MCTS Energy Optimal Trajectory for Variable Joint Stiffness Underactuated Finger

Triniti Armstrong[1] Ali Jones[2] Satyam Bhawsinghka[3]

I. Introduction

Underactuated, tendon-driven, compliant grippers have been shown to be more robust compared to rigid grippers [1][2] [3]. However, the application of robots in increasingly unstructured, challenging environments and to non-trivial manipulation tasks require more adaptive end-effectors. Underactuated tendon driven grippers lack direct control over joint angles, which restricts them from generating arbitrary desired trajectories in configuration space. The stiffness of joints plays a crucial role in how these grippers close around an object (i.e the pose of each member link of the gripper with respect to the object). Therefore, by actively tuning the joint stiffness, grasp mechanics can be altered on the fly for any desired task (Fig. 2). Various design solutions exist to vary joint stiffness for adaptive grasping [4][5][6].

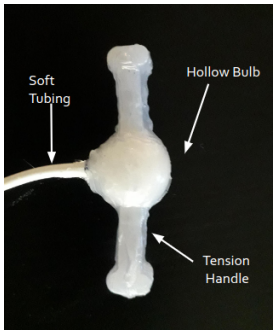


Fig. 1: Hydraulic knuckle is used as a mean to tune joint stiffness.

In this work, a hydraulic knuckle (Fig. 1) is used as a means to vary joint stiffness in an underwater environment. It has a hollow bulb in the center with tension handles on either sides and is made of silicone (Dragonskin 10). The knuckle is pressurized with water from the outside environment to change its stiffness. The joint knuckle stiffness is characterized by it's internal pressure and joint state. The system under consideration consists of a parallel gripper with two fingers, each with three links that are connected to each other and a base with hydraulic knuckles (Fig. 3). The system is tendon driven with no direct control over joint angles. Instead, the internal pressures of the knuckles (P_1, P_2, P_3) and tendon tension (T) are used as control inputs as a means to modify the system configuration (joint angles).

This work deals with the task of transitioning the system from a given starting configuration to any desired ending configuration while expending minimal actuator effort and complying with actuator constraints. The dynamics of this underactuated soft system are not only highly non-linear [7] but also uncertain; therefore, nontrivial control strategies must be developed for modifying the configuration for energy optimal control.

{[1]armsttri, [2]jonesal9, [3]bhawsins}@oregonstate.edu

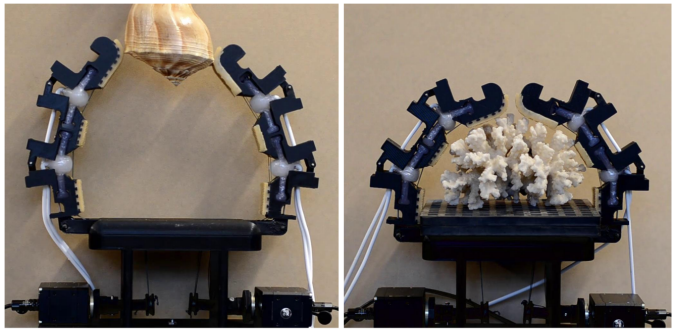


Fig. 2: Grasp primitives achieved by modulating joint stiffness for planar tendon driven gripper: Pinch (left) and Wrap (right).

Monte Carlo methods have shown success in developing energy optimal control strategies for path planning under uncertainty in both continuous and discrete action spaces. Monte Carlo Tree Search [8] and its extension to Monte Carlo Path Planning [9] have been used to solve both MDP and POMDP problems. In this work, our contributions include the following:

- We develop a framework for Monte Carlo Plath Planning in action space to generate valid control sequences for an variable stiffness, underactuated, tendon-driven gripper. These trajectories minimize actuator energy.
- We present the results for total planning time and path cost for two different grasp configurations.
- We compare the effect of transition uncertainty on algorithm performance.
- We discuss the effect of varying neutral joint configurations on algorithm convergence.

II. Related Work

There are a number of works that aim to find optimal grasps for problems modeled as Markov Decision Processes (MDPs) or Partially Observable Markov Decision Processes (POMDPs). For example, Garg et al. 2019 [10] models adaptive grasping as a POMDP, solves for a set of objects, and then trains a grasp policy using an RNN for fast execution. Marino et al., 2019 [11] developed Deep Bayesian Neural

Networks to solve an MDP planning problem. However, we were interested in methods that could potentially apply to both MDPs and POMDPS; we formulated our problem as an MDP (see section III) to focus on algorithm implementation, but in the future it could be extended to a POMDP representation.

