HAL
open science

Motor-level N-MPC for Cooperative Active Perception with Multiple Heterogeneous UAVs

Martin Jacquet, Max Kivits, Hemjyoti Das, Antonio Franchi

To cite this version:

Martin Jacquet, Max Kivits, Hemjyoti Das, Antonio Franchi. Motor-level N-MPC for Cooperative Active Perception with Multiple Heterogeneous UAVs. IEEE Robotics and Automation Letters, In press, 7 (2), pp.2063 - 2070. 10.1109/LRA.2022.3143218 . hal-03482081v1

HAL Id: hal-03482081

https://laas.hal.science/hal-03482081v1

Submitted on 15 Dec 2021 (v1), last revised 24 Mar 2022 (v3)

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.

Motor-level N-MPC for Cooperative Active Perception with Multiple Heterogeneous UAVs

Martin Jacquet[1], Max Kivits[2], Hemjyoti Das[2], Antonio Franchi[2,1]

***Abstract*—This paper introduces a cooperative control framework based on Nonlinear Model Predictive Control (NMPC) for solving an Active Information Acquisition problem (AIA) using a system of multiple multirotor UAVs equipped with onboard sensors. The observation task of the NMPC is a minimum-uncertainty pose estimation of a moving feature which is observed by the multi-UAV system, using a cooperative Kalman filter. The controller considers a full nonlinear model of the multirotors – including the motor-level actuation units and their real constraints in terms of maximum torque – and embeds the Kalman filter estimation uncertainty in its prediction. The framework allows and exploits heterogeneity in the actuation and sensing systems by considering a generic model of UAV – including both quadrotors and tilted-propeller multirotors – and a generic model of range-and-bearing sensor with arbitrary rate and field of view. The capability of the proposed framework to reduce the cooperative estimation uncertainty of a static or a moving feature, thus leading the system to optimal sensing configurations, is demonstrated through Gazebo simulations and real experiments. The software is provided open-source.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly used in a large range of applications, from aerial monitoring or exploration to work in high-risk places or human-denied areas. The monitoring tasks in particular are rapidly being deployed since they do not imply physical interactions, nor the practical issues related to safety in cluttered or human-populated environments. Such tasks include, for instance, search-and-rescue [1], autonomous cinematography [2], or indoor building inspection [3]. Furthermore, the increasing efficiency and decreasing weight of the available sensors and computation units allowed the deployment of recent efficient computer vision algorithms on UAVs [4], [5].

However, achieving such tasks in autonomy necessarily requires perception-awareness of the environment and of the object or phenomenon to be monitored. This requirement can often collide with the motion task handled by the UAV. In particular, since the sensing domain of the UAV is limited and the underactuation of most UAVs requires them to tilt in order to achieve lateral motions, the observation can be disturbed. This led the research community to study the generation of motion according to these limitations, in order to maximize the gathered information (e.g., maximizing the time coverage of a dynamic phenomenon), while increasing the accuracy of the measurements (i.e., minimizing the acquisition uncertainty). Such problem is called an Active Information Acquisition (AIA) problem [6] and is an active field of research.

To solve this problem, the monitoring is often handled by a team of several UAVs, where the various agents can share the observation burden [7]–[10]. This approach, when designed to be scalable [11], allows in particular to cover a larger set of moving features or a large area. Furthermore, in [9], the authors introduced a solution to an AIA problem relying on the Probability Hypothesis Density (PHD) filter, designed to scale with respect to (w.r.t.) the number of measurements and tracked features. This approach is extended in [10], which proposes a framework that includes collision avoidance and exploits the heterogeneity of the sensing team. The sensing area is divided into cells (Voronoi cells), according to the motion of the features, to which the agents are assigned. Another approach is to exploit the properties of the linear Gaussian model often used to represent moving features. Indeed, it has been shown in [8] that under these assumptions, a (stochastic) AIA problem can be formulated as a deterministic Optimal Control Problem (OCP). Also, under these assumptions, the feature can be optimally estimated using a Kalman filter. In later works, Model Predictive Control (MPC) – a control method that uses the dynamic model of the system to predict its behavior over a finite receding horizon – has been used to solve such AIA problems [12], [13]. In [12], an MPC approach is used on a ground robot to track a single moving feature. Therein, an MPC is used in conjunction with a Kalman filter in order to reduce the observation uncertainty over the receding horizon and thus achieve optimal sensing. Similarly, in [13], an MPC controls three UAVs in an outdoor tracking scenario. The optimal sensing configuration for the system is determined analytically, and the system is driven to this configuration by constraining the MPC. However, in the real world the robot dynamics is nonlinear and as such it cannot be handled by an MPC, which is a control method for linear systems. Because of this limitation, in the aforementioned works the MPC is used 'de facto' as a local planner that provides a position reference to the UAV, rather than generating the UAV inputs and taking into account the real input constraints of the UAV.

