Distributed planning of Multi-rotor drone fleets using the Smooth Robustness of Signal Temporal Logic

Yash Vardhan Pant, Houssam Abbas, Rahul Mangharam
{yashpant, rahulm}@seas.upenn.edu, housssam.abbas@oregonstate.edu

ABSTRACT

In this work, we present a distributed method for planning of multi-rotor drones with Signal Temporal Logic (STL) specifications. Starting with a smooth (infinitely differentiable) approximation of the STL robustness function, we maximize it using a distributed optimization method to generate trajectories that robustly satisfy the STL specification. We also compare the approach with a centralized non-convex optimization based method to show its viability.

1. INTRODUCTION

Uncertainties and errors in Cyber-Physical Systems (CPS) can affect both the cyber components (e.g., software bugs, imperfect information from perception-based algorithms) and physical components of a system (e.g., sensor failures, imperfect actuation). To deal with *unforeseen* problems, the system must be controlled at runtime such that it not only satisfies its design specifications, but it satisfies them robustly. Since these problems are, by definition, unforeseen, unmodeled and only detected by their effect on the output, the notion of robustness must be computable using only the output behavior of the system. The *robustness* of Metric Temporal Logic (MTL), or with some differences Signal Temporal Logic (STL), specifications [2, 1] is a rigorous notion that has been used successfully for the testing of automotive systems, medical devices, and general CPS. In details, STL is a formal language for expressing complex reactive requirements with time constraints. Given a STL specification φ and a system execution $\mathbf{x}$, the robustness $\rho_\varphi(\mathbf{x})$ of the spec relative to $\mathbf{x}$ measures two things: its sign tells whether $\mathbf{x}$ satisfies the spec ($\rho_\varphi(\mathbf{x})>0$) or violates it ($\rho_\varphi(\mathbf{x})<0$). Its magnitude $|\rho_\varphi(\mathbf{x})|$ measures how *robustly* the spec is satisfied or falsified, i.e. any perturbation to $\mathbf{x}$ of size less than $|\rho_\varphi(\mathbf{x})|$ will not cause its truth value to change. Thus, a planning algorithm can *maximize* the robustness over all possible trajectories to determine the trajectory that a robot (or robots) need to follow.

Unfortunately, the robustness function ρ_φ is hard to work with. In particular, it is non-differentiable, so we have to resort to heuristics or costly non-smooth optimizers. This makes its optimization a challenge - indeed, most existing approaches treat it as a black box and apply heuristics to its optimization. In previous work [5], we introduced smooth (infinitely differentiable) approximations to the robustness function of arbitrary MTL formulae, which can be made arbitrarily close to the true robustness. This smooth approximation can also be applied to the robustness of STL formulae in a straightforward manner [6]. It was shown shown in [5, 6] that maximizing this smooth robustness (using powerful off-the-shelf gradient based optimizers) for control and planning outperforms heuristics and Mixed-Integer Linear Program (MILP) based approaches [7]. However, all the methods used so far rely on solving the robustness maximization problem in a centralized manner, i.e. where a single computation resource generates the plans for each robot. In many realistic operating scenarios, especially with communication constraints, this is not a feasible approach. [4] presents a method for decentralized control of agents with STL specificationsbut its limitations restrict it from use in cases where two agents need to be separated by some minimum distance as in the multi-agent mission we consider here (Sec. 2). **In this work, we present a method to maximize the smooth robustness of STL specifications in a distributed manner, with application to multi-rotor drone planning**. We adapt the non-convex optimization algorithm of [8], and use it with the formulation of [6] to generate trajectories for multi-rotor drones.

Outline of approach: For brevity, we only present a brief outline of the approach. The formulation for the optimization to be solved can be found in [6]. We start by dividing the fleet of drones into groups. At each iteration of the algorithm, we select a particular group, where the gradient of smooth robustness w.r.t the variables of the drones in that group is used to solve a local co-ordinate descent like convex problem (see [8] for more details) separately (possibly in parallel) for each drone in the group. For simplicity we assume a single drone does this optimization for the entire group. Following this, the solution is communicated to each group in the fleet. At the next iteration, we again select a group (could be same group as before) to repeat the optimizations with the updated solution, and this process goes on till the solution converges. For the simulations here, we consider a cyclic schedule where one group solves for C consecutive iterations of the algorithm, followed by other groups doing the same in a predefined manner. Initial results on a simple multi-drone mission show that this method's performance is comparable to the centralized approach [6]. This suggests, that with additional effort, this distributed method can be used for Urban Air-Traffic according to the requirements outlined by the Federal Aviation Administration [3].

Yash Vardhan Pant and Rahul Mangharam are with the Dept. of Electrical and Systems Engineering, University of Pennsylvania. Houssam Abbas is with the School of Electrical and Computer Engineering, Oregon State University.

2. MULTI DRONE REACH-AVOID MISSION

The objective of the drone d is to reach a goal set Goal

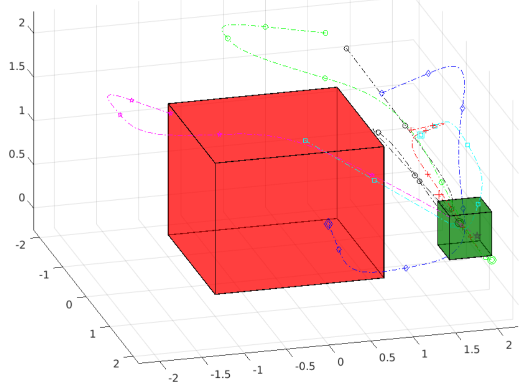

Figure 1: The reach-avoid mission workspace. The large set in red is the unsafe set, while the smaller set in green is the goal set that has to be visited by all drones within 6 seconds. Also shown are the trajectories generated by our method for D=6 drones. The upper bound on robustness for this workspace is 0.25. An example of such a mission (from [6]) being flown can be seen at: https://bit.ly/varvel8

Table 1: Mean robustness ($\rho_{\varphi_{mra}}$) and computation times taken with increasing number of drones (D) for the distributed method (in bold) presented here and the centralized method [6].

