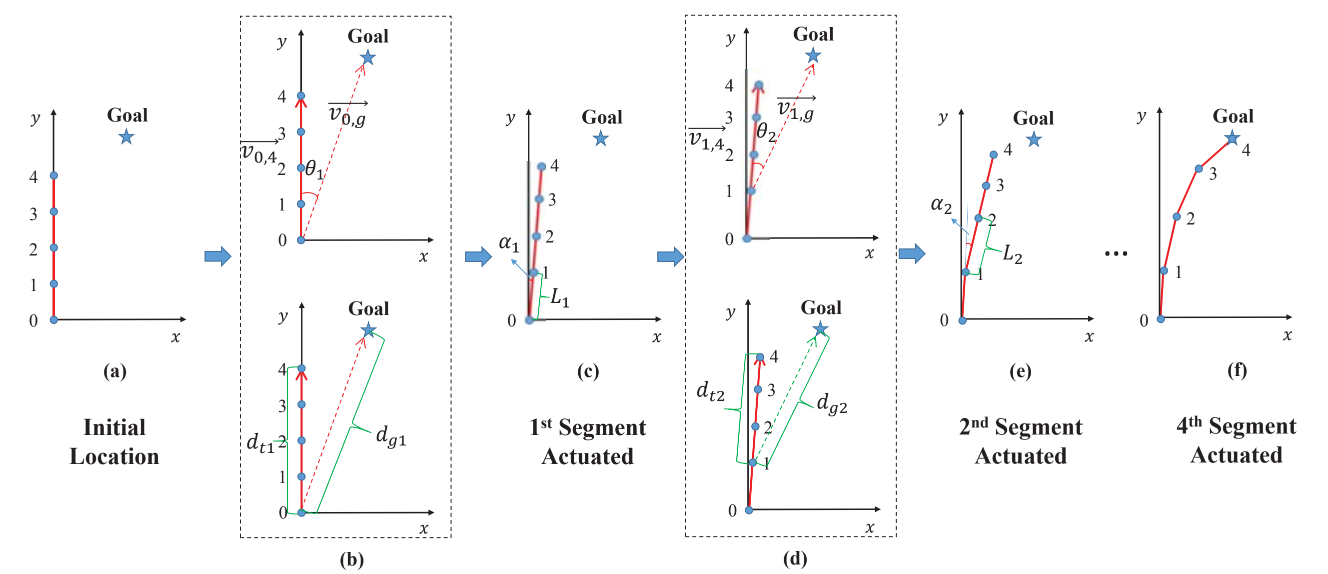


Fig. 2: Illustration of the distributed planning algorithm in a four-segment soft arm for goal reaching.

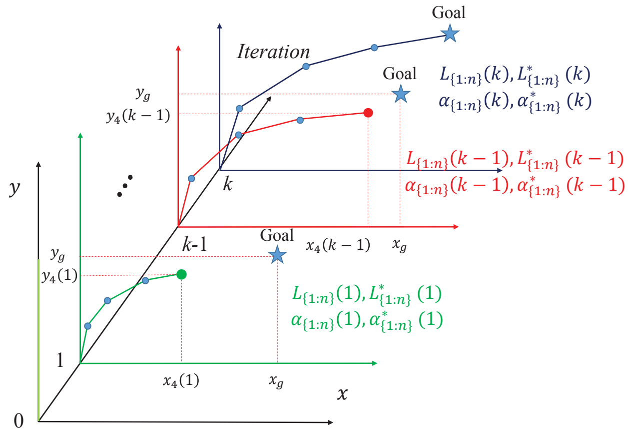


Fig. 3: Conceptual diagram of the distributed iterative learning algorithm of a four-segment soft arm for goal reaching.

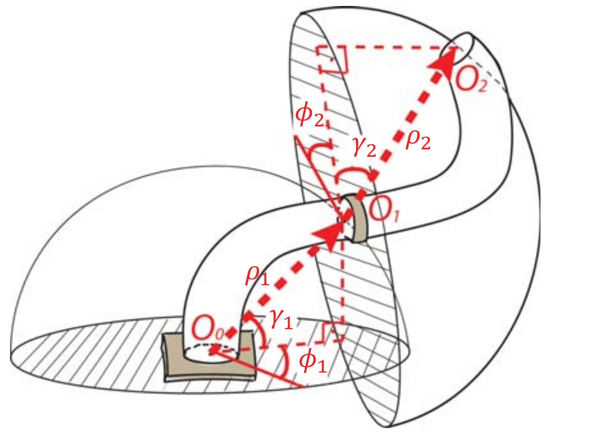


Fig. 4: A soft robot arm in 3D space represented using the spherical coordinate system.