Another usage of optimal receding horizon control methods which has gained popularity in recent years is to embed the visibility constraints directly inside the control of the UAV [14]–[17]. Indeed, optimization-based controllers are typ-

[1]LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France, martin.jacquet@laas.fr, antonio.franchi@laas.fr
[2]Robotics and Mechatronics lab, Faculty of Electrical Engineering, Mathematics & Computer Science, University of Twente, Enschede, The Netherlands, m.p.w.kivits@student.utwente.nl, h.das@utwente.nl, a.franchi@utwente.nl
This work was partially funded by the ANR, under the Projects ANR-17-CE33-0007 'MuRoPhen' and European Commission project H2020 AERIAL-CORE (EC 871479).
The source code associated with this work can be found on git: https://redmine.laas.fr/projects/active-perception-nmpc

ically well suited to include constraints from various semantics in a single control scheme. In these works, the task of maintaining visibility over a feature or a set of features is expressed in the cost function and/or as active constraints. Therein, in contrast to what is typically done in multi-agent AIA, the full nonlinear dynamic model of the UAVs is used. This approach makes use of Nonlinear MPC (NMPC), and allows a more realistic prediction over the receding horizon, under some highly nonlinear perception objectives and constraints. This made possible also to control the UAVs via motor-torque inputs of the platform [16], [18], ensuring the feasibility of the generated motion and allowing the controller to consider the realistic actuation limitations. However, no NMPC approach has been proposed to solve an AIA problem yet.

In this work, we propose a decentralized NMPC approach to reduce the uncertainty of the observation of a mobile feature, with an arbitrary number of Generically Tilted MultiRotor (GTMR) UAVs. The UAV team can be heterogeneous, each agent having its own dynamic model, motion task and actuation limitations. We introduce a time-continuous formulation of the Intermittent Kalman filter, as well as a novel way of including a Kalman filter inside the NMPC equations, hence exploiting its predictive capabilities over the receding horizon at a reduced computational cost. The capability of the proposed framework to yield an efficient sensing configuration that reduces the overall estimation uncertainty is validated in simulations, while its capability to control a system of UAVs is demonstrated in a real experiments. Finally, the whole framework is made open-source along with the explanations and the materials required to run the simulations.

The paper is organized as follows: Sec. II presents the dynamic model of the GTMR and the feature motion and sensing models. Next, Sec. III details the uncertainty estimation process and the inclusion of the Kalman filter in the NMPC. Then, Sec. IV presents the full OCP, while Sec. V contains the simulation and experimental results. We discuss these results in contrast with the literature in Sec. VI, before concluding in Sec. VII.

II. Modeling

A. Generically Tilted Multirotor Dynamics

In the vein of previous works [16], [18], for the sake of genericity, we model the UAVs as GTMR. We refer the reader to these works for the complete modeling, as only the essentials will be reported here.

In particular, this model includes standard collinear quadrotors, and goes beyond by including tilted-propeller platforms with 4 or more propellers. What changes according to the geometry of the platform is the number of inputs, and allocation matrices. We refer to, e.g., [19] for concrete examples.

We define the world inertial frame $\mathcal{F}_W$, with its origin O_W and its axes $\mathbf{x}_W, \mathbf{y}_W, \mathbf{z}_W$. The other frames are denoted using the same convention, e.g., $\mathcal{F}_B$ is the body frame of a robot.

A GTMR is defined as a rigid body of mass m, actuated by $n \geq 4$ propellers arbitrarily placed and oriented around O_B. The position of O_B w.r.t. the $\mathcal{F}_W$ is denoted by ${}^W\mathbf{p}_B$ and the unit quaternion representing the rotation from $\mathcal{F}_B$ to $\mathcal{F}_W$ is denoted by ${}^W\mathbf{q}_B$; and similarly for all the other frame pairs.

