A Distributed Architecture for Onboard Tightly-Coupled Estimation and Predictive Control of Micro Aerial Vehicle Formations

I. Kagan Erunsal[1,2], Rodrigo Ventura[2] and Alcherio Martinoli[1]

[1] Distributed Intelligent Systems and Algorithms Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
kagan.erunsal@epfl.ch, alcherio.martinoli@epfl.ch
[2] Institute for Systems and Robotics, Instituto Superior Técnico, Lisbon, Portugal
rodrigo.ventura@isr.tecnico.ulisboa.pt

Abstract. This paper presents a distributed estimation and control architecture for leader-follower formations of multi-rotor micro aerial vehicles. The architecture involves multi-rate extended Kalman filtering and nonlinear model predictive control in order to optimize the system performance while satisfying various physical constraints of the vehicles, such as actuation limits, safety thresholds and perceptual restrictions. The architecture leverages exclusively onboard sensing, computation, and communication resources and it has been designed for enhanced robustness to perturbations thanks to its tightly coupled components. The architecture has been initially tested and calibrated in a high-fidelity robotic simulator and then validated with a real two-vehicle system engaged in formation navigation and reconfiguration tasks. The results not only show the high formation performance of the architecture while satisfying numerous constraints but also indicate that it is possible to achieve full navigation and coordination autonomy in presence of severe resource constraints as those characterizing micro aerial vehicles.

Keywords: Formation control, micro aerial vehicles, distributed nonlinear model predictive control, relative and onboard localization, distributed estimation

1 Introduction

Recent years have been characterized by various improvements in the coordination and cooperation strategies of autonomous robots. This is because loosely connected systems controlled by decentralized or distributed strategies have inherent advantages over centrally governed systems such as additional flexibility, robustness, and scalability [1]. Among those strategies, formation control, usually aimed at achieving prescribed inter-vehicle geometric constraints, is one of the most actively investigated topics [2]. One promising method to optimally accomplish this goal while satisfying diverse constraints is the Nonlinear Model Predictive Control (NMPC) due to its architectural heterogeneity, simple reconfigurability, and hierarchical flexibility [3]. Furthermore, to improve the performance, the distributed version of this algorithm is a prominent candidate

because it approximates the global objective more accurately than its decentralized counterpart by including inter-robot communication.

However, NMPC-based algorithms require an accurate model of the system, correct state estimates, and sufficient computational resources. Furthermore, the pure distributed version of the algorithm requires a robust communication framework. These requirements might be challenging to satisfy primarily on resource-constrained Micro Aerial Vehicles (MAVs). For example, system identification techniques might not yield a faithful model, the system might be time-varying due to the intrinsic nature of multiple vehicles, and only the partial state information and limited computational power might be available because of hardware and software constraints. As a result, these problems should be addressed in a tightly-coupled estimation and control architecture that considers the various constraints in order to be deployed successfully on real robots. Another critical challenge for deploying such systems is the dependency of the robotic platforms on global localization systems, such as Motion Capture Systems (MCSs) or Global Navigation Satellite Systems (GNSSs), for indoor and outdoor scenarios, respectively.

There are various successful real-world implementations of model predictive approaches for the formations of multi-robot systems. Among those, we focus on decentralized and distributed approaches with real-world implementations. In [4], authors propose a trajectory planner that considers various task requirements such as swarm coordination, flight efficiency, obstacle avoidance for a swarm of micro flying robots and demonstrates the method in cluttered environments. Although the strategy seems highly effective and scalable for flocking navigation, visual-inertial odometry (VIO) coupled with Ultra-Wide Band (UWB) distance measurement (without orientation information) might not be accurate enough for formation control applications. In [5], authors present a decentralized strategy for the compact swarm of unmanned aerial vehicles without communication and external localization. The approach is well-constructed for outdoor flocking applications; however, the accuracy of the ultra-violet (UV) direction and ranging technique is limited for indoor formation applications. Furthermore, since the predictive information is not shared among the vehicles, it might yield low performance for dynamic scenarios. A distributed methodology is studied for formation control and obstacle avoidance of homogeneous robots by employing multi-robot consensus in [6]. The method is scalable and efficient with the number of robots; however, the implementation is not carried out onboard, and a motion capture system is used to obtain state feedback. In [7], a distributed predictive approach for the swarms of drones in cluttered environments and under high sensor noise is presented. Although the real-world experiments with sixteen quadrotors show the scalability of the strategy, the computations are not carried out onboard, and the localization solution is global. In [8], authors introduce a decentralized robotic front-end for autonomous aerial motion-capture in an outdoor environment and integrate perception constraints into the optimization problem. Although the computations are performed onboard, GNSS information is collected and shared among vehicles to generate relative position information, which is not always reliable. Other works with real-world implementations include dynamic encirclement [9], outdoor predictive flocking [10], formation in motion [11], formation with communication constraints [12] and indoor forma-

tion with visual-inertial odometry [14]. However, the works listed previously relies on global localization systems and/or offboard computation and/or shared coordinate frames and/or reliable communication network.

This paper presents a distributed estimation and control architecture based on a multi-rate Extended Kalman Filter (EKF) and NMPC to perform leader-follower formations of multi-rotor MAVs under state, actuation, and perception constraints and in the presence of environmental disturbances and model uncertainties. Our focus here is mainly on the follower subsystem since it constitutes the central part of the leader-follower formations. We validate the performance and applicability of this architecture, leveraging exclusively onboard resources and relative sensing in a high-fidelity robotic simulator and a real-world setup with diverse experiments. In particular, the followers obtain the relative pose and body-fixed velocity information by employing their coupled optical flow and distance sensor, and an RGB camera tracking April bundles [15].

The main contribution of this paper is to meet the following requirements simultaneously. First, the formation subsystems exclusively employ onboard resources for localization, computation, communication, and sensing. This brings the multi-robot system closer to practical applications. Second, the overall architecture components are designed to be scalable, performant, and robust on the network level. The main ingredient that allows such a characteristic is the tight-coupling of perception, communication, estimation, and control at the individual robot level. This unified architecture leads each component not only to exploit its full potential but also to assist the other components in maintaining a robust operation. In particular, the distributed NMPC exhibits high performance with an accurate model, state estimates, information from neighboring vehicles, and enough computational resources. For this purpose, the computationally efficient distributed EKF on each vehicle employs both the information received over communication and perception assets to provide uninterrupted state and uncertainty estimates, even if the communication channel has flaws. On the other hand, the individual EKFs require sufficient perception and communication update rates to perform successfully. This is where the distributed NMPC helps the perception and communication components to satisfy their various constraints, such as Field Of View (FOV) and range. Furthermore, the simple but efficient prediction model selected for NMPC not only minimizes the communication requirements but also maximizes the computational efficiency. Finally, the distributed architecture can be scaled up linearly by increasing the computational power up to a maximum number of neighbors.