For a given segment i at iteration k, the following learning law is proposed for bending angle

$$\alpha_i(k) = \alpha_i^p(k) + K_{n1}\left[\alpha_i^*(k-1) - \alpha_i(k-1)\right] + K_{f1}\alpha_i(k-1), \tag{1}$$

where $\alpha_i^p(k)$ represents the bending angle calculated using Algorithm 1 and it could change across iterations. It can be seen that the learning algorithm adds two additional terms. The second term learns the bending angle constraint, which means if $\alpha_i^*(k-1) < \alpha_i(k-1)$ (the segment is not bent as much as we want), it will reduce the bending angle setpoint for this iteration to avoid saturation. The third term serves as goal learning, which can be understood as a feedforward term to make the arm reach the goal faster. The second and third terms demonstrate a trade-off between the hardware constraints and goal reaching performance, which can be tuned by the two gain values K_{n1} and K_{f1}. Similarly, the planning for inflation ratio can be achieved by

$$L_i(k) = L_i^p(k) + K_{n2}\left[L_i^*(k-1) - L_i(k-1)\right] + K_{f2}L_i(k-1), \tag{2}$$

where $L_i^p(k)$ is the inflation ratio calculated using Algorithm 1. Similarly, the second and third terms are for constraint and goal learning, respectively. It should be noted the iterative learning laws only use information from the last iteration, and the algorithm is still distributed.

IV. DISTRIBUTED PLANNING IN 3D SPACE

The proposed distributed soft arm planning algorithm with iterative learning can be extended to 3D space. For 3D planning, the spherical coordinate system is employed and three elements (ρ, γ, ϕ) are used to uniquely represent the 3D location of any node on the soft arm. Similar to the 2D case, emphasis will be given to the connecting node between two segments. Therefore, the planning for the i^{th} segment needs to solve for the radial distance ρ_i, elevation angle γ_i, and azimuth angle ϕ_i. The 3D planning approach with iterative learning is summarized in Algorithm 2. The idea is similar to the 2D case, and there are two loops: the outer loop is for iterations, and the inner loop is for different segments in one iteration. Now in 3D space, the angle between two nodes are described using the overall azimuth (Γ) and elevation (Φ) angles, so γ_i^p and ϕ_i^p are determined by averaging the overall angles between remaining segments. Similarly, the inflation ratio is determined by comparing the distances from the base to goal ($\|\overrightarrow{v_{i-1,g}}\|$) and from the base to tip ($\|\overrightarrow{v_{i-1,n}}\|$). The iterative learning follows similar procedures, and now there are six gains to be determined, two for each element in the spherical coordinate system.

V. SIMULATION RESULTS

A. Distributed Planning in 2D Space

In this subsection, two simulation results are presented to validate the efficacy of the distributed learning algorithm and also demonstrate influence of physical limits on the goal reaching performance, A soft arm with 100 segments are simulated, each segment is assumed to be with a length of 1 when it is not actuated, and the end-effector position is set as (0,100) initially. It is assumed that for the segments that have not been actuated yet, they will just keep the original length and point to the same direction as the current actuated segment. Fig. 5 shows the soft arm location after the planning of each segment. The bold black line shows the final configuration of the soft arm. The goal location is set as (15,55), the limit of inflation ratio is $L_i \in [0.5, 1.5]$, and the limit of the bending angle is $|\alpha_i| \leq 90°$. The end-effector of the soft arm stops at (15.03,55.00), so the soft arm can reach the goal position accurately with in one trial because the arm is allowed to compress to half of its original length. However, when we keep the same goal position but change the physical limit to $L_i \in [0.85, 1.15]$ and $|\alpha_i| \leq 60°$, the soft arm cannot reach the goal position with only the distributed planning and the end-effector stops at (18.96,52.29), as is shown in Fig. 6a. The reason is that the planning algorithm does not know the physical limits, and only the last few segments can be planned when the algorithm realizes the constraints. Since the last segment cannot be extended as much as we want, the end-effector cannot reach the goal position. Similarly, when

Algorithm 2 Distributed planning with iterative learning of a n-segment soft robotic arm for goal reaching in 3D space