The robot state $\mathbf{x}$ is expressed as the concatenation of the body state $\mathbf{x}_b$ and actuator state $\mathbf{x}_a$, which are defined as

$$\mathbf{x}_b = [\mathbf{p}^\top \mathbf{q}^\top \mathbf{v}^\top \boldsymbol{\omega}^\top]^\top \in \mathbb{R}^{13}, \tag{1a}$$

$$\mathbf{x}_a = \boldsymbol{\gamma} \in \mathbb{R}^n, \tag{1b}$$

where $\mathbf{p} = {}^W\mathbf{p}_B$, $\mathbf{q} = {}^W\mathbf{q}_B$, $\mathbf{v}$ is the velocity of O_B expressed in $\mathcal{F}_W$, $\boldsymbol{\omega}$ is the angular velocity of $\mathcal{F}_B$ w.r.t. $\mathcal{F}_W$, expressed in $\mathcal{F}_B$, and $\boldsymbol{\gamma} \in \mathbb{R}^n$ are propeller thrusts.

Accordingly, we define the actuation dynamics:

$$\dot{\boldsymbol{\gamma}} = \mathbf{u}. \tag{2}$$

where $\mathbf{u} \in \mathbb{R}^n$ are the system control inputs, directly related to the torques applied to the brushless motors, see [18].

The dynamic equations of a multirotor are given by (2) plus

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{3a}$$

$$\dot{\mathbf{q}} = \frac{1}{2}\, \mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}, \tag{3b}$$

$$\begin{bmatrix} m\ddot{\mathbf{p}} \\ \mathbf{J}\dot{\boldsymbol{\omega}} \end{bmatrix} = \begin{bmatrix} -mg\mathbf{z}_W \\ -\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega} \end{bmatrix} + \begin{bmatrix} \mathbf{q} \otimes \mathbf{G}_f \boldsymbol{\gamma} \otimes \mathbf{q}^* \\ \mathbf{G}_\tau \boldsymbol{\gamma} \end{bmatrix}, \tag{3c}$$

where $\otimes$ denotes the Hamilton product of two quaternions, $\mathbf{q}^*$ is the conjugate quaternion of $\mathbf{q}$, $\mathbf{J} \in \mathbb{R}^{3\times 3}$ is the positive definite inertia matrix, g is the intensity of the gravity force, and $\mathbf{G}_f$ and $\mathbf{G}_\tau \in \mathbb{R}^{3\times n}$ are respectively the *force* and *moment allocation matrices* [19], mapping the forces produced by each propeller to the total force and moment acting on the body.

In this work, we consider a team of $N_a \geq 1$ UAVs that all comply to this general GTMR model but may be heterogeneous, i.e., having different shapes, number of rotors or sensing parameters. In the following, when relevant, the quantities referring to the i-*th* agent are subscripted $\cdot_i$.

B. Feature Motion

The observed feature M, whose state is denoted $\mathbf{x}_M$, is modeled as a linear Gaussian system:

$$\dot{\mathbf{x}}_M = \mathbf{A}\mathbf{x}_M + \boldsymbol{\eta}_\mathbf{Q}, \quad \boldsymbol{\eta}_\mathbf{Q} \sim \mathcal{N}(0, \mathbf{Q}), \tag{4}$$

where $\mathbf{A}$ is the transition matrix and $\boldsymbol{\eta}_\mathbf{Q}$ is the 0-mean Gaussian noise of covariance matrix $\mathbf{Q}$.

C. Sensor Measurement Model

The sensor installed on each UAV is modeled as a punctual device S_i rigidly attached to the body, such that the pose transformation between $\mathcal{F}_{S_i}$ and $\mathcal{F}_B$ is constant and known. Each sensor has a specific field of view (FoV), and provides an uncertain 3D measurement of an object of interest as long as it falls into its FoV. The observation model for the i-*th* sensing agent is defined as

$$\mathbf{z_i} = \mathbf{H}_i \mathbf{x}_M + \boldsymbol{\eta}_{\mathbf{R}_i}, \quad \boldsymbol{\eta}_{\mathbf{R}_i} \sim \mathcal{N}(0, \mathbf{R}_i), \tag{5}$$

where $\mathbf{H}_i$ is the observation matrix, and $\boldsymbol{\eta}_{\mathbf{R}_i}$ is the 0-mean Gaussian measurement noise of covariance matrix $\mathbf{R}_i$.

The fact that the i-*th* sensor provides a measurement of M is described using a FoV function [20], [21], defined as

$$\lambda_i = \begin{cases} 1, & \mathbf{z}_i \text{ is inside the FoV} \\ 0, & \mathbf{z}_i \text{ is outside the FoV} \end{cases}. \tag{6}$$

This FoV function depends of the relative pose between the agent and the feature, hence can discriminate whether a predicted configuration would yield a measurement [12].

We note that the feature motion is expressed in $\mathcal{F}_W$, and so are the measurements. We assume known the localization of $\mathcal{F}_{S_i}$ in $\mathcal{F}_W$, hence leave the impact of the transformation between the two frames on the measurement uncertainty $\mathbf{R}_i$ out of the scope of this work.