2 Problem Formulation

Similar to [16], the following notation for vectors and rotation matrices will be adopted assuming that $\{a\}, \{b\}, \{c\}$ are the coordinate frames and x is the name of the vector:

$\boldsymbol{x}^c_{a/b}$: Vector quantity $x \in \mathbb{R}^n$ of the origin of $\{a\}$ with respect to the origin of $\{b\}$ expressed in $\{c\}$.

$\boldsymbol{R}^b_a$: Rotation matrix $R \in \mathbb{SO}^3$ that transforms a vector expression from $\{a\}$ to $\{b\}$ with Z-Y-X convention.

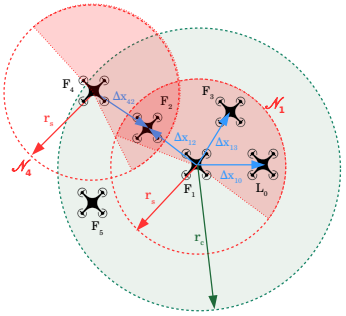

Fig. 1: An example of formation network consisting of one leader (L_0) and five followers (F_i, $i = 1,..,5$): two sub-networks for the F_1 and F_4, the corresponding relative positions, sensing and communication ranges are also indicated.

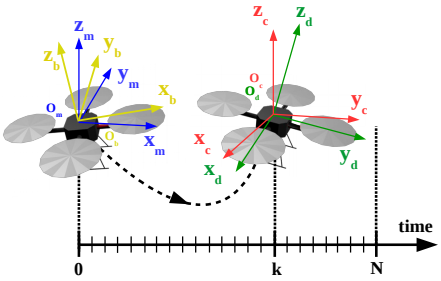

Fig. 2: MPC coordinate frames in a prediction horizon of length N: $\{m\}$ (blue), $\{b\}$ (yellow), $\{c\}$ (red), $\{c\}$ (green).

$\| \, . \, \|$ denotes the Euclidean norm of a vector or the spectral norm of a real matrix and $\| \, . \, \|_P := \sqrt{x^T P x}$ (with $P \in \mathbb{R}^{n \times n}$ and $P \succ 0$) stands for the weighted Euclidean norm of $x \in \mathbb{R}^n$.

Consider a formation network $\mathcal{N}$ which consists of $K+1$ aerial robots. Assume that one of them is assigned as a leader L and the rest are followers F_i so that $\mathcal{N} = \{L, F_1, ..., F_K\}$. All robots are operating in a workspace $\mathcal{W} \subset \mathbb{R}^3$. The state of each robots $i \in \mathcal{N}$ is defined as follows:

$$\boldsymbol{\chi}_i = [\boldsymbol{x}_{b/n}^{n\ T} \ \boldsymbol{v}_{b/n}^{n\ T} \ \boldsymbol{t}_{b/n}^{n\ T} \ \boldsymbol{w}_{b/n}^{b\ T}]^T \tag{1}$$

where $\boldsymbol{x}_{b/n}^n$ and $\boldsymbol{v}_{b/n}^n$ represent the position and linear velocity, $\boldsymbol{t}_{b/n}^n$ is the Euler angles, $\boldsymbol{w}_{b/n}^b$ denotes the angular velocities. Here, $\{n\}$ is the Earth-fixed inertial frame and $\{b\}$ is the body-fixed frame. Then, for instance, the vector $\boldsymbol{x}_{b/n}^n$ would represent the position of the origin of $\{b\}$ with respect to $\{n\}$ and expressed in the latter coordinate system.

We adopt the full dynamics of the multi-rotor MAV including dominant (propeller) and auxiliary (drag) aerodynamic effects based on [17] and [18]. Note that although this model can be used as a plant model for any simulation, it is too complex for the MPC purposes. We will explain the simplified model employed for MPC in the Section 3. Finally, due to the physical limits of the propellers and electric motors of the aerial vehicle, inputs should be inside a constraint set, i.e. $\boldsymbol{u}_{b/n}^b \in \mathcal{U}_c$. In addition, due to the safety limits of the vehicle, we have $\boldsymbol{\chi}_i \in X_c$.

Consider that a robot $i \in \mathcal{N}$ has a sensing area of $A_s \subset B_s(x_i, r_s)$ and communication area $A_c \subset B_c(x_i, r_c)$ where x_i is the center and r_s and r_c are the radius of norm-balls B_s and B_c, respectively. Assume $\Delta x_{ij} = x_j - x_i$ is the relative position of the robot $j \in \mathcal{N}$ with respect to the robot $i \in \mathcal{N} \setminus \{L\}$. Then the sub-network for the follower i is defined as follows: $\mathcal{N}_i := \{i, j \in \mathcal{N},\ j \neq i : \|\Delta x_{ij}\| < min(r_s, r_c)\}$. Fig. 1 represents such a network structure.

In this work, all robots including the leader have output type of feedback (i.e. no state-based feedback across the vehicles are possible). The only vehicle that has access to its absolute position by a global localization system is the

leader. As a result, the leader is responsible for the absolute navigation of the robot ensemble. All vehicles are equipped with a coupled optical flow - distance sensor to obtain linear velocities and ground distance, an Inertial Measurement Unit (IMU) and magnetometer to acquire linear accelerations, rotational velocities, and Euler angles. All follower vehicles are endowed with an onboard, limited range and FOV, 3D, relative localization system for measuring inter-vehicle poses of neighboring vehicles (i.e., relative range, bearing and orientation). We selected to use a RGB camera and April bundles to obtain such 3D relative pose information of the neighbor vehicles due to its accuracy and implementation simplicity. We do not assume any reliable communication network among vehicles but we benefit from it when it is available. Furthermore, all vehicles are under the disturbed effect of model uncertainty and slowly-varying airflow disturbance due to the wake effect of neighboring vehicles. Finally, all sensors are characterized by zero-mean Gaussian noise.

