



HAL
open science

Trajectory planning for multicopters connectivity maintenance through distributed optimization

Vincent Marguet, Vittorio Casagrande, Francesca Boem, Ionela Prodan

► **To cite this version:**

Vincent Marguet, Vittorio Casagrande, Francesca Boem, Ionela Prodan. Trajectory planning for multicopters connectivity maintenance through distributed optimization. SAGIP MARSEILLE 2023, Société d'Automatique, de Génie Industriel et de Productique, Jun 2023, Marseille, France. hal-04136478

HAL Id: hal-04136478

<https://hal.science/hal-04136478>

Submitted on 21 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Trajectory planning for multicopters connectivity maintenance through distributed optimization *

Vincent Marguet¹, Vittorio Casagrande², Francesca Boem², Ionela Prodan¹

¹ Université Grenoble Alpes, Grenoble INP [‡], LCIS, F-26000, Valence, France
{vincent.marguet, ionela.prodan}@lcis.grenoble-inp.fr

² Dept. of Electronic and Electrical Engineering, University College London, UK
{vittorio.casagrande.19, f.boem}@ucl.ac.uk

Index Terms : *Team of Multicopters, Trajectory planning, Connectivity maintenance, Distributed Model Predictive Control*

1 Introduction

Swarms of Unmanned Aerial Vehicles (UAVs) are nowadays used in various domains, e.g. precision agriculture, military and photography [1]. In some precision agriculture applications, the drones must cooperate and maintain connectivity to collect data from a group of ground sensors. Hence, the drones trajectories must be planned such that objectives like trajectory length or energy minimization together with connectivity must be satisfied. A popular approach in the literature is to first generate the trajectories offline and then follow these trajectories online using Model Predictive Control. In large networks, we may not want to share all the data among all the agents, in particular the ones that are far, a distributed system appears to be the best solution.

In this work, the multiple agents trajectory optimization problem is formulated via MPC (Model Predictive Control) and solved by a distributed algorithm using the Python package DISROPT [4]. In this framework, at each time step, each agent of the network formulates a local optimization problem which is interconnected to the others through coupling constraints which describe the connectivity maintenance. Preliminary simulation and experimental results show promise for the proposed approach.

2 Preliminaries

Let us provide in the following, the general expression of the constrained-coupled optimization problem solved at each step :

$$\min_{\{\mathbf{x}_1, \dots, \mathbf{x}_N\}} \sum_{i=1}^N f_i(\mathbf{x}_i) \text{ s.t. } \mathbf{x}_i \in \mathbf{X}_i \text{ and } \sum_{i=1}^N g_i(\mathbf{x}_i) \leq 0 \quad (1)$$

in which \mathbf{x}_i , f_i , \mathbf{X}_i and g_i are respectively the local decision variable, the local objective function, the local constraint set and the coupled function of component i . The distributed dual subgradient algorithm presented in [3] is used to solve this optimization problem without communicating the estimate of the decision variable \mathbf{x}_i .

Furthermore, we consider a linear time-invariant model for the multicopter systems. See also [2] where a nonlinear feedback controller linearizes the multicopter model in closed loop, described in fact by a double integrator on each axis x,y,z. Each agent i has its state space representation :

$$\begin{aligned} \mathbf{x}_i[k+1] &= \mathbf{A}_i \mathbf{x}_i[k] + \mathbf{B}_i \mathbf{u}_i[k] \\ \mathbf{y}_i[k] &= \mathbf{C}_i \mathbf{x}_i[k] \end{aligned} \quad (2)$$

*This work is supported by the French National Research Agency in the framework of the "France 2030" program (ANR-15-IDEX-0002) and by the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01).

