

# 動的環境に対応したコンフィギュレーション時空間による動作計画

## Configuration-time-space for Motion Planning in Dynamic Environment

学 木南 貴志 (中部大) 山内 悠嗣 (中部大)

Takashi KINAMI, Chubu University, er17026-6673@sti.chubu.ac.jp

Yuji YAMAUCHI, Chubu University, yuu@isc.chubu.ac.jp

A robot in a dynamic environment needs to avoid static obstacles and perform motion planning while considering dynamic obstacles. This paper presents a Configuration-time-space (C-time-space), in which time-series information is added to the configuration space (C-space). The C-time-space is represented by the mapping position and joint displacement of the robot at each time in the generalized coordinate space. Motion planning using the C-time-space can generate a trajectory that avoids dynamic obstacles. The experimental evaluation results indicate that compared to the conventional method, the trajectory distance of the robot arm can be shortened in environments where dynamic objects exist using the proposed method.

**Key Words:** Motion Planning, Dynamic Environment, Configuration-time-space

### 1 はじめに

近年、労働力の不足や人件費の削減などの理由により、単純な作業をロボットが担うケースが増加している。産業用ロボットの多くは、ロボットを人間から隔離した場所に配置し、十分な安全が確保された静的な環境下で運用されることが多い。近年では、限られた狭いスペースで動作するロボットや高度なタスクが要求されることから、人間と同じ空間で作業する協調ロボットの需要が高まっている。人間と同じ空間で安全に作業するためには、静的な環境下だけでなく、人間を含む動的な障害物にも対応できるような動きが求められる。

ロボットの自動化には教示と動作計画の2種類が存在する。教示は作業者がロボットを操作して覚えさせる方法であり、直感的に動作を学習させることができる。しかし、動作を学習させることに多くの時間が必要になることや、80w以上のロボットの教示には資格が必要になるなどの多くの問題が存在するため、近年では動作計画に関する研究が盛んに取り組まれている。ロボットの基礎的技術の1つである動作計画は、ロボットが与えられたタスクを達成するための動作を計画することである。一般に Configuration-space(C-space)と呼ばれるロボットの姿勢を表現した空間を用いて、初期姿勢から目標姿勢までの最適な軌跡を生成する。センサ等で未知の障害物を発見した場合、得られたセンサ情報から障害物のみを抽出して障害物とロボットアームとの位置関係を算出する。そして、センサの座標系をロボットの座標系に変換した後、障害物の位置が更新された状態で動作計画を実行することで障害物を避けるような軌跡を生成する。

動作計画手法は現在までに様々な手法 [1, 2, 3] が提案されており、産業用ロボットに適用されている。多くの動作計画手法は空間中を移動する物体が存在しないと仮定した静的な環境下で計画をする。人間が同じ空間で作業するような動的な環境下においては、ロボットの動作中に人間を含む動的障害物に衝突する危険性がある。また、動的障害物を回避するために軌跡を再計算することが必要になるため、非効率な動作計画となる。

そこで、本研究では C-Space を拡張して時系列変化に対応させた Configuration-time-space(C-time-space) を提案する。C-time-space は C-space に時間軸を追加することで静的物体に加えて、各時間の動的物体を表現した空間である。C-time-space を導入することにより動的な障害物を考慮したロボットの軌跡を生成することが可能となる。

### 2 関連研究

多くの動作計画手法は C-space 上で最適経路を求めることを問題設定としている。C-space を用いた動作計画では、C-space 上で初期姿勢と目標姿勢を設定して障害物に衝突しないような経路を求める。この際に求めた経路がロボットの軌跡に相当し、軌跡

に従するようにロボットを動作させることで障害物に衝突しないような動作が可能となる。

#### 2.1 静的な環境下における動作計画法

現在に至るまでに、動作計画法に関する研究が多く行われてきた。動作計画法は軌跡を生成する過程によって、幾つかの種類に分けられる。

##### ポテンシャルベース

目標姿勢に引力ポテンシャル、障害物に斥力ポテンシャルを仮想的に設置したポテンシャル場を作成することで目標姿勢までの軌跡を生成するポテンシャルベースの手法 [4] が提案されている。この手法は、障害物の回避と目標姿勢への到達を同時に実行できる。一方で、ポテンシャル関数の勾配が0になる停留現象が生じる可能性がある。また、3次元以上の多次元空間において計算量が膨大になる。