Monte Carlo Tree Search (MCTS) methods have shown success in finding near-optimal policies for both MDP and POMDP problems. Bai et al., 2010 [12] developed Monte Carlo Value Iteration for continuous POMDPs by sampling both the robot's state space and belief space. Janson et al., 2018 [13] developed Monte Carlo Motion Planning for POMDPs, which computes a control sequence in a receding-horizon framework. Lee et al., 2020 [8] used MCTS in continuous space, utilizing gradient-based optimization to address errors that arise from discretization. Finally, Dam et al., 2022 [9] proposed Monte Carlo Path Planning as a general MCTS framework for robotic path planning, providing proof of convergence for both MDPs and POMDPs. Our implementation is most closely related to this final work.

The system dynamics are based on Davidson et al., 2015 [7], and described in section III.

III. Problem Statement

The system dynamics of the gripper under consideration under quasi-static assumption are as follows:

$$K\Delta\theta^T + J_a^T f_a = 0 \tag{1}$$

where

$$J_a = [r_1, r_2, r_3]$$

$$\Delta\theta = [\Delta\theta_1, \Delta\theta_2, \Delta\theta_3]$$

$$K = \begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{bmatrix}$$

Eq. 1 is used to calculate $(\Delta\theta_1, \Delta\theta_2, \Delta\theta_3)$ for a given set of control inputs. K_1, K_2, and K_3 are the stiffnesses of the knuckle joints, and f_a is the tension force of the tendon. $\Delta\theta_1, \Delta\theta_2, \Delta\theta_3$ are the change in joint configuration from their neutral position for the proximal, middle, and distal joints, respectively. J_a is the actuator Jacobin, where r_j is the distance of the j'th knuckle to the tendon (set constant at $r_j = 1$ for this work). Fig. 3 shows a finger schematic defined by these variables.

This work focuses on finding the sequence of control inputs (knuckle pressures and tendon tension) to generate an energy-optimal trajectory from a starting neutral joint configuration to a desired goal joint configuration. The problem is formulated as an MDP, $\langle M = S, A, R, T \rangle$, where S is the state-space, A is the action-space, $R : SxAxS \in \mathbb{R}$ is the reward function, and $T : SxAxS$ is the state transition model.

Here, the state space includes the knuckle pressures and stiffnesses, tendon tension, and corresponding joint configuration. The model is assumed to be perfect and uncertainty is introduced through the mapping between knuckle pressure and knuckle stiffness. The stiffness is considered to vary linearly with pressure with a Gaussian error (Fig. 4).

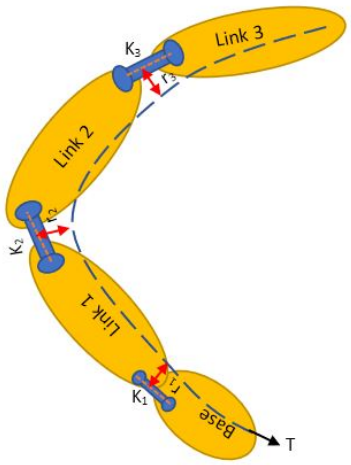


Fig. 3: Finger schematic showing three links and a base with hydraulic knuckles as active joints. K_j is the stiffness of each joint and r_j is the distance of tendon from the center of the joint j. θ_j is the joint angle and $\Delta\theta_j$ is the change in joint angle of joint j from it's neutral position.

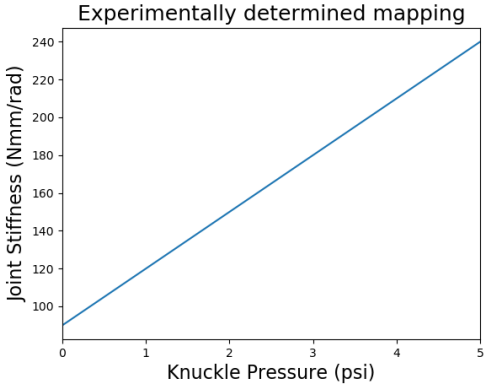


Fig. 4: Ideal mapping of joint stiffness vs knuckle pressure. The actual stiffness is calculated by adding Gaussian noise whose standard deviation is some percentage of ideal stiffness to the ideal stiffness at that pressure.