[‡]Institute of Engineering and Management Univ. Grenoble Alpes.

with $\mathbf{x}_i[k] = [\mathbf{p}_i[k]^T \mathbf{v}_i[k]^T]^T \in \mathbb{R}^6$ the state vector representing the positions and velocities on each axis, $\mathbf{u}_i[k] \in \mathbb{R}^3$ the input vector representing the acceleration on each axis and $\mathbf{A}_i \in \mathbb{R}^{6 \times 6}$, $\mathbf{B}_i \in \mathbb{R}^{6 \times 3}$, $\mathbf{C}_i \in \mathbb{R}^{3 \times 6}$ matrices corresponding to the double integrator dynamics.

3 Distributed optimization problem formulation for UAV trajectory generation

We detail here the optimization procedure followed in parallel by every agent of the network. At each step j , each agent i has to minimize a cost function corresponding to the sum of the minimal displacements over a prediction horizon N_{pred} corresponding to a number of steps, in other words the length of the planned trajectory over the prediction horizon is minimized :

$$\min_{\{\mathbf{u}_i[0], \dots, \mathbf{u}_i[N_{pred}-1]\}} \sum_{k=1}^{N_{pred}} \|\mathbf{y}_i[k] - \mathbf{y}_i[k-1]\| \quad (3)$$

with $\mathbf{y}_i[k]$ the position of agent i at time step k on axis x,y,z. Furthermore, while minimizing the objective function (3) the following constraints are considered :

$$\left\{ \begin{array}{l} \text{dynamical model (2)} \\ \text{current positions and velocities at initial step} \\ \text{desired positions and velocities at step } N_{step} \\ \text{velocity } \mathbf{v}_i[k+1] \in \left[\frac{-v_{max}}{\sqrt{3}}, \frac{v_{max}}{\sqrt{3}} \right]^3 \text{ and acceleration } \mathbf{u}_i[k] \in \left[\frac{-a_{max}}{\sqrt{3}}, \frac{a_{max}}{\sqrt{3}} \right]^3 \text{ constraints} \\ \text{position constraints } \mathbf{p}_i[k+1] \in [x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}] \\ \text{the distance between 2 agents has to be shorter than the communication range} \\ \text{communication constraints for each agent } \mathbf{A}_c(\mathbf{p}_i - \mathbf{p}_{i+1}) \leq \mathbf{b}_c. \end{array} \right. \quad (4)$$

The communication constraints are coupling constraints written as in (1) $\sum_{i=1}^N g_i(\mathbf{x}_i) \leq 0$

which is equivalent with $\sum_{i=1}^N \mathbf{G}_i \mathbf{p}_i - \mathbf{h}_i \leq 0$. Furthermore, using the half-space representation of a polytopic region we have that $\mathbf{A}_c(\mathbf{p}_i - \mathbf{p}_{i+1}) \leq \mathbf{b}_c$ to approximate (by a polytope) the communication range sphere constraining the relative positions of the agents. We apply this constraint for the first next predicted position of each agent to ensure that the communication is kept. The procedure followed by each agent is explained in Algorithm 1.

Algorithm 1: Procedure followed by each agent i

Input: Initial and target positions and velocities of agent i ;

Parameters $N, N_{pred}, N_{steps}, v_{max}, a_{max}, x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max}, \mathbf{A}_c, \mathbf{b}_c$;

Output: Feasible trajectory of the agent i ;

```

1  $t \leftarrow 0$ ;
2 while  $t < N_{steps}$  do
3   Build local constraints, coupling constraint and objective function over  $N_{pred}$ ;
4   Solve optimization problem using Distributed Dual Subgradient algorithm;
5   Apply first optimal input on each axis  $\mathbf{u}_i^*[0]$ ;
6   Upload position and velocity of all the agents;
7    $t \leftarrow t + 1$ ;
8 end

```

4 Simulation results

The numerical values are given by the experimental platform available (the operation space available of the nano-drones). Hence, we have that $x_{min} = -1.1m$, $x_{max} = 1.1m$, $y_{min} = -1.1m$, $y_{max} = 1.1m$, $z_{min} = 0m$, $z_{max} = 2.0m$ for the axis limits, we set $v_{max} = 1m/s$, $a_{max} = 5m/s^2$, a communication range $\rho = 0.7m$, a number of 20 steps to reach the next waypoint, a sampling time of 200 ms and a prediction horizon of 20 steps.