##### ランダムサンプリングベース

C-space 上にランダムにノードをサンプリングし、ノード間を結ぶことで目標姿勢までの軌跡を生成するランダムサンプリングベースの手法が提案されている。代表的な手法としては Probabilistic Road Map(PRM)[5] や Rapidly-exploring Random Tree(RRT)[6] が存在する。PRM と RRT はどちらもノードをランダムにサンプリングする手法であるが、前者は事前にサンプリングを行うのに対して、後者は逐次的にサンプリングを行う。PRM では、事前にランダムにサンプリングされたノード間を結ぶことで軌跡を生成する。RRT では、ランダムにサンプリングされたノードと最も近いノードを結ぶことを逐次的に繰り返すことで目標姿勢までの軌跡を生成する。RRT は高速に動作し実装も容易なことから、最適性を付与した RRT\*[7] や、RRT を拡張した多くの手法 [8, 9, 10, 11] が提案された。また、高次元の C-space においても比較的高速に動作することから多軸ロボットの動作計画にも用いられる。

##### 軌跡最適化ベース

勾配法を用いて、軌跡の生成と最適化を同時に実行する軌跡最適化ベースの手法 [12, 13] が提案されている。軌跡最適化法では、最初に初期姿勢と目標姿勢を直線で結び、その直線上に一定間隔でノードを設置する。そして、勾配法を用いて各ノードのコストが最小になるようにノードを移動させることで最適な軌跡を生成する。コストの例としては、障害物との距離などがある。この手法は、ノードの数が一定であるという利点が存在する。一方で、勾配法を用いるため局所的な解に収束する可能性がある。

##### 学習ベース

動作計画法に機械学習を導入した手法が提案されている。深度センサーから取得した点群データ、ロボットの初期姿勢、目標姿勢を学習データとして軌跡を学習する手法 [14] や、強化学習とランダムサンプリングによる手法を組み合わせることで学習効率を維

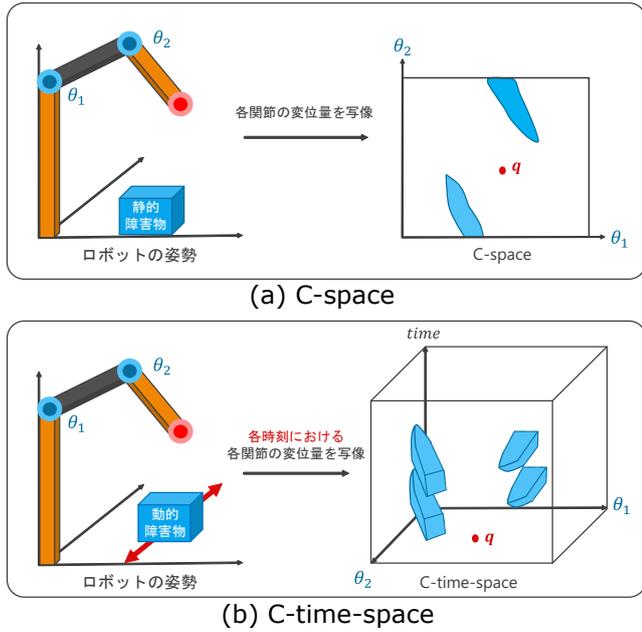


Fig.1 C-space and C-time-space.

持ちながら高速に推論する手法 [15] などが提案されている。学習ベースによる手法は学習データが大量に必要であることが多い。一方、障害物が未知の環境下でも動作する利点がある。

これらの静的な環境下における動作計画法は、安全が確保された環境においては効率よく動作する。しかし、動的な環境では動的物体を発見した際に軌跡の再計画が必要になり無駄のある軌跡を生成する。また、障害物が回避できなかった場合は障害物に衝突する恐れがあり人間に危険が及び可能性も存在する。

## 2.2 動的な環境下における動作計画法

近年では動的環境に対応した動作計画法も提案されている。Levenら [15] は、PRM をベースとした Dynamic Roadmaps (DRM) [16] を提案した。DRM は障害物が存在しないと仮定した環境下でロードマップ法によりグラフを作成し、軌跡を追従する際に動的物体を観測した場合にグラフを部分的に修正し、軌跡を再計画することで動的物体を回避する。Kunzら [17] は、DRM をベースとした手法 [17] を提案している。この手法では、衝突判定の回数を可能な限り削減することで人間の反応速度の約 2 倍の 100ms 未満での動作計画に成功している。Sungら [18] は、人間の動作を機械学習によって予測することで安全な動作計画を実行する Intention-Aware Motion Planning (IAMP) [18] を提案した。