Input: $(x_g, y_g, z_g), (x_0, y_0, z_0), (x_n, y_n, z_n),$

Output: $L_{\{1:n\}}(k), \gamma_{\{1:n\}}(k), \phi_{\{1:n\}}(k)$

$k = 1$
while $k \leq N$ **do**
  $i = 1$
  while $i \leq n$ **do**
    // Determine the azimuth and elevation angles $(\gamma_i^p(k), \phi_i^p(k))$.
    overall elevation and azimuth angles $(\Gamma_i(k), \Phi_i(k)) \leftarrow \angle(\overrightarrow{v_{i-1,n}(k)}, \overrightarrow{v_{i-1,g}(k)})$
    planning without learning $\gamma_i^p(k) = \Gamma_i(k)/(n-i+1), \phi_i^p(k) = \Phi_i(k)/(n-i+1)$
    distributed planning with iterative learning $\gamma_i(k) = \gamma_i^p(k) + K_{g1}\left[\gamma_i^*(k-1) - \gamma_i(k-1)\right] + K_{g2}\gamma_i(k-1)$ $\phi_i(k) = \phi_i^p(k) + K_{p1}\left[\phi_i^*(k-1) - \phi_i(k-1)\right] + K_{p2}\phi_i(k-1)$
    // Determine the inflation ratio $L_i(k)$.
    $L_i^p(k) = \left\|\overrightarrow{v_{i-1,g}(k)}\right\| / \left\|\overrightarrow{v_{i-1,n}(k)}\right\|$
    distributed planning with iterative learning $L_i(k) = L_i^p(k) + K_{l1}\left[L_i^*(k-1) - L_i(k-1)\right] + K_{l2}L_i(k-1)$
    Actuate the i^{th} segment based on $\gamma_i(k)$, $\phi_i(k)$, and $L_i(k)$.
    Measure the tip location of the i^{th} segment (x_i, y_i, z_i) and the last segment of the arm (x_n, y_n, z_n)
    $i \leftarrow i + 1$
  end while
  $k \leftarrow k + 1$
end while

the goal position is very challenging, more segments will hit the physical limits, resulting failure in goal reaching.

B. Distributed Planning with Iterative Learning in 2D Space

Given the challenge of unknown physical limits, the iterative learning function is added into the distributed planning, as Eqs. (1) and (2) suggest. The same soft arm with 100 segments is employed in this case. Fig. 6b shows the results after 3 iterations, and the arm is closer to the goal position and it stops at (15.65,53.68). Fig. 6c demonstrates the distributed planning results at the 10^{th} iteration, and the end-effector of the arm stops at (15.15,54.41). It can be seen that the algorithm enables more proactive segment behaviors and the end-effector can successfully reach the goal position. Moreover, compared to Fig. 5, the soft arm shows interesting morphology in this case, which is learned all by itself. This highlights the efficacy of the proposed algorithm under unknown constraints, and it is also computationally friendly without the need of collecting prior training data.

C. Distributed Planning with Iterative Learning in 3D Space

Given the space limit, only simulation results with Algorithm 3 are given with a 40-segment arm. In this case,

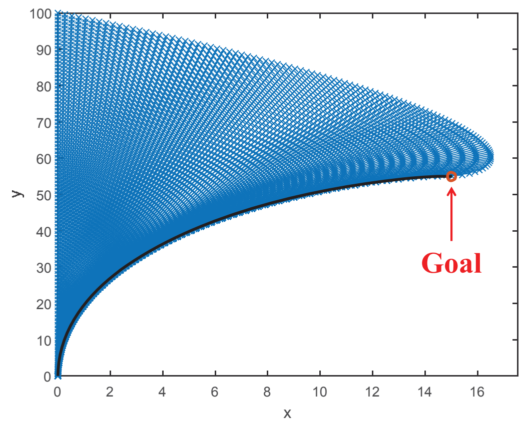


Fig. 5: Simulation results with distributed planning only and less challenging physical limits.