Problem 1. Consider one leader and K follower robots under the input, perception, communication and state constraints $\boldsymbol{u}^b_{b/n} \in \mathcal{U}_c$, $\boldsymbol{z}^b_{b/n} \in \mathcal{Z}_c$ and $\boldsymbol{\chi} \in X_c$. Here, the leader can be considered as an separate and independent trajectory tracking robot. Under the assumptions explained on the previous paragraph, design a control and estimation strategy for the followers so that the robotic network will follow a predefined reference trajectory $\boldsymbol{r}_p \in \mathbb{R}^3$ while maintaining formation structure and connectivity defined by, possibly time-varying, relative positions references $\Delta\widetilde{\boldsymbol{x}}_{ij}$, relative yaw angles references $\Delta\widetilde{\psi}_i$ where $i \in \mathcal{N}_i \setminus \{L\}$ and $j \in \mathcal{N}_i$.

3 Methodology

The components of our distributed architecture are selected and designed in order to take resource constraints into account. In particular, we have paid attention to distribute the architecture over the network while preserving its strong integration at the robot level. The former feature allows us to obtain a scalable behavior, while the latter one brings high performance and robustness due to the tight coupling of the perception, communication, estimation and control components.

The proposed architecture is represented in Fig. 3. Here, the neighborhood consists exclusively of followers, but the leader could also be part of the subnetwork. One can observe the tight-coupling of components on the vehicle level and their distribution over the network. The details of the estimation and formation control components and their relationship with the perception and communication will be explained in the next section.

In order to comprehend the details of estimation-control architecture, we should define the coordinate frames employed here. The first frame is the body-fixed reference frame $\{b\}$ and it is defined for the real-time horizon, the second one is the body-fixed MPC reference frame $\{d\}$ anchoring to the body throughout any MPC prediction horizon. As a result, $\{b\}$ and $\{d\}$ coincide at the beginning of each prediction horizon. In addition, $\{m\}$ is defined as the inertial MPC-EKF frame, which corresponds to the zero vector in the beginning of the motion. The final reference frame is introduced as MPC control frame $\{c\}$, which is a roll and

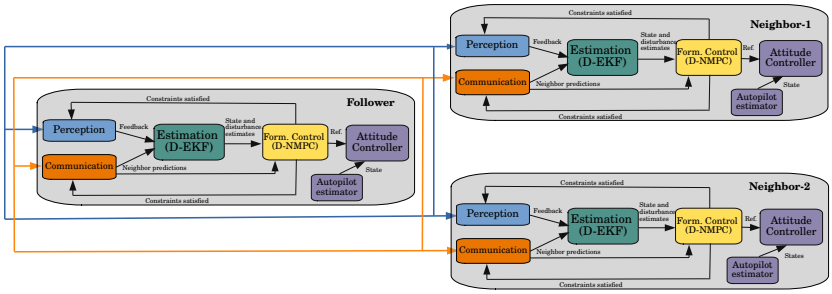

Fig. 3: Distributed architecture for a sub-network, tight coupling of components on the agent level is observable.

pitch free frame anchored to the $\{d\}$ and adopted in order to define control and reference variables. These frames are represented in Fig. 2.

3.1 Follower Estimation Scheme

We adopt a distributed multi-rate EKF approach for carrying out the estimation function due to its computational efficiency. The related literature is vast and we refer the reader to [19], [20] and the references there in. For the sake of clarity, we will present the process, and the perception models and highlight the important points.

Consider a new extended state definition for each robot $i \in \mathcal{N} \setminus \{L\}$:

$$\begin{aligned} \boldsymbol{\xi}_{ii} &= [\boldsymbol{v}_i^T \ \dot{\psi}_i \ \boldsymbol{d}_{i,a}^T]^T \\ \boldsymbol{\xi}_{ij} &= [\Delta\boldsymbol{x}_{ij}^T \ \boldsymbol{v}_j^T \ \Delta\psi_{ij} \ \dot{\psi}_j]^T \end{aligned} \tag{2}$$

where $\Delta\boldsymbol{x}_{ij}$, $\Delta\boldsymbol{v}_{ij}$ and $\Delta\psi_{ij}$ are the relative positions, velocities and yaw angles of the robot i with respect to $j \in \mathcal{N}_i$; $\boldsymbol{v}_i$, $\boldsymbol{v}_j$, $\dot{\psi}_i$ and $\dot{\psi}_j$ are the linear and yaw velocities of the robot i and $j \in \mathcal{N}_i$, respectively. Finally, $\boldsymbol{d}_{i,a}$ stands for the slowly varying disturbances on the acceleration dynamics of the agent i. Note that all states are expressed in the inertial frame $\{m\}$. Let us concatenate the states defined so that we obtain one extended state for an individual robot $i \in \mathcal{N}_i \setminus \{L\}$:

$$\boldsymbol{\xi}_i = [\boldsymbol{\xi}_{ii}^T \ \boldsymbol{\xi}_{ij_1}^T \ \boldsymbol{\xi}_{ij_2}^T \ ... \ \boldsymbol{\xi}_{ij_N}^T]^T \tag{3}$$

where $j_1, j_2, ..., j_N \in \mathcal{N}_i$.

Consider a generic nonlinear time-discrete model of an individual agent i:

$$\begin{aligned} \boldsymbol{\xi}_i[k+1] &= f_i(\boldsymbol{\xi}_i[k], \boldsymbol{u}_i[k], \boldsymbol{w}_i[k]) \\ \boldsymbol{z}_i[k] &= h_i(\boldsymbol{\xi}_i[k], \boldsymbol{\nu}_i[k]) \end{aligned} \tag{4}$$

where f_i is the process map, h_i is the perception map, $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}_w)$ and $\boldsymbol{\nu}_i \in \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma}_\nu)$ are the zero-mean Gaussian noise vectors on the process and perception model, respectively. Then, inspired by [21], for $\forall j \in N_i$, the process model can be written explicitly as follows:

$$\boldsymbol{v}_i[k+1] = \boldsymbol{v}_i[k] + \Delta t \left(\boldsymbol{R}_i[k] \frac{\boldsymbol{F}_i[k]}{m} - \boldsymbol{g} + \boldsymbol{d}_{i,a}[k] + \boldsymbol{w}_{i,a}[k] \right) \tag{5}$$