上記の手法は動作計画法を提案している。一方で本手法は動的環境における空間表現方法を提案しているため、今回の評価実験で適用した RRT\* の他にも様々な動作計画法と組み合わせることが可能である。そのため、動作する環境によって適切な動作計画法を採用することができる。

## 3 提案手法

提案手法は、C-time-space の作成と C-time-space 上での経路計画で構成される。

### 3.1 C-space

C-space は各時刻におけるロボットの位置や各関節の変位等を一般化座標空間に写像した空間である。 $N$  軸のロボットアームの場合、 $n$  番目の関節の変位量を  $\theta_n$  とすると時刻  $t$  における  $N$  軸ロボットアームの関節変位と時刻のベクトルは  $q = (\theta_1, \theta_2, \dots, \theta_N, t)$  となり、C-time-space 上でノードとして表現される。C-time-space 全体を  $C_{all}$ 、障害物と衝突、もしくは自己干渉する空間を  $C_{collision}$  とすると C-time-space 上でロボットが動作可能な空間は  $C_{free} = C_{all} \setminus C_{collision}$  として表現される。また、C-time-space においてノード  $q$  が設置可能な領域

は  $q \in C_{free}$  となる。従来の C-space と C-time-space の概要を図 1 に示す。図 1 からわかるようにロボット周辺で動的な障害物が周期的な動作を行うと、C-time-space では C-space を時間軸方向に重ねるように衝突領域が写像される。

### 3.2 C-time-space 上での経路計画

C-time-space 上での経路計画では  $C_{free}$  内で最適な経路を探索する。提案手法は動作計画法として様々な手法が適用可能であるが、今回は動作計画に広く用いられる RRT\* を適用する。ただし、生成される経路は時間軸方向に常に増加するように制約を設ける。これは、時間に逆行するような経路の生成を防止するためである。初期関節変位  $q_{start}$  と目標関節変位  $q_{goal}$  を結ぶ経路を  $Q = (q_{start}, q_2, \dots, q_{goal-1}, q_{goal})$  とした時、C-time-space 内での最適経路  $Q^*$  は式 (1) で表現される。

$$Q^* = \arg \min_{Q \in C_{free}} Q \quad (1)$$

### RRT\*による C-time-space 上での経路計画

C-time-space における RRT\*による経路計画のアルゴリズムを Algorithm 1 に示す。基本的な手順は RRT\*と同様であるが、生成する経路の時間成分が常に増加するように制約を設ける。新しいノード  $q_{new}$  を設置する際に、 $q_{new}$  の時間要素  $t$  が、接続する親ノードの時間要素  $t$  よりも小さい場合は  $q_{new}$  を設置しない。これにより時間を逆行する経路の生成を防止する。また、再接続の際に子ノードの時間要素  $t$  が親ノードの時間要素  $t$  よりも小さい場合はノードの再接続を行わない。2 軸ロボットアームを使用した C-Time-Space 上での経路計画の結果例を図 2 に示す。図 2 より初期生成から目標姿勢までを結ぶ最適経路が生成できていることが確認できる。

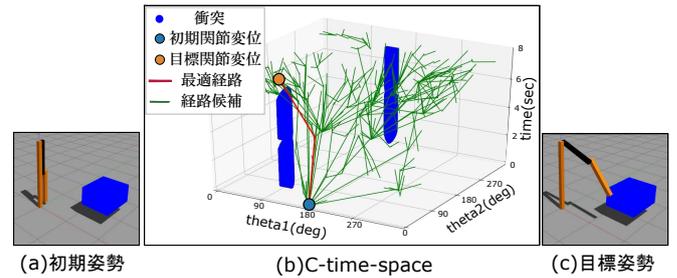


Fig.2 path planning on C-time-space.

## 4 評価実験

提案手法の有効性を確認するために評価実験を行う。

### 4.1 実験概要

従来の C-space を使用した動作計画と C-time-space を使用した動作計画を比較する。評価実験にはロボットシミュレータ Gazebo [19] を用いる。ロボットアームには 2 軸ロボットアーム RRBot<sup>1</sup> を使用する。RRBot の各軸の回転範囲は  $[-180, 180)$  となる。初期姿勢と目標姿勢が異なる 3 シーンを用意し、ロボットアームの手先の軌跡の距離と経路計画の処理時間を比較する。この際、軌跡を追従する時間は処理時間に含まない。乱数による結果のばらつきを抑制するため、同条件で 5 回の実験を行い、その平均距離を比較する。従来法では、動的障害物を事前に知ることができないため、動的障害物に衝突した場合、改めて軌跡を再計画する。