The set of possible actions at each step includes a change of any one of the control inputs (knuckle pressures and tendon tension) by a set threshold. We assume transition of control variables to be error free. Pressure is constrained to be no more than 5 psi. There is no constraint applied on the upper limit of tension because realistically the tension actuator can produce the required torque.

The actuator effort will be used as the cost (or negative reward) function. The actuator effort is shown in Eq. 2.

$$C = \sum_{i=1}^{n} ((\Delta\theta_i - \Delta\theta_{i-1}) K_i \Delta\theta_{i-1}^T + \gamma P_{i-1} (P_i - P_{i-1})^T) \tag{2}$$

where

$$P = [P_1, P_2, P_3]$$

Here, P is 1x3 matrix of the proximal, middle, and distal knuckle pressures and K is the stiffness matrix and $\Delta\theta$ is 1x3 matrix of change in angle of the proximal, middle, and distal knuckle joints relative to its neutral position. Now P_i, K_i and $\Delta\theta_i$ are these matrices at some instant i. γ is a constant scalar (experimentally determined value = 20). The two terms on the right hand side account for the effort needed to control tendon tension and knuckle pressures, respectively.

Our algorithm aims to minimize the total path cost estimated using Eq. 2 for control sequences between $i = 1$ to n or when the goal is reached.

IV. Algorithm Description

The algorithm that we use to to generate an energy-optimal path for the gripper is Monte Carlo Tree Search (MCTS). MCTS has four main phases: selection, expansion, simulation (or rollout), and backpropagation. During the selection phase, a tree policy (we use a modifed UCT) is used to construct a path from the root, the starting node, to the most promising leaf node. A leaf node is a node with unexplored child nodes. The tree is then expanded by generating any number of child nodes for the selected leaf node. Then, during the rollout phase, an action is simulated from the current state to the goal or until the simulation is terminated, and its reward calculated. Once a terminal state (e.g time budget expanded) or goal is reached, backpropagation begins. MCTS traverses back through the path taken, updating the rewards for each node and incrementing the number of times each node has been visited. The algorithm is described in more detail below and pseudocode can be seen in algorithm 1.

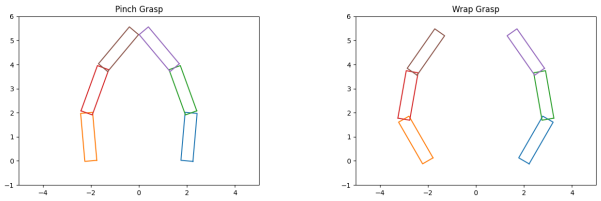
While the MCTS algorithm is widely-known and used, there are several aspects of this algorithm that we develop specifically to our application. These are itemized here:

- The transition function, a mapping which takes state *s* to state *s'* based on action *a*. This function is used to generate child nodes in the expansion phase and valid paths in the simulation phase. It is determined by the MDP problem formulation; it will take one set of pressure and tension values to a second set with the probability associated with the choice of action.
- The reward function, used to determine the value of each state in both the selection and simulation phases. The reward is based off the cost function (Eq. 2), which is a measure of energy that depends on stiffness, pressure, and joint angles.
- The tree policy, which determines the best child nodes to visit during the selection phase. We use a modified Upper Confidence Bound for Trees (UCT). UCT is a common selection policy that balances exploration and exploitation by maximizing a reward while minimizing the number of visits to each node. We use a similar formulation that minimizes cost instead of maximizing reward.
- The depth to which to grow the tree at each expansion step. Since the problem is formulated as an MDP, and therefore has uncertainty in its transition model, we replan the best action to take at every step. The depth to which the tree should be grown at each step will impact algorithm performance; a single-step lookahead would perform more greedily, while an infinite horizon would be more computationally expensive.

V. Simulation & Results

The performance of this algorithm was validated on a kinematic Python simulation representation of the underactuated gripper. The quasistatic equation (Eq. 1) was used to calculate joint angles as the algorithm moved through control space, and the joint angles were used to determine the forward kinematics of the gripper. The kinematics were animated to evaluate the behavior and accuracy of the generated minimum-energy control sequence.