D	Run-time (s)	$\rho_{\varphi_{mra}}$
2	**0.083**/0.86	**0.19**/0.18
4	**0.68**/1.83	**0.19**/0.15
6	**10.49**/9.08	**0.09**/0.10
8	**14.58**/15.86	**0.11**/0.07

($[1.5,2] \times [1.5,2] \times [0.5,1]$) within the time interval $[0,T]$, while avoiding an unsafe set Unsafe ($[-1,1] \times [-1,1] \times [0,1]$) throughout the interval, in the 3D position space. The environment is shown in Fig. 1. With p denoting the drone's 3D position, the mission for a single drone d is:

$$\varphi^d_{\text{sra}} = \Box_{[0,T]} \neg (p \in \text{Unsafe}) \wedge \Diamond_{[0,T]} (p \in \text{Goal}) \tag{1}$$

The Multi drone Reach-Avoid problem adds the requirement that every two drones d, d' must maintain a minimum separation $\delta_{\min} > 0$ from each other: $\varphi^{d,d'}_{\text{sep}} = \Box_{[0,T]} (||p^d - p^{d'}|| \geq \delta_{\min}$. Over D drones, the specification is:

$$\varphi_{\text{mra}} = \wedge_{d=1}^{D} \varphi^d_{\text{sra}} \bigwedge \wedge_{d=1}^{D} (\wedge_{d' \neq d} \varphi^{d,d'}_{\text{sep}}) \tag{2}$$

The horizon of this formula is $hrz(\varphi_{\text{mra}}) = T = 6s$. We analyze the runtimes and achieved robustness of the proposed distributed method and the centralized one of [6] by running a 10 simulations, each from randomly-chosen initial positions of the drones in the free space $X/(\text{Goal} \cup \text{Unsafe})$).

Simulation results: Fig. 2 shows how the robustness increases as the number of iterations of the algorithm increase for D=8 drones divided into 2 groups, and the varying performance as the number of consecutive iterations given to a particular group (in the cyclic schedule outlined previously) to are varied. Note that not all such schedules result in trajectories that satisfy (i.e. $\rho_{\varphi_{mra}} > 0$) the specification of (2). Table 1 shows that the distributed method outlined (for its best case) here performs similar to the centralized version.

3. CONCLUSION

We briefly outline a method for distributed planning of multi-rotor fleets by maximizing the smooth robustness of

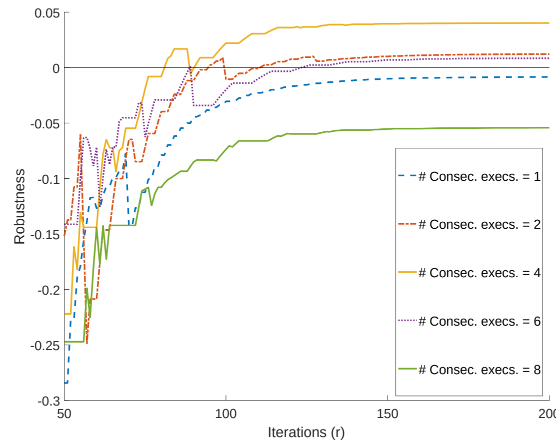

Figure 2: Robustness of the trajectories versus the number of iterations of the algorithm for different cyclic schedules for one instance of the multi drone reach-avoid mission.

a given STL specification in a distributed method. Initial simulation results show that the method performs well. On going work is focused on a systematic method for scheduling in the algorithm (see outline of approach), tuning the hyper-parameters of the algorithm and on taking communication requirements into consideration to group drones together in order for the method to scale to real world scenarios.

4. REFERENCES

[1] A. Donzé and O. Maler. *Robust Satisfaction of Temporal Logic over Real-Valued Signals*. Springer Berlin Heidelberg, 2010.

[2] G.E. Fainekos, A. Girard, and G. Pappas. *Temporal Logic Verification Using Simulation*, pages 171–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[3] Federal Aviation Administration. Integration of civil unmanned aircraft systems (uas) in the national airspace system (nas) roadmap. https://www.faa.gov/uas/media/UAS_Roadmap_2013.pdf, 2013. Retrieved May 20, 2018.

[4] Lars Lindemann and Dimos V. Dimagogonas. Decentralized robust control of coupled multi-agent systemsunder local signal temporal logic tasks. In *American Control Conference (ACC)*, 2018.

[5] Yash Vardhan Pant, Houssam Abbas, and Rahul Mangharam. Smooth operator: Control using the smooth robustness of temporal logic. In *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, 2017.

[6] Yash Vardhan Pant, Houssam Abbas, Rhudii A Quaye, and Rahul Mangharam. Fly-by-logic: control of multi-drone fleets with temporal logic objectives. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, 2018.

[7] V. Raman, A. Donze, M. Maasoumy, R. M. Murray, A. Sangiovanni-Vincentelli, and S. A. Seshia. Model predictive control with signal temporal logic specifications. In *53rd IEEE Conference on Decision and Control*, pages 81–87, Dec 2014.

[8] Meisam Razaviyayn, Mingyi Hong, Zhi-Quan Luo, and Jong-Shi Pang. Parallel successive convex approximation for nonsmooth nonconvex optimization. In *Advances in Neural Information Processing Systems*, 2014.