### 4.2 実験結果

提案手法と従来法の比較の結果を表 1、経路計画で生成された経路の結果を図 4 に示す。最初に、手先の軌跡の距離を比較する。図 5 に各シーンにおいて生成された軌跡を示す。赤色の線が手先の軌跡である。従来法では、動的障害物に妨害され、無駄のある軌跡を生成した。一方、提案手法では動的障害物を避けた軌

<sup>1</sup>[https://github.com/ros-simulation/gazebo\\_ros\\_demos](https://github.com/ros-simulation/gazebo_ros_demos)

---

**Algorithm 1: RRT\* on C-time-space**


---

```

1  $V \leftarrow \{q_{start}\}; E \leftarrow \emptyset;$ 
2 for  $i = 1, \dots, n$  do
3    $q_{rand} \leftarrow \text{SampleFree}_i;$ 
4    $q_{nearest} \leftarrow \text{Nearest}(G = (V, E), q_{rand});$ 
5    $q_{new} \leftarrow \text{Steer}(q_{nearest}, q_{rand});$ 
6   if  $\text{ObstacleFree}(q_{nearest}, q_{new})$  then
7      $Q_{near} \leftarrow \text{Near}(G = (V, E), q_{new}, \min\{\gamma * (\log |V|/|V|)^{1/d}, \epsilon\});$ 
8      $V \leftarrow V \cup \{q_{new}\};$ 
9      $q_{min} \leftarrow q_{nearest}; c_{min} \leftarrow \text{Cost}(q_{nearest}) + c(\text{Line}(q_{nearest}, q_{new}));$ 
10    foreach  $q_{near} \in Q_{near}$  do
11      if  $\text{CollisionFree}(q_{near}, q_{new}) \wedge \text{Cost}(q_{near}) + c(\text{Line}(q_{near}, q_{new})) < c_{min}$  then
12        |  $q_{min} \leftarrow q_{near}; c_{min} \leftarrow \text{Cost}(q_{near}) + c(\text{Line}(q_{near}, q_{new}));$ 
13    if  $t \in q_{min} < t \in q_{new}$  then
14      |  $E \leftarrow E \cup \{q_{min}, q_{new}\};$ 
15      foreach  $q_{near} \in Q_{near}$  do
16        | if  $\text{CollisionFree}(q_{new}, q_{near}) \wedge \text{Cost}(q_{new}) + c(\text{Line}(q_{new}, q_{near})) < \text{Cost}(q_{near}) \wedge t \in q_{new} < t \in q_{near}$  then
17          | |  $q_{parent} \leftarrow \text{Parent}(q_{near});$ 
18          | |  $E \leftarrow (E \setminus \{(q_{parent}, q_{near})\}) \cup \{(q_{new}, q_{near})\};$ 
19    else
20      |  $V \leftarrow V \setminus \{q_{new}\};$ 
21 return  $G = (V, E)$ 

```

---

跡を生成するため、より短い距離で目標姿勢に到達することができた。次に、経路計画の処理時間を比較する。表 1 より、シーケンス 1、シーケンス 2 においては両手法とも差がみられないことが分かる。シーケンス 3 においては従来法のほうが処理時間が短くなる結果となった。この結果より、動的物体と衝突する回数が少ないシチュエーションでは従来手法のほうが処理時間が短くなることが確認できた。図 4 の C-space を用いた経路計画の結果を見てもシーケンス 1、シーケンス 2 は 2 回衝突しているが、シーケンス 3 は一度しか衝突していないことが分かる。それに対して、C-time-space を用いた経路計画では障害物に衝突せずに、より直線的な経路を生成できていることが分かる。

#### 4.3 考察

衝突回数が少ないシチュエーションにおいて、従来手法の方が処理時間が短かった要因として C-space が C-time-space に比べて次元が少ないことが挙げられる。RRT\* では、新しいノードを中心とした半径  $r$  内にあるノードの再接続を行うが、この際に次元数が少ない場合は  $r$  の大きさを決定するパラメータ  $\gamma$  が小さくても十分に最適な経路が生成できるため C-space の方が処理時間が短くなったと思われる。一方で、シーケンス 1、シーケンス 2 のような障害物と多く衝突するシチュエーションでは再計画が必要があるため、従来手法の処理時間が増加していることが確認できる。提案手法では全てのシーケンスにおいて再計画を実行していないため、どのシチュエーションでも同じような処理時間となっている。このことから、提案手法は動的物体が多く存在するような環境下では従来手法よりも短い処理時間で動作することが考えられる。