$$\psi_i[k+1] = \psi_i[k] + \Delta t\,(w_{i,\psi}[k]) \tag{6}$$
$$\dot{\psi}_i[k+1] = \dot{\psi}_i[k] + w_{i,\psi}[k] \tag{7}$$
$$\Delta \boldsymbol{x}_{ij}[k+1] = \Delta \boldsymbol{x}_{ij}[k] + \Delta t(\boldsymbol{v}_j[k] - \boldsymbol{v}_i[k]) - \tag{8}$$
$$\frac{\Delta t^2}{2}\left(\boldsymbol{R}_i[k]\frac{\boldsymbol{F}_i[k]}{m} - \boldsymbol{g} - \boldsymbol{d}_{i,a}[k] - \boldsymbol{w}_{ij,\Delta a}[k]\right) \tag{9}$$
$$\boldsymbol{v}_j[k+1] = \boldsymbol{v}_j[k] + \Delta t\,(\boldsymbol{w}_{j,a}[k]) \tag{10}$$
$$\Delta \psi_{ij}[k+1] = \Delta \psi_{ij}[k] + \Delta t(\Delta \dot{\psi}_{ij}[k] + \boldsymbol{w}_{ij,\Delta\psi}[k]) \tag{11}$$
$$\Delta \dot{\psi}_{ij}[k+1] = \Delta \dot{\psi}_{ij}[k] + \boldsymbol{w}_{ij,\Delta\psi}[k] \tag{12}$$
$$\boldsymbol{d}_{i,a}[k+1] = \boldsymbol{d}_{i,a}[k] + \boldsymbol{w}_{i,d}[k] \tag{13}$$

where $\boldsymbol{F}_i$ is the input, $\boldsymbol{R}_i$ stands for the rotation matrix, $\boldsymbol{g}$ is the gravitational acceleration vector, Δt is the sample time, $\boldsymbol{w}$'s represent the Gaussian noise, $\boldsymbol{d}$'s are the slowly varying disturbance (or uncertainty) on the corresponding states. Note that these lumped disturbance estimations accounting for model uncertainty, environmental disturbance and time-varying characteristics of the vehicle are crucial for the robust behavior of the predictive formation controller. In addition, some definitions in the model can be written as follows: $\boldsymbol{w}_{ij,\Delta a} := \boldsymbol{w}_{j,a} - \boldsymbol{w}_{i,a}$ and $\boldsymbol{w}_{ij,\Delta\psi} := \boldsymbol{w}_{j,\psi} - \boldsymbol{w}_{i,\psi}$. Note that all equations are written with respect to $\{m\}$ frame and $\forall j \in \mathcal{N}_i$. Furthermore, $\forall j \in \mathcal{N}_i$, the perception model is given as follows:

$$\boldsymbol{z}_{i,\Delta x}[k] = g(\boldsymbol{R}_i[k]\Delta \boldsymbol{x}_{ij}[k] + \boldsymbol{\nu}_{i,\Delta x}[k]) \tag{14}$$
$$\boldsymbol{z}_{i,v}[k] = \boldsymbol{R}_i[k]\boldsymbol{v}_i[k] + \boldsymbol{\nu}_{i,v}[k] \tag{15}$$
$$\boldsymbol{z}_{i,\Delta\psi}[k] = \Delta \boldsymbol{\psi}_{ij}[k] + \boldsymbol{\nu}_{i,\Delta\psi}[k] \tag{16}$$
$$\boldsymbol{z}_{i,\psi}[k] = \boldsymbol{\psi}_{ij}[k] + \boldsymbol{\nu}_{i,\psi}[k] \tag{17}$$
$$\boldsymbol{z}_{i,\dot{\psi}}[k] = \dot{\boldsymbol{\psi}}_{ij}[k] + \boldsymbol{\nu}_{i,\dot{\psi}}[k] \tag{18}$$
$$\boldsymbol{z}_{i,v_j}[k] = \boldsymbol{R}_j[k](\boldsymbol{v}_j[k]) + \boldsymbol{\nu}_{i,v_j}[k] \tag{19}$$

where $\boldsymbol{z}$ represents the measurement, $\boldsymbol{\nu}$ stands for the perception noise and function $g(.)$ maps the relative position to the range and bearing information. In addition, it is assumed that the robot has an accurate attitude estimator that generates the roll and pitch estimates $\phi_i[k]$ and $\theta_i[k]$ so that the rotation matrix $\boldsymbol{R}_i$ is fully defined.

Remark 1. (Neighboring vehicle states): Eq. (19) is critical in the perception model. $\boldsymbol{v}_j$ represents the velocity of any neighboring vehicle $j \in \mathcal{N}_i$ received by communication and it needs to be mapped to the vehicle i's reference frame by $\boldsymbol{R}_j$. However, if it is not available due to various communication flaws, the EKF provides a continuous estimation of the velocity of neighboring vehicles.

3.2 Follower Controller Scheme

Reliable and high performance attitude controllers are commonly available for today's aerial vehicles. In order to separate high frequency task (attitude control) from low frequency ones (formation control), to exploit the full potential

of the autopilot and onboard computer and to simplify the prediction model, we leverage a cascaded architecture as shown in Fig. 3. The attitude controller is selected as the PID-based controller proposed by [22] and the formation controller is a Distributed Nonlinear Model Predictive Controller (D-NMPC). The reason behind the latter choice is that NMPC can simultaneously handle constraints and optimize the performance in a systematic manner and the distributed version inherently has a higher performance since it approximates the centralized problem more accurately compared to its decentralized counterpart.

Here, we use the following notation to distinguish the predicted values from the actual ones: $\boldsymbol{v}[m|n]$ is the value of variable $\boldsymbol{v}$ at discrete instant $k = n$, predicted at $k = m$ where $m \geq$n. Furthermore, N is defined as the prediction (and control) horizon and Δt is the sampling time. Now, let's write the Open Loop Control problem (OCP) $\mathcal{P}(\hat{\boldsymbol{\xi}}[0], k)$ (at time k with initial estimated state $\hat{\boldsymbol{\xi}}[0]$) inspired by [23]:

$$\min_{\substack{\boldsymbol{\xi}_i[k|k],\ldots,\boldsymbol{\xi}_i[k+N|k] \\ \boldsymbol{u}_i[k|k],\ldots\boldsymbol{u}_i[k+N-1|k]}} \sum_{n=0}^{N-1} J_k(\boldsymbol{\xi}_i[k+n|k], \boldsymbol{u}_i[k+n|k], \boldsymbol{r}_i[k+n|k]) + J_N(\boldsymbol{\xi}_i[k+N|k], \boldsymbol{u}_i[k+N|k], \boldsymbol{r}_i[k+N|k]) \tag{20}$$

subject to the following constraints for n= 1,2, ..., N,

$$\boldsymbol{\xi}_i[k+n+1|k] = \hat{f}(\boldsymbol{\xi}_i[k+n|k], \boldsymbol{u}_i[k+n|k]) \tag{21}$$

$$g_n(\boldsymbol{\xi}_i[k+n|k], \boldsymbol{u}_i[k+n|k]) \leq 0 \tag{22}$$

$$h_n(\boldsymbol{\xi}_i[k+n|k], \boldsymbol{u}_i[k+n|k]) = 0 \tag{23}$$

$$\boldsymbol{\xi}_i[k|k] = \hat{\boldsymbol{\xi}}_i[0] \tag{24}$$

Here, J_k is the stage cost, J_N is the terminal cost, $\boldsymbol{\xi}_i$, $\boldsymbol{u}_i$ and $\boldsymbol{r}_i$ stand for state, input and reference for the vehicle i, respectively. Additionally, $\hat{f}$ is the prediction function and g_n and h_n represent inequality and equality constraints.