Two grasps - wrap and pinch - were selected to quantitatively and qualitatively validate the minimum-energy path generated. The ideal goal position of these two grasps are shown in Fig. 5.



(a) (b)

Fig. 5: Grasp goal positions with their proximal, middle, and distal joint angles: (a) Pinch $(85°, 25°, 20°)$, (b) Wrap $(60°, 40°, 25°)$

The time taken to reach the goal, the cost expended, and the control values along the path were compared for each grasp. Additionally, time and cost were recorded for different neutral positions of the gripper.

	Time (sec)	Cost (N-mm)
Wrap Grasp	21.0 (σ1.1)	135.3 (σ7.3)
Pinch Grasp	17.4 (σ1.6)	153.4 (σ6.6)

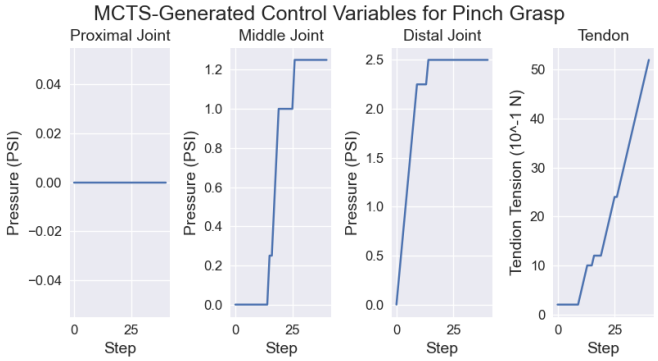
TABLE I: Average time and cost of the two grasps over 10 trials without noise.

Table I shows the average performance of our algorithm on two different grasp configurations, wrap and pinch (Fig. 5). The wrap grasp took a longer time to reach the goal while expanding less of a cost. Both configurations take a feasible amount of time to generate and have reasonably low variance. Fig. 6 shows how the control variables change along the path for the same two grasp configurations.

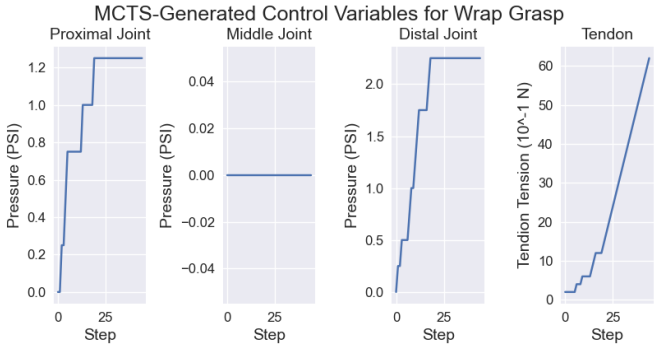
Additionally, we compared the performance of the two grasps at varying proximal knuckle neutral configurations. Fig. 7 shows the effect of varying the neutral configuration between $(25°, 0°, 0°)$ and $(60°, 0°, 0°)$ on the performance (final path cost and runtime to reach the goal). The generated control sequences didn't reach the goal for neutral proximal

Algorithm 1 MCTS

procedure SEARCH
  $T = Tree$ ▷ Nodes are states (s) and branches are actions (a)
  $P = Policy$
  while *Goal* not Reached **do**
    $T, r \leftarrow$ SELECTANDEXPAND(T)
    $r \leftarrow$ SIMULATE(s, r)
    BACKPROPAGATE(T, r)
    BESTACTION()
    $P.append($BESTACTION()$)$
    $s \leftarrow s'$
  end while
end procedure
function SELECTANDEXPAND(T)
  $r = 0$ ▷ Total reward
  $s = T.state$
  while *SelectandExpand* not terminated **do**
    $a \leftarrow$ SELECTACTION() ▷ Heuristically selection action
    $s' \leftarrow Transition(s, a)$ ▷ Sample transition
    $r \leftarrow r + Reward(s, a, s')$ ▷ Reward for taking action
    if $T.Children$ is None **then**
      $T.Children \leftarrow Node()$ ▷ Add leaf node
    end if
    return $T.Children$, r
  end while
  $T \leftarrow T.Children$
  $s \leftarrow s$
  return T, r
end function
function BACKPROPAGATE(T, r)
  while T is not Null **do**
    $T.Reward \leftarrow T.Reward + r$
    $T.Count \leftarrow T.Count + 1$
    $T \leftarrow T.Parent$
  end while
end function
function SIMULATE(s, r)
  while *Simulation* not terminated **do**
    $a \leftarrow RandomAction$
    $s' \leftarrow Transition(s, a)$ ▷ Sample transition
    $r \leftarrow r + Reward(s, a, s')$ ▷ Reward for taking action
  end while
  return r
end function
function SELECTCHILD(T)
  for *Children* **do**
    $SelectionPolicy.append($
    $\frac{Child.cost}{Child.count} - C \times \sqrt{\frac{log(Parent.Count)}{Child.Count}}$
    $)$
  end for
  return *argmin SelectionPolicy*
end function
function BESTACTION
  return *BestNode* ▷ Return node with highest reward and lowest count
end function