#### 4.4 多軸ロボットへの展開

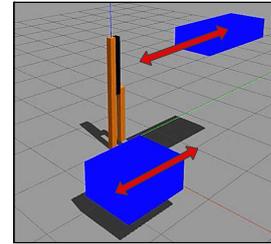
3 軸以上の多軸ロボットにおいても本手法は適用可能である。7 軸ロボットを用いて検証した結果を図 6 に示す。7 軸ロボットアームにおいても動的物体を回避して目標姿勢に到達できていることが確認できる。しかし、C-time-space が多次元になると探索範囲も広くなることから、より最適な経路を求めたい場合はサンプリング回数を増やす必要があり、これに伴い処理時間が増加する。

## 5 おわりに

本研究では、動的環境を考慮した動作計画法を提案した。従来の C-space に時系列情報を付与した C-time-space を用いることで動的障害物を回避する軌跡を生成することができた。今後は 3 軸以上の多軸ロボットを用いた提案手法の評価と、人間の行動予

**Table 1** Summary of results.

シーケンス	手先の軌跡の距離 [m]			平均処理時間 [sec]		
	1	2	3	1	2	3
C-space	8.41	10.13	12.47	12.77	12.12	8.70
提案手法	5.94	6.23	6.77	12.53	12.68	12.16

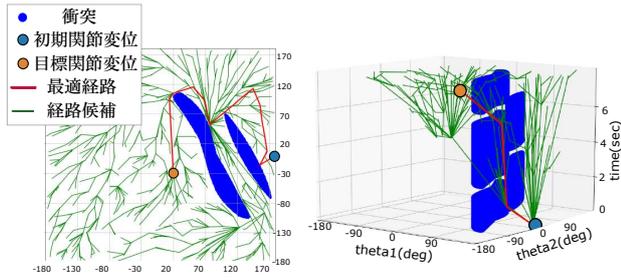


**Fig.3** Simulation environment.

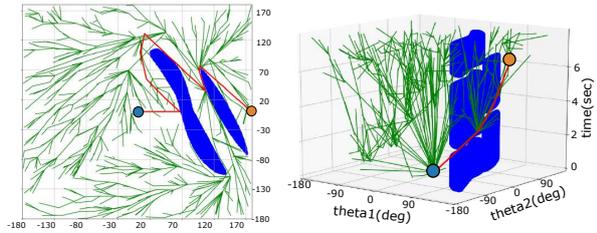
測と組み合わせて動的障害物を人間とした時の人間とロボットの協調動作に関する研究に取り組む予定である。

#### 参考文献

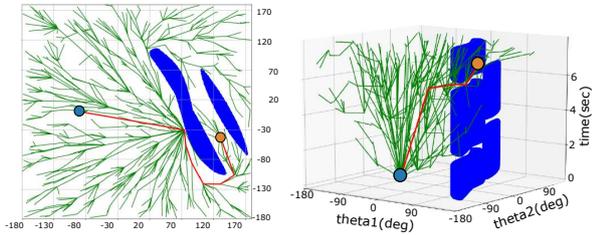
- [1] 森重 功一, 竹内 芳美, “C-Space に基づいた 5 軸制御加工における工具姿勢の決定法”, JSPE, vol. 66, no. 7, pp. 1140–1144, 2000.
- [2] 原田 研介, 万 偉偉, ラミレス・イクシエル, 山野辺 夏樹, 辻 徳生, “データベースに基づく産業用ロボットの作業動作計画”, JRSJ, vol. 37, no. 8, pp. 679–682, 2019.
- [3] H. Wörn, C. Wurll, and D. Henrich, “Automatic off-line programming and motion planning for industrial robots”, *ISR*, 1998.
- [4] Y. Koren, and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation”, vol. 2, pp. 1398–1404, 1991.
- [5] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces”, *IEEE Trans. Rob. Autom.*, vol. 12, no. 4, pp. 566–580, 1996.
- [6] S. M. Lavalle, “Rapidly-exploring random trees: A new tool for path planning”, *technical report*, 1998.