Before explaining the individual terms of the OCP, it is worth to visit the pinhole camera model [24] so that we can extract the coordinates of the detected vehicle on image plane and incorporate this information to OCP to obtain a perception-aware behavior. Let's consider that $\Delta\boldsymbol{x}_{ij}$ is a representation of the relative 3D coordinates of neighbor vehicle j and $\boldsymbol{p}_j$ is the image coordinates of the same vehicle assuming that the origin is the image center. Then we can obtain the following relationship,

$$\boldsymbol{p}_j = \boldsymbol{K}\boldsymbol{C}\Delta\boldsymbol{x}_{ij} \tag{25}$$

where $\boldsymbol{C}$ is the camera matrix, which is a function of focal length f of the camera and K is the gain.

Now, the stage and terminal cost functions can be expressed as follows:

$$J_k(\boldsymbol{\xi}_i[k+n|k], \boldsymbol{u}_i[k+n|k], \boldsymbol{r}_i[k+n|k]) = \sum_{j\in\mathcal{N}_i} \|\Delta\widetilde{\boldsymbol{x}}_{ij}[k+n|k] - \Delta\boldsymbol{x}_{ij}[k+n|k]\|_{Q_{\Delta x}} +$$

$$\sum_{j\in\mathcal{N}_i} \|\boldsymbol{v}_j[k+n|k] - \boldsymbol{v}_i[k+n|k]\|_{Q_v} + \|\boldsymbol{u}_i[k+n|k]\|_{Q_u} + \|\Delta\boldsymbol{u}_i[k+n|k]\|_{Q_{\Delta u}} + \sum_{j\in\mathcal{N}_i} \|\boldsymbol{p_j}\|_{Q_p} \tag{26}$$

$$J_N(\boldsymbol{\xi}_i[k+N|k], \boldsymbol{u}_i[k+N|k], \boldsymbol{r}_i[k+N|k]) = \sum_{j\in\mathcal{N}_i} \|\Delta\widetilde{\boldsymbol{x}}_{ij}[k+N|k] - \Delta\boldsymbol{x}_{ij}[k+N|k]\|_{Q_{\Delta x_N}} + \sum_{j\in\mathcal{N}_i} \|\boldsymbol{v}_j[k+N|k] - \boldsymbol{v}_i[k+N|k]\|_{Q_{v_N}} + \sum_{j\in\mathcal{N}_i} \|\boldsymbol{p_j}\|_{Q_{p_N}} \tag{27}$$

where $\Delta\widetilde{\boldsymbol{x}}_{ij}$ represents the geometric formation reference and $\Delta\boldsymbol{u}_i$ stands for the rate of change of inputs. As a result, in the stage cost function, the first term minimizes the formation errors, the second term stabilizes the relative velocities, the third and fourth terms optimizes the control effort and the last one is responsible for satisfying camera FOV constraints.

The continuous version of the prediction function, which includes not only ego but also the neighboring vehicle motion characteristics, $\hat{f}$ expressed in eq. (21) and can be discretized with any efficient method such as the implicit Runge Kutta (i-RG), can be written as follows:

$$\begin{aligned}
\dot{\Delta\boldsymbol{x}}^m_{d/m,ij} &= \boldsymbol{v}^m_{d/m,j} - \boldsymbol{v}^m_{d/m,i} \\
\dot{\boldsymbol{v}}^m_{d/m,i} &= \frac{\boldsymbol{R}^m_d}{m_b}\boldsymbol{F}^d_{d/m,i} + \boldsymbol{g} + C_d\boldsymbol{v}^m_{d/m,i} + \boldsymbol{d}_{i,a} \\
\dot{\boldsymbol{t}}^m_{d/m,i} &= \frac{1}{\boldsymbol{\tau}_i}(\boldsymbol{k}_i\boldsymbol{t}_{ref,i} - \boldsymbol{t}^m_{d/m,i}) \\
\dot{\psi}^m_{d/m,j} &= 0, \quad \forall j \in \mathcal{N}_i
\end{aligned} \tag{28}$$

where inputs are thrust $\boldsymbol{F}^d_{d/m,i}$ and attitude angle references $\boldsymbol{t}_{ref,i}$. Finally, g_n and h_n includes constraints to impose the safety conditions/limits such as the maximum velocity, thrust and Euler angle that the robotic agent is allowed to achieve. The following remarks will be helpful to gain more insights about the OCP:

Remark 2. (Information sharing): At each sample instant, $\boldsymbol{v}^m_{d/m,j}$, the predicted velocity of neighboring vehicles should be received and $\boldsymbol{v}^m_{d/m,i}$, the the vehicle's predicted velocity should be transmitted over the communication channel for the whole horizon. If the received information not available for some reason, the D-NMPC leverages the estimates sent by the local EKF for the prediction. In this case, the formation controller assumes a constant velocity over the prediction horizon.

Remark 3. (Prediction model): The prediction model includes the closed-loop attitude subsystem as three first order linear differential equations. This simplifies the prediction model and allows a seamless separation of tasks. Their parameters can be identified easily by conducting attitude tracking tests.

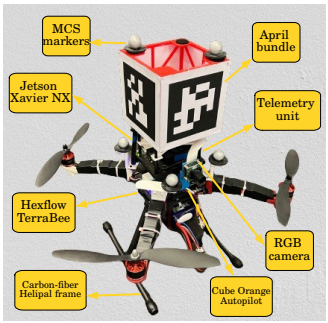

Fig. 4: The Helipal Quadrotor and its components.

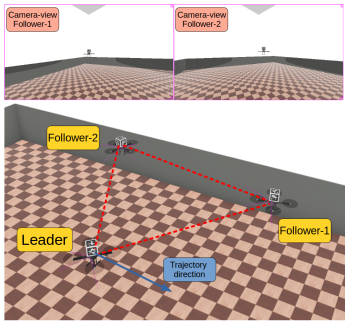

Fig. 5: Webots simulation consisting of three drones equipped with April-bundles and a RGB camera.