(a) Pinch grasp control variables along the path, starting at a neutral joint configuration of $(50°, 0°, 0°)$



(b) Wrap grasp control variables along the path, starting at a neutral joint configuration of $(30°, 0°, 0°)$

Fig. 6: Performance of pinch and wrap grasp control variables without noise. The pinch grasp configuration ends with a higher distal knuckle pressure, which is in line with expected grasp behavior.

angles below 35° for pinch grasps, and above 45° for wrap grasp within a cut-off runtime of 100 secs. There was an increase in execution time and final path cost with the increase in angle for wrap grasp and a decrease for pinch grasp.

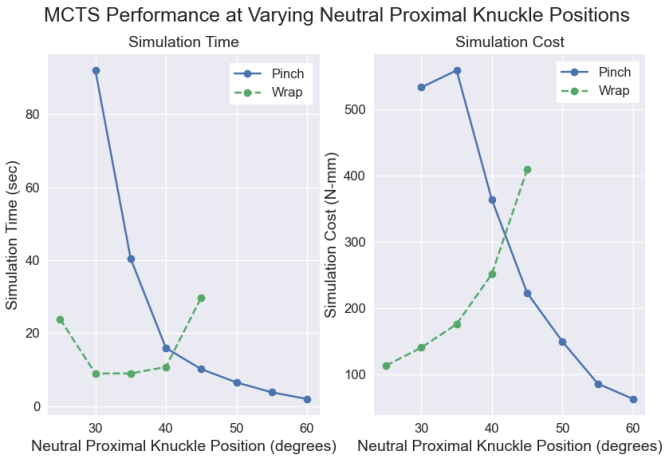


Fig. 7: Algorithm time and path cost for wrap and pinch grasps starting from different neutral proximal knuckle positions. Middle and distal knuckle neutral positions are 0°.

We also studied the effect of extend of transition uncertainty on algorithm's performance. The uncertainty is expressed via the standard deviation of the Gaussian noise in stiffness estimation as the percentage of ideal stiffness from the stiffness pressure mapping. The transition uncertainty is varied by varying this percentage. It can be observed in Fig. 8 that both the final path cost and runtime for reaching the goal generally increases as transition uncertainty increases.

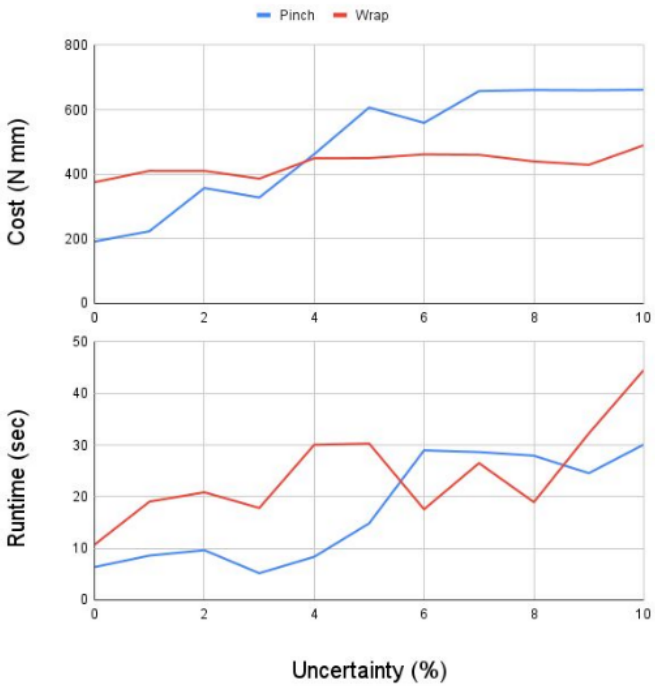


Fig. 8: Effect of uncertainty on performance (final path cost and runtime to reach the goal) for pinch and wrap grasp as goals from a neutral starting joint configuration of $(45°, 0°, 0°)$.