(a)シーケンス1



(b)シーケンス2

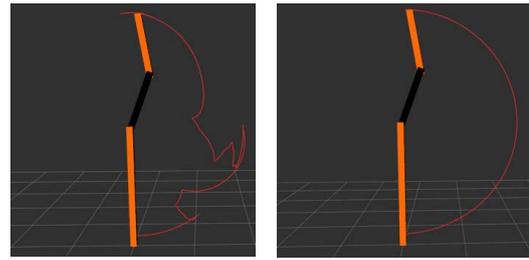


(c)シーケンス3

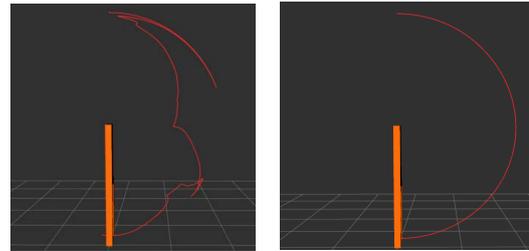
C-space

C-time-space

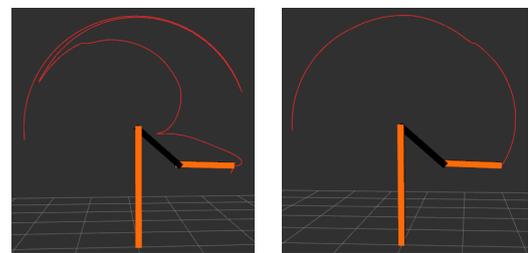
Fig.4 Comparison of C-space and C-time-space.



(a)シーケンス1



(b)シーケンス2



(c)シーケンス3

C-space

C-time-space

Fig.5 Comparison of hand trajectory.

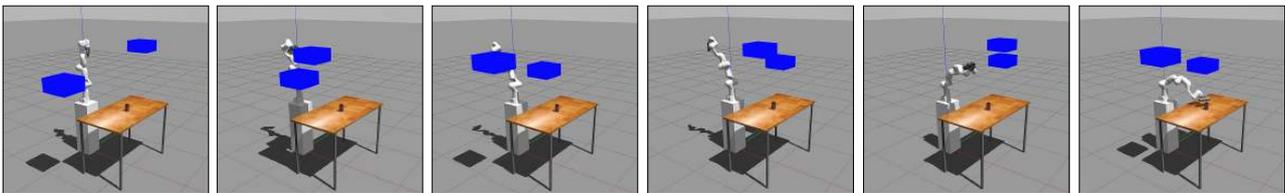


Fig.6 motion planning by 7-axis robot arm.

- [7] S. Karaman, and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning", *RSS*, vol. 104, no. 2, 2010.
- [8] J. J. Kuffner, and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning", *ICRA*, vol. 2, 2000, pp. 995–1001.
- [9] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt\*: Optimal incremental path planning focused through an admissible ellipsoidal heuristic", *IROS*, vol. 2, no. 1, 2014, pp. 3–1.
- [10] N. Chao, Y. K. Liu, H. Xia, M. J. Peng, and A. Ayodeji, "DI-rrt\* algorithm for least dose path re-planning in dynamic radioactive environments", *NET*, vol. 51, no. 3, pp. 825–836, 2019.
- [11] N. Garca, R. Surez, and J. Rosell, "Hg-rrt\*: Human-guided optimal random trees for motion planning", *ETFA*, 2015, pp. 1–7.
- [12] M. Zucker, N. Ratliff, A. Dragan, M. Pivtoraiko, M. Klingensmith, C. Dellin, J. A. D. Bagnell, and S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning", *Int. J. Rob. Res.*, vol. 32, no. 9, pp. 1164–1193, 2013.
- [13] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning", *Int. J. Rob. Res.*, pp. 4569–4574, 2011.
- [14] A. H. Qureshi, M. J. Bency, and M. C. Yip, "Motion planning networks", *ICRA*, 2019, pp. 2118–2124.
- [15] H. Ha, J. Xu, and S. Song, "Learning a decentralized multi-arm motion planner", *CoRL*, 2020.
- [16] P. Leven, and S. Hutchinson, "A framework for real-time path planning in changing environments." *IJRR*, vol. 21, no. 12, pp. 999–1030, 2002.
- [17] T. Kunz, U. Reiser, M. Stilman, and A. Verl, "Real-time path planning for a robot arm in changing environments", *IROS*, 2010, pp. 5906–5911.
- [18] J. S. Park, C. Park, and D. Manocha, "I-planner: Intention-aware motion planning using learning-based human motion prediction", *IJRR*, vol. 38, no. 1, pp. 23–39, 2019.
- [19] N. Koenig, and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator", *IROS*, vol. 3, 2004, pp. 2149–2154.