Remark 4. (Disturbance estimates): The disturbances acting on the prediction model, $\boldsymbol{d}_{i,a}$, are received from the estimator and assumed constant over the horizon. This is a valid assumption if the horizon length is not very long and/or the main disturbance source is model uncertainty.

Remark 5. (Perception requirements): The horizontal and vertical field of view requirements are not added to h_n in order to not increase the number of constraints on the OCP. Although it does not provide guarantees, the inclusion of the last terms in the cost function ensures a reliable perception operation.

Real-time iteration scheme

We adopt a modified Real-Time Iteration (RTI) strategy combined with estimation. This is one of the state-of-the-art solution methods applied to real-time systems as explained by [25]. In this method, first discretization then linearization are applied to the OCP around operating points and Quadratic Programming (QP) solvers are used successively to solve the generated problem. The optimality of the solution is determined by the Karush-Kuhn-Tucker (KKT) value. The computational constraints can be satisfied by monitoring the online solver time. Please refer to [13] for the details of the RTI scheme repeated for each control iteration. Note that to realize the algorithm, the nonlinear program solver ACADO [**?**] is integrated with the necessary modifications. Note that ACADO employs qpOASES as convex solver.

4 Experiments and Results

The architecture proposed has been firstly tested in simulation leveraging the high-fidelity robotics simulator Webots [27] interfaced with the Robot Operating System (ROS). A screen-shot from a formation experiment with three MAVs is shown in Fig. 5. The simulator employs a realistic quadrotor model obtained and identified in our previous work [13] where we tuned the parameters of both the estimator and controller to faithfully match the reality. Since the interfaces of the Webots simulator with ROS is exactly the same as that of physical quadrotors, the transition to physical reality was quite smooth.

The quadrotor used in our experiments is a modified Helipal Storm Drone-4 v3, endowed with a Hex Cube Orange autopilot, a Jetson Xavier NX onboard

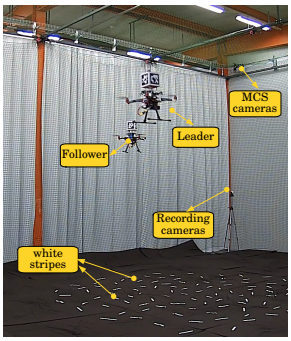

Fig. 6: Experiment arena with two drones.

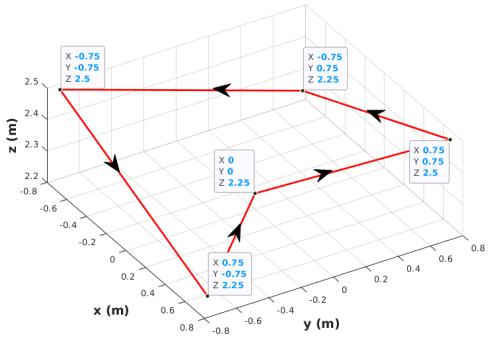

Fig. 7: Leader's trajectory for the navigation experiment.

computer, an IMU, a coupled optical flow - distance sensor, a wide FOV global shutter RGB camera and a bundle consisting of April tags, as shown in Fig. 4. The instrumented quadrotor weighs 1.71 kg and has a center-to-propeller distance of 21 cm. All computations are performed onboard by leveraging ROS. The Jetson Xavier NX computer has six parallel NVIDIA Carmel ARM v8.2 (max. 1.9 GHz) cores and one dedicated GPU (384 NVIDIA CUDA cores and 48 Tensor cores). For the experiments, all CPUs are activated with maximum power. Only the leader has access to 3D position information generated by a MCS with millimeter accuracy. The communication among the robots are realized using Wide Area Network (WAN) and the *multimasterfkie* ROS package. The trajectories are generated inside an indoor volume of 27 m^3. This arena including two quadrotors can be seen in Fig. 6. The measurement update rates of the 3D relative localization solution, optical flow sensor, distance sensor, IMU and magnetometer are 10 Hz, 20 Hz, 60 Hz, 100 Hz and 40 Hz, respectively. The control and estimation frequency is 40 Hz. NMPC for the follower solves the OCP with 40 horizon steps, which corresponds to 1 second of prediction window. The optical flow and distance sensors are selected and calibrated for indoor use. As can be seen from Fig. 6, various white stripes are placed on the floor to improve the optical flow quality. The appropriate camera and tag selection and calibration for the mission and the measures taken against vibration and blur are the critical parts of the integration. With this solution, we have a detection range of 6 m and field of view of 120 degrees. The magnetometer is not fully reliable indoor, however, the onsite and on-the-fly calibration improves the performance significantly. Lastly, the computational resources of both the onboard computer and autopilot are allocated in order to have smooth perception, communication, estimation and control.

We will present the results of two different types of experiments: formation navigation and reconfiguration, which are two common tasks for multi-robot systems performing formations. Note that you can view the video of the experiments on the following research page https://www.epfl.ch/labs/disal/research/quadrotorformationmpc/.

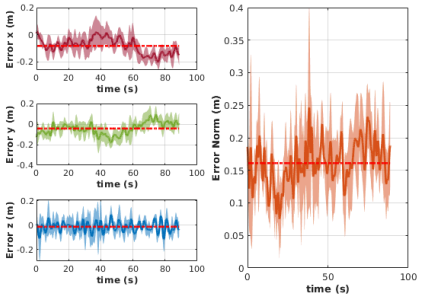

Fig. 8: Formation errors for the navigation experiment: x,y,z and norm error, dashed red lines indicate the averages over time.

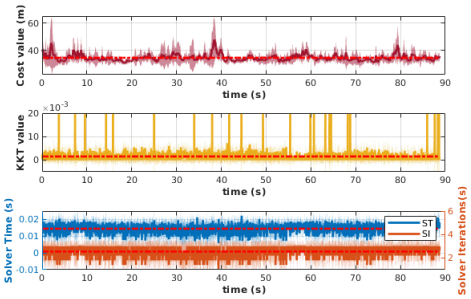

Fig. 9: Solver metrics: cost function value, KKT value, solver time and the number of iterations, dashed red lines indicate the averages over time.

4.1 Formation navigation

For this section, after the follower engaged into the formation, the leader follows a predefined 3D trajectory and the follower tries to maintain the formation geometry. The formation reference is given as [2,2,0] m and the leader's path can be seen from the Fig. 7. The total duration of the experiment is nearly 90 seconds and the same scenario is executed five times.

