Robust Trajectory Execution for Multi-Robot Teams Using Distributed Real-time Replanning

Baskın Şenbaşlar Wolfgang Hönig Nora Ayanian

I. INTRODUCTION

Motion planning for multi-robot systems is particularly important in cases where many robots must interact with each other in confined spaces, potentially with many obstacles. Modern planning algorithms can find trajectories that effectively coordinate hundreds of robots while approximately optimizing objectives such as total energy used [2]; however, all such solutions assume that the resulting trajectories can be executed nearly perfectly, which is an unrealistic assumption.

To compensate for changes in the environment or imperfect execution, one might apply cooperative collision avoidance strategies, such as ORCA [3], at runtime. However, such algorithms often operate locally and do not take the pre-planned trajectories into account.

We propose an algorithm for robust trajectory execution that compensates for a variety of dynamic changes, including newly appearing obstacles, robots breaking down, imperfect motion execution, and external disturbances. Robots do not communicate with each other and only sense other robots' positions and the obstacles around them.

II. APPROACH

Consider a group of m robots. Each robot i is given the following:

$\mathbf{o}_i(t)$: original trajectory of i^{th} robot where $t \in [0, T_i]$,
c : order of derivative up to which smoothness is required,
$R(\mathbf{p})$: convex collision shape of any robot at position $\mathbf{p}$,
γ_k : dynamic limit of the robot for the k^{th} derivative of its trajectory.

Each robot i can sense the positions $\{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ of other robots as well as the current occupied space $\mathcal{O}_i$ around it. Robots are unaware of the other robots' planned trajectories, and cannot communicate with each other. Each robot i must execute a trajectory $\mathbf{f}_i(t)$, where $\mathbf{f}_i(t)$ is a solution to the following optimization problem:

$$\text{minimize} \int_0^{T_i} \|\mathbf{f}_i(t) - \mathbf{o}_i(t)\|^2 \, dt \tag{1}$$

subject to

$$\begin{aligned}
&\mathbf{f}_i(t) \text{ is continuous up to degree } c,\\
&\frac{d^j \mathbf{f}_i}{dt^j}(0) = \frac{d^j \mathbf{p}_i}{dt^j}(0) \text{ for } j \in \{0, 1, ..., c\}\\
&\mathbf{f}_i(t) \text{ is collision-free, and}\\
&\left\| \frac{d^k \mathbf{f}_i(t)}{dt^k} \right\| \leq \gamma_k \text{ for all desired } k,
\end{aligned}$$

where $t \in [0, T_i]$.

We solve this problem approximately, using a dynamic receding horizon approach iteratively. At every iteration K, robot i plans a trajectory $\mathbf{f}_i^K(t)$ that starts at the robots' current position and is safe to execute up to the user-provided period δt. Replanning is done at a fixed period of δt. In each iteration K, we sense the other robots' positions to compute the buffered Voronoi cell $\mathcal{V}_i$ [4], update our current representation of the occupied space ($\mathcal{O}_i$), and compute a trajectory $\mathbf{f}_i^K(t)$.

We execute the following three major components iteratively: *discrete planning* that is used to efficiently plan around new obstacles, *trajectory optimization* to generate smooth and collision-free trajectories, and *temporal rescaling* to enforce the dynamic limits of the robot.

At the end of each iteration, each robot has its trajectory $\mathbf{f}_i^K(t)$ that is guaranteed to be collision-free up to time δt; is continuous up to the c^{th} derivative; obeys the dynamic limits of the robot; tries to stay close to the original trajectory; and is a good starting point for the next iteration.

III. EVALUATION

We test our approach both in simulations and physical experiments using differential drive robots with varying external disturbances. We show that our algorithm i) scales well with the number of robots and the number of obstacles; ii) avoids collisions and deadlocks better than ORCA; and (iii) unlike ORCA, results in smooth trajectories.

This paper is an extended abstract of work accepted at DARS 2018 [1]. A supplemental video containing some of our simulations and physical experiments is available at https://youtu.be/LbWRvLfdwTA. This research was supported in part by Office of Naval Research grant N00014-14-1-073 and National Science Foundation grant 1724399. B. Şenbaşlar gratefully acknowledges the support from the Fulbright program sponsored by U.S. Department of State. All authors are with the Department of Computer Science, University of Southern California, Los Angeles, CA, USA. Email: {baskin.senbaslar, whoenig, ayanian}@usc.edu

REFERENCES

[1] B. Şenbaşlar, W. Hönig, and N. Ayanian, "Robust trajectory execution for multi-robot teams using distributed real-time replanning," in *DARS*, 2018, accepted. To appear.

[2] W. Hönig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE T-RO), Special Issue on Aerial Swarm Robotics*, 2018.

[3] J. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha, "Reciprocal *n*-body collision avoidance," in *ISRR*, 2009, pp. 3–19.

[4] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE RA-L*, vol. 2, no. 2, pp. 1047–1054, 2017.