VI. Conclusion

In this work, we have demonstrated that Monte Carlo Tree Search can be used to find energy-optimal trajectories for an underactuated, tendon-driven, compliant gripper. The control sequences for wrap and pinch grasp primitives were generated using the algorithm which suggests a stiffer distal joint is required to execute a pinch grasp which conforms with the general notion. The effect of varying the neutral position of proximal joint for different grasp primitive on path cost and runtime were presented. This would have design implications while developing a physical gripper prototype. Finally, the effect of transition uncertainty was studied which suggests the algorithm was able to generate control sequences to reach the desired goal even with 10% uncertainty.

VII. Future Work

In this work, it is assumed that the stiffness is dependent on pressure alone. However, in reality the stiffness of the knuckle also depends on the joint state. Moreover, the pressure in the knuckle changes as the angle changes. For simplicity, the problem was formulated as an MDP but in future for a more accurate representation the problem will be formulated as a POMDP where the change in pressure with the angle will serve as the observation model and a more accurate transition model will be considered. In future, we will consider inertia and gravity in the system dynamics model. To improve the runtime of the algorithm, we will include some random simulations along with the greedy simulations during the rollout phase. Also, instead of expanding all nodes, some random nodes can be expanded during the expansion phase. In future, we will work on tuning these parameters for optimal performance.

References

[1] A. M. Dollar and R. D. Howe, "The highly adaptive sdm hand: Design and performance evaluation," *The international journal of robotics research*, vol. 29, no. 5, pp. 585–597, 2010.

[2] L. Birglen, T. Laliberté, C. Gosselin, L. Birglen, T. Laliberté, and C. Gosselin, "Design and control of the laval underactuated hands," *Underactuated robotic hands*, pp. 171–207, 2008.

[3] R. R. Ma, L. U. Odhner, and A. M. Dollar, "A modular, open-source 3d printed underactuated hand," in *Proc. IEEE International Conference on Robotics and Automation*, 2013, pp. 2737–2743.

[4] A. Firouzeh and J. Paik, "Grasp mode and compliance control of an underactuated origami gripper using adjustable stiffness joints," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 5, pp. 2165–2173, 2017.

[5] E. Fox and F. L. Hammond, "Soft variable stiffness joints for controllable grasp synergies in underactuated robotic hands," in *Proc. 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 586–592.

[6] H. Godaba, A. Sajad, N. Patel, K. Althoefer, and K. Zhang, "A two-fingered robot gripper with variable stiffness flexure hinges based on shape morphing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8716–8721.

[7] J. R. Davidson and C. Mo, "Mechanical design and initial performance testing of an apple-picking end-effector," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 57397. American Society of Mechanical Engineers, 2015, p. V04AT04A011.

[8] J. Lee, W. Jeon, G.-H. Kim, and K.-E. Kim, "Monte-carlo tree search in continuous action spaces with value gradients," in *Proc. AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4561–4568.

[9] J. P. T. Dam, G. Chalvatzaki and J. Pajarinen, "Monte-carlo robot path planning," in *IEEE Robotics and Automation Letters*, vol. 08, no. 04, 2022, p. 11213–11220.

[10] N. P. Garg, D. Hsu, and W. S. Lee, "Learning to grasp under uncertainty using pomdps," in *Proc. International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2751–2757.

[11] D. L. Marino and M. Manic, "Modeling and planning under uncertainty using deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 8, pp. 4442–4454, 2019.

[12] H. Bai, D. Hsu, W. S. Lee, and V. A. Ngo, "Monte carlo value iteration for continuous-state pomdps," in *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2011, pp. 175–191.

[13] L. Janson, E. Schmerling, and M. Pavone, "Monte carlo motion planning for robot trajectory optimization under uncertainty," in *Robotics Research: Volume 2*. Springer, 2017, pp. 343–361.