Fig. 8 presents the mean and standard deviations of the formation errors for each axis and their norm over five experiments. For this calculation, the MCS is employed as a ground truth. As can be seen from the figure, while the average norm error is less than 0.16 m, $\approx 5\%$ of the total range, the maximum norm error is lower than 0.27 $m \approx 9\%$ of the total range . While the mean errors on the x and y axes are different than zero, the one for the z axis is very close to zero. This is possibly due to the fact that modelling uncertainty for the altitude dynamics is less time-varying and compensated well with higher gains.

Fig. 9 shows the mean and standard deviations of the solver metrics over five experiments. These are the cost function value, KKT value, solver time and iteration number. Despite some occasional peaks, the cost value stays below a upper limit and KKT value is lower than 0.005, which is a solid indication that Nonlinear Program (NP) is converging to Quadratic Program (QP) approximations. Furthermore, the solver time remains below 0.025 seconds, with at most three SQP iterations, which in turn defines/results the sample time of the controller.

Finally, Fig. 10 shows the estimation errors, again calculated with respect to the ground truth, and disturbance estimations. As can be observed from the figure, the average error is around 0.1 m and it follows a similar trend as the formation errors. The absence of sudden peaks indicates that the FOV is maintained so that the 3D relative pose measurement is received steadily. Furthermore, the velocity estimation error in the body frame is less than 0.1 m/s on average and the disturbance estimations are mostly steady, which indicates that the biggest disturbance source in this experiment is the time-invariant model uncertainty.

4.2 Formation reconfiguration

For these experiments, the position of the leader is kept fixed and the formation reference of the follower is changed over a prescribed duration in order to

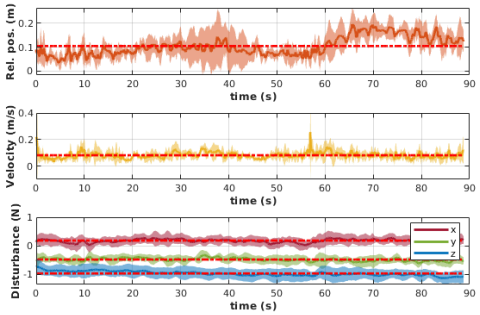

Fig. 10: Estimation errors (relative position, velocity) and disturbance estimations in 3D, dashed red lines indicate the averages over time.

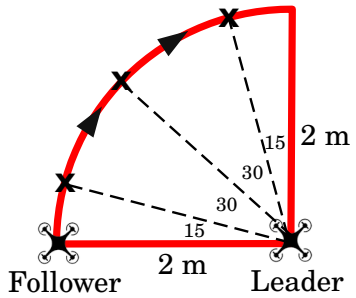

Fig. 11: Follower's reference during the reconfiguration experiment. 15-30-30 corresponds to the angles of relative position vectors

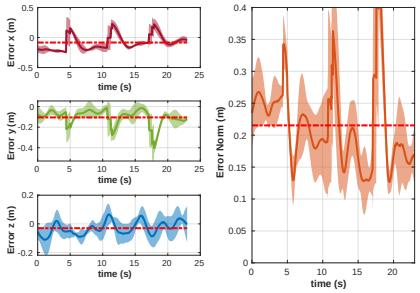

Fig. 12: The formation errors for the reconfiguration experiment: x,y,z and norm errors, dashed red lines indicate the averages over time

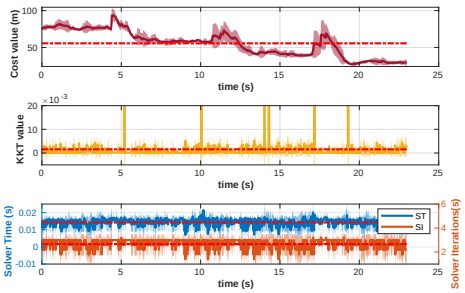

Fig. 13: Solver metrics: cost function value, KKT value, solver time and the number of iterations, dashed red line indicate the averages over time

reconfigure the formation geometry. Three distinct relative position references for the follower are displayed in Fig. 11. The total duration of the experiment is around 23 seconds and the scenario is realized five times.

The mean and standard deviations of the formation errors are given in Fig. 12. We can see the oscillatory behavior of the errors due the fact that the reference commands are sent separately over the course of experiment. As a result the mean error is around 0.21 $m \approx 7\%$ of the total range. However, the errors reach steady state after some time and on average, they stabilize around 0.17 m.

Solver metrics are shown in Fig. 13. We can make similar observations for the solver metrics, except the cost value. It can be seen that cost value is overall higher but reaches the same low levels at the end of the experiment. This is because the experiment is more dynamic compared to the formation navigation experiment. Furthermore, the plateaus observed are possibly due to the transient compromise between the individual terms in the cost function.

4.3 Discussion

In order for the reader to compare the quality of the results, we present a brief benchmarking about relative position estimation and formation errors. Recently, [28] introduced a decentralized VIO-UWB perception and estimation scheme for aerial swarms. For an indoor trajectory of 27.8 m, they obtained about 0.08 m

average relative position estimation error, which is comparable to our vision-based solution. Another state-of-the-art localization method presented by [5] targets outdoor applications with their ultra-violent (UV) markers and camera, and they achieved 1.16 m relative localization RMSE. Finally, [21] employed a distributed infrared-based relative localization system for the formations of quadrotors. The system attained an error of less than 0.20 m in range, 5 degrees in bearing, and 10 degrees in elevation for the sensor-to-target distance of 3 m. Considering formation control errors, [21] adopted a distributed graph-based formation control method that obtained a 0.27 m range and 5 degrees elevation error for the desired range of 2.26 m, which corresponds to 12% of the full range. In our previous work [13], we proposed a decentralized predictive approach and achieved an average formation error of 0.44 m. The high error observed in this work was due to the exclusion of aerodynamic disturbances occurring between vehicles and the constant neighbor velocity assumption in the prediction model.

The low average estimation ($\leq$ 0.1 m) and formation ($\leq$ 0.17 m) errors obtained in two different formation control scenarios reveal the high performance of our strategy. By leveraging the tightly-coupled distributed MPC and EKF scheme, we could simultaneously comply with the perception, communication, and vehicle constraints. Although the method is scalable in terms of algorithmic properties, a couple of issues should be addressed for real-world scalability. First, at least three wide-lens cameras should be used to perceive the neighbor vehicles in all directions. Next, the April-bundle pose detection algorithm is not lightweight, and in our current implementation, it runs on the CPU. There are efficient implementations of this algorithm using a GPU [29]. Finally, obtaining the SQP solution of NMPC is also costly, which limits the maximum number of neighbors each vehicle can have; however, there is no theoretical limit on the total swarm size.