This trajectories planning optimisation problem is solved using DISROPT, a Python package for distributed optimization over peer-to-peer networks [4]. It is usually used to define and solve convex distributed optimization problems with linear constraints and dynamics.

Once the agents reach a waypoint, they repeat the procedure until the next waypoint. The results of the simulation are represented in Figure 1. The initial and desired positions of the 3 UAVs are represented with red dots. The communication sphere of agent 2 is intersecting both spheres of agents 1 and 3 so the communication is kept. We verified also this constraint on the whole simulation by plotting the inter-UAV distance. The simulation time taking into account 3 agents is 1218 seconds, for only 2 agents it is reduced to 679 seconds.

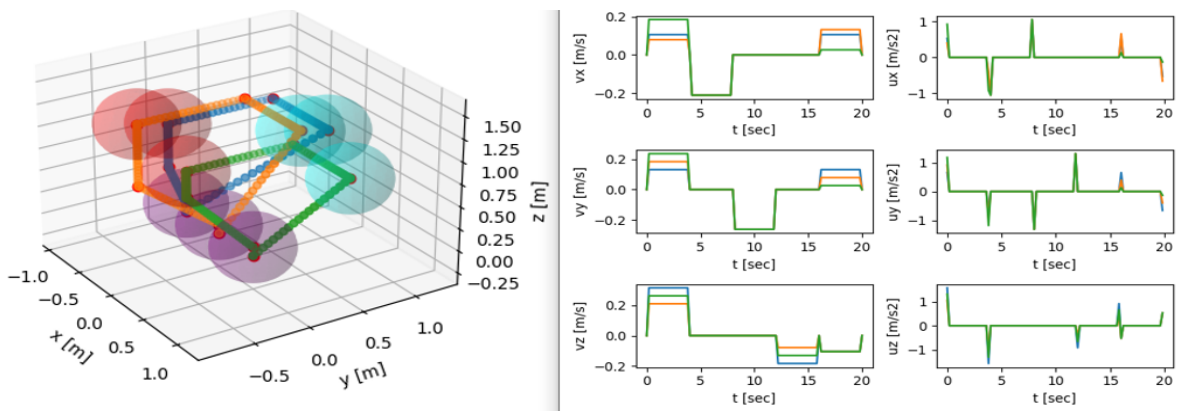


FIG. 1 – (left) Obtained positions per step of the 3 agents reaching 5 waypoints each, the spheres represented at 2 steps of the simulation have a radius of $\frac{\rho}{2}$ for clarity : the agents communicates if there sphere intersects, (right) velocities and inputs on each axis of each agent

5 Conclusions

This paper proposed a distributed Model Predictive Control algorithm for trajectory planning of multiple aerial drones while ensuring communication constraints. The contribution resides in the use of distributed optimization for solving off-line the MPC problem. The trajectories generated are then followed on-line by real UAVs (Crazyflie) using the Qualisys Track Manager.

Références

- [1] KW Chan, U Nirmal, and WG Cheaw. Progress on drone technology and their applications : A comprehensive review. In *AIP Conference Proceedings*, volume 2030, page 020308. AIP Publishing LLC, 2018.
- [2] Huu-Thanh Do, Ionela Prodan, and Florin Stoican. Analysis of alternative flat representations of a uav for trajectory generation and tracking. In *2021 25th International Conference on System Theory, Control and Computing (ICSTCC)*, pages 58–63. IEEE, 2021.
- [3] Alessandro Falsone, Kostas Margellos, Simone Garatti, and Maria Prandini. Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica*, 84 :149–158, 2017.
- [4] Francesco Farina, Andrea Camisa, Andrea Testa, Ivano Notarnicola, and Giuseppe Notarstefano. Disropt : a python framework for distributed optimization. *IFAC-PapersOnLine*, 53(2) :2666–2671, 2020.