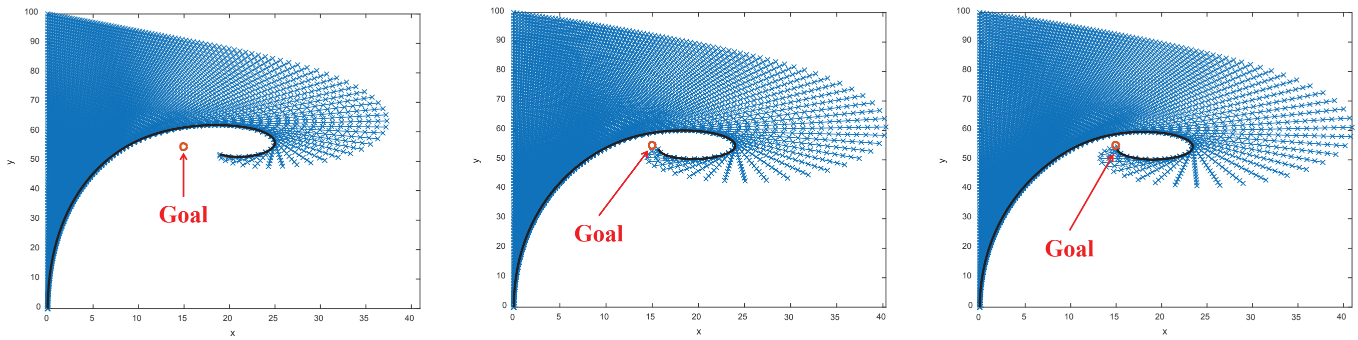
the end-effector is assumed to be at (0,0,40) initially and the goal of the arm is (45,5,5) in 3D space. The physical limits of the arm are set as $L_i \in [0.85, 1.15]$, $|\gamma_i| \leq 60°$, and $|\phi_i| \leq 60°$. Fig. 7 shows the simulation results after 30 iterations, and it can be seen that the arm can reach the goal locations given the physical limits. The end-effector stops at (45.47,4.89,4.83), which is very close to the goal position. It is clear that 3D planning is more challenging given more coupled effects between the configuration variables, but the algorithm can still yield satisfactory planning results, which further demonstrates its efficacy for 3D distributed planning.

D. Discussions

The simulation results demonstrated the good performance of the distributed planning algorithm with iterative learning in both 2D and 3D cases. Specifically, 1) the algorithm allows the soft arm to learn its own configuration and physical constraints given the goal position, 2) the planning algorithm is computationally friendly and can yield good goal tracking performance after a short time of iterative learning, and 3) planning in 3D space is more challenging as the configuration space becomes more complex and the decision variables are strongly coupled. Future work will include understanding the effect of the learning functions and optimizing the iterative learning gains, especially for the 3D case.

VI. CONCLUSIONS AND FUTURE WORK

This paper presented a distributed planning algorithm with iterative learning for a multi-segment soft robot arm. The algorithm planned each segment sequentially from the proximal to distal segments to let the soft arm plan its configuration. For each segment only local information, end-effector location, and goal position were needed, which made the algorithm decentralized and highly scalable. Given the challenge of unknown physical limits, an iterative learning function was added to help each segment learn from the previous trial errors and be more proactive at the beginning segments. Simulation results were presented for both 2D and



(a) Results after 1^{st} iteration (b) Results after 3^{rd} iteration (c) Results after 10^{th} iteration

Fig. 6: Simulation results of distributed planning with iterative learning with challenging physical limits

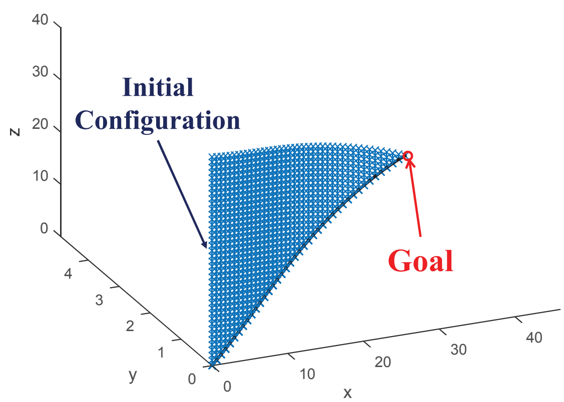


Fig. 7: Simulation results of 3D distributed planning with iterative learning and challenging physical limits.

3D planning, which demonstrated the performance of the proposed algorithm for goal reaching without prior training.

As future work, the proposed algorithm will be integrated with dynamic modeling and control techniques for soft arm control. As suggested in the discussion section, the fine tuning of the learning gains will be studied so the learning effect can be optimized given different soft robot arms and goal locations. Obstacle avoidance and gravity compensation will be integrated into the planning framework for real-world applications. A multi-segment soft arm has been designed and fabricated, and the proposed algorithm will be implemented to verify this algorithm framework experimentally.