5 Conclusion

Obtaining a robust, autonomous, performant and scalable control and estimation architecture for multi-robot systems is a significant challenge. In this work, we present a unified, predictive and distributed architecture to carry out leader-follower formations with MAVs. Our approach employs onboard resources and integrates various physical constraints such as camera FOV, actuation, velocity and acceleration limits, into the architecture. After a parameter tuning process in a high-fidelity simulator with three vehicles, we validated and showed the performance of our approach with two real quadrotors, one serving as leader and the other as follower, for two common formation tasks.

In the future, we intend to validate the approach with more following vehicles, improve the autonomy of the leader, analyze the robustness in presence of various communication problems and compare our solution with alternative approaches in detail.

Acknowledgment

This work has been partially sponsored by the FCT grant [PD/BD/135151/2017], the FCT doctoral program RBCog and the FCT project [UIDB/50009/2013]

References

1. Ebel, H., Ardakani, E. & Eberhard, P. Distributed model predictive formation control with discretization-free path planning for transporting a load. *Robotics and Autonomous Systems*. **96** pp. 211-223 (2017)
2. Oh, K., Park, M. & Ahn, H. A survey of multi-agent formation control. *Automatica*. **53** pp. 424-440 (2015)
3. Eren, U., Prach, A., Koçer, B., Raković, S., Kayacan, E. & Açıkmeşe, B. MPC in aerospace systems: Current state and opportunities. *Journal of Guidance, Control, and Dynamics*. **40**, 1541-1566 (2017)
4. Zhou, X., Wen, X., Wang, Z., Gao, Y., Li, H., Wang, Q., Yang, T., Lu, H., Cao, Y., Xu, C. & Others Swarm of micro flying robots in the wild. *Science Robotics*. **7**, eabm5954 (2022)
5. Petráček, P., Walter, V., Báča, T. & Saska, M. Bio-inspired compact swarms of unmanned aerial vehicles without communication and external localization. *Bioinspiration and Biomimetics*. **16**, 026009 (2020)
6. Alonso-Mora, J., Montijano, E., Nägeli, T., Hilliges, O., Schwager, M. & Rus, D. Distributed multi-robot formation control in dynamic environments. *Autonomous Robots*. **43**, 1079-1100 (2019)
7. Soria, E., Schiano, F. & Floreano, D. Distributed Predictive Drone Swarms in Cluttered Environments. *IEEE Robotics and Automation Letters*. **7**, 73-80 (2022)
8. Allamraju, R., Price, E., Ludwig, R., Karlapalem, K., Bülthoff, H., Black, M. & Ahmad, A. Active Perception Based Formation Control for Multiple Aerial Vehicles. *IEEE Robotics and Automation Letters*. **4**, 4491-4498 (2019)
9. Hafez, A., Marasco, A., Givigi, S., Iskandarani, M., Yousefi, S. & Rabbath, C. Solving multi-UAV dynamic encirclement via model predictive control. *IEEE Transactions on Control Systems Technology*. **23**, 2251-2265 (2015)
10. Yuan, Q., Zhan, J. & Li, X. Outdoor flocking of quadcopter drones with decentralized model predictive control. *International Society of Automation Transactions*. **71** pp. 84-92 (2017)
11. Van Parys, R. & Pipeleers, G. Distributed MPC for multi-vehicle systems moving in formation. *IEEE Robotics and Autonomous Systems*. **97** pp. 144-152 (2017)
12. Abichandani, P., Levin, K. & Bucci, D. Decentralized Formation Coordination of Multiple Quadcopters under Communication Constraints. *IEEE International Conference on Robotics and Automation*. pp. 3326-3332 (2019)
13. Erunsal, I., Ventura, R. & Martinoli, A. NMPC for Formations of Multi-Rotor Micro Aerial Vehicles: An Experimental Approach. *International Symposium on Experimental Robotics*. pp. 449-461 (2020)
14. Thakur, D., Tao, Y., Li, R., Zhou, A., Kushleyev, A. & Kumar, V. Swarm of inexpensive heterogeneous micro aerial vehicles. *International Symposium on Experimental Robotics*. pp. 413-423 (2020)
15. Olson, E. AprilTag: A robust and flexible visual fiducial system. *IEEE International Conference on Robotics and Automation*. pp. 3400-3407 (2011)
16. Fossen, T. Handbook of marine craft hydrodynamics and motion control. John Wiley and Sons (2011)
17. Mahony, R., Kumar, V. & Corke, P. Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor. *IEEE Robotics and Automation Magazine*. **19**, 20-32 (2012)
18. Omari, S., Hua, M., Ducard, G. & Hamel, T. Nonlinear control of VTOL UAVs incorporating flapping dynamics. *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 2419-2425 (2013)
19. Jazwinski, A. Stochastic processes and filtering theory Jazwinski. AH Academic Press (1970)

20. Quan, Q. Introduction to multicopter design and control. Springer (2017)
21. Dias, D. Distributed State Estimation and Control of Autonomous Quadrotor Formations Using Exclusively Onboard Resources. (EPFL–IST PhD Thesis, No. 9224) (2019)
22. https://docs.px4.io/, Autopilot control, accessed in July 2022
23. Magni, L. & Scattolini, R. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*. **42**, 1231-1236 (2006)
24. Szeliski, R. Computer vision: algorithms and applications. Springer Science Business Media (2010)
25. Gros, S., Zanon, M., Quirynen, R., Bemporad, A. & Diehl, M. From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*. **93**, 62-80 (2020)
26. Ferreau, H., Kraus, T., Vukov, M., Saeys, W. & Diehl, M. High-speed moving horizon estimation based on automatic code generation. *IEEE Conference On Decision And Control*. pp. 687-692 (2012)
27. Michel, O. Webots: Professional Mobile Robot Simulation. *International Journal of Advanced Robotic Systems*. **1**, 39-42 (2004)
28. Xu, H., Zhang, Y., Zhou, B., Wang, L., Yao, X., Meng, G. & Shen, S. Omni-Swarm: A Decentralized Omnidirectional Visual–Inertial–UWB State Estimation System for Aerial Swarms. *IEEE Transactions on Robotics*. (2022)
29. https://github.com/NVIDIA-AI-IOT/isaac_ros_apriltag, accessed in September 2022.