References

[1] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-octor," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2005, pp. 2578–2585.

[2] K. J. Kim and S. Tadokoro, "Electroactive polymers for robotic applications," *Artificial Muscles and Sensors*, 2007.

[3] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Advanced Robotics*, vol. 26, no. 7, pp. 709–727, 2012.

[4] I. D. Walker, D. M. Dawson, T. Flash, F. W. Grasso, R. T. Hanlon, B. Hochner, W. M. Kier, C. C. Pagano, C. D. Rahn, and Q. M. Zhang, "Continuum robot arms inspired by cephalopods," in *Proc. SPIE*, vol. 5804, 2005, pp. 303–314.

[5] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Science Robotics*, vol. 2, no. 8, p. eaan3028, 2017.

[6] C. M. Best, M. T. Gillespie, P. Hyatt, L. Rupert, V. Sherrod, and M. D. Killpack, "A new soft robot control method: Using model predictive control for a pneumatically actuated humanoid," *IEEE Robot. Autom. Mag.*, vol. 23, no. 3, pp. 75–84, 2016.

[7] P. H. Nguyen, S. Sridar, W. Zhang, and P. Polygerinos, "Design and control of a 3-chambered fiber reinforced soft actuator with off-the-shelf stretch sensors," *Int. J. Intelligent Robotics and Applications*, vol. 1, no. 3, pp. 342–351, 2017.

[8] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, p. 467, 2015.

[9] F. Iida and C. Laschi, "Soft robotics: challenges and perspectives," *Procedia Computer Science*, vol. 7, pp. 99–102, 2011.

[10] P. Polygerinos, Z. Wang, J. T. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, "Modeling of soft fiber-reinforced bending actuators," *IEEE Trans. Robot*, vol. 31, no. 3, pp. 778–789, 2015.

[11] H. Wang, W. Chen, X. Yu, T. Deng, X. Wang, and R. Pfeifer, "Visual servo control of cable-driven soft robotic manipulator," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2013, pp. 57–62.

[12] B. Mosadegh, P. Polygerinos, C. Keplinger, S. Wennstedt, R. F. Shepherd, U. Gupta, J. Shim, K. Bertoldi, C. J. Walsh, and G. M. Whitesides, "Pneumatic networks for soft robotics that actuate rapidly," *Advanced Functional Materials*, vol. 24, no. 15, pp. 2163–2170, 2014.

[13] S. Ozel, E. H. Skorina, M. Luo, W. Tao, F. Chen, Y. Pan, and C. D. Onal, "A composite soft bending actuation module with integrated curvature sensing," in *IEEE Int. Conf. Robotics and Autom.*, 2016, pp. 4963–4968.

[14] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *Int. J. Robotics Res.*, vol. 35, no. 8, pp. 1000–1019, 2016.

[15] H. Zhao, J. Jalving, R. Huang, R. Knepper, A. Ruina, and R. Shepherd, "A helping hand: Soft orthosis with integrated optical strain sensors and emg control," *IEEE Robot. Autom. Mag.*, vol. 23, no. 3, pp. 55–64, 2016.

[16] A. D. Marchese and D. Rus, "Design, kinematics, and control of a soft spatial fluidic elastomer manipulator," *Int. J. Robotics Res.*, vol. 35, no. 7, pp. 840–869, 2016.

[17] K. Wu, L. Wu, and H. Ren, "Motion planning of continuum tubular robots based on centerlines extracted from statistical atlas," in *IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2015, pp. 5512–5517.

[18] A. Melingui, O. Lakhal, B. Daachi, J. B. Mbede, and R. Merzouki, "Adaptive neural network control of a compact bionic handling arm," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 2862–2875, 2015.

[19] L. Margheri, C. Laschi, and B. Mazzolai, "Soft robotic arm inspired by the octopus: I. from biological functions to artificial requirements," *Bioinspiration & Biomimetics*, vol. 7, no. 2, p. 025004, 2012.

[20] R. Kang, D. T. Branson, T. Zheng, E. Guglielmino, and D. G. Caldwell, "Design, modeling and control of a pneumatically actuated manipulator inspired by biological continuum structures," *Bioinspiration & Biomimetics*, vol. 8, no. 3, p. 036008, 2013.