This type of sensor is called *range-and-bearing* sensor, since it provides both a 2D angular information and a 1D depth information. Such sensors are typically lidars or depth cameras (stereo or infrared), and are in particular opposed to *bearing only* sensors, such as monocular cameras. However, the latter type of device is the most easy to embark on a UAV because of its lightweight, and thus the most widely used. An extra software processing layer, using either some prior geometric information of the object [22] or a deep learning based algorithm [4], can be added, since the *bearing sensors + software* block can be considered as a *range-and-bearing* sensor.

D. Measurement Uncertainty

Following the linear Gaussian sensing model (5), for a *range-and-bearing* sensor, the measurement covariance matrix $\mathbf{R}$ can be pictured as a 3D ellipsoid, centered on the measurement mean, and whose eigenvalue σ_z along the bearing vector (i.e., associated with the *range*) is different from the eigenvalues σ_{xy} along the two orthogonal directions (associated with the *bearing*) [6], [7], [23]. For cameras (RGBD, stereo or monocular), σ_z is typically larger that σ_{xy}, since the depth information either comes from image processing that adds up the noise from several pixel measurements (typically stereo alignment or *PnP*), or from an imprecise infrared measurement. Hence, the measurement uncertainty matrices $\mathbf{R}_i$ can be written using a *SVD*, whose transformation matrix $\mathbf{T}_{\theta_i}$ is the 3D rotation matrix about the angle θ_i between $\mathbf{z}_W$ and the bearing vector (see Fig. 1).

Additionally, the distance toward the feature influences the uncertainty. It is commonly assumed [7], [23] that the observation is optimal at a given distance d_{ref} from the feature. Hence, we propose to scale the eigenvalues σ_{xy} and σ_z according to this distance, which might be arbitrarily defined.

Finally, the measurement covariance matrix is written

$$\mathbf{R}_i = \mu \cdot (d_i - d_{\mathrm{ref}})^2 \cdot \mathbf{T}_{\theta_i} \begin{bmatrix} \sigma_{xy} & 0 & 0 \\ 0 & \sigma_{xy} & 0 \\ 0 & 0 & \sigma_z \end{bmatrix} \mathbf{T}_{\theta_i}^\top, \qquad (7)$$

where d_i is the distance from sensor to feature, μ is a scalar scaling factor. The relation between the measurement covariance and the agents/feature mutual poses plays a crucial role because it is what the NMPC will be able to exploit to reduce the estimation uncertainty.

III. Estimation Uncertainty Minimization

In this section, we introduce the estimation process and how the estimator is included inside the NMPC predictive scheme to minimize the uncertainty along the receding horizon.

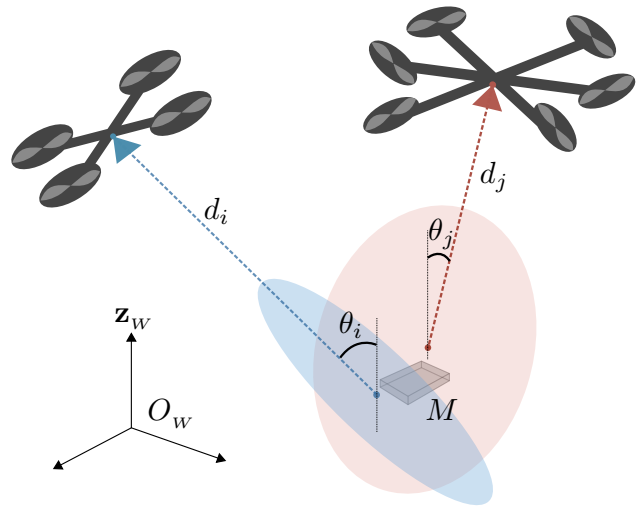

Fig. 1: Two heterogeneous UAVs, numbered i and j, equipped with sensors, observing a feature M. The measurement uncertainties are represented by the two colored ellipsoids, centered on there respective measurement mean. The dashed lines correspond to the two bearing vectors. d_i and d_j are the respective distances toward the feature.

A. Feature Estimation and Prediction

The feature state estimation is done by using an intermittent-measurement time-continuous Kalman-Bucy filter. For the sake of readability, the equations of this Kalman filter use the matrices notations introduced in Sec. II-B and II-C.

In [20], the authors presented an intermittent Kalman filter which takes into consideration the observation weighting function λ. This filter has been extended to the multi-agent case in [21].

We propose a time-continuous extension of these filters, which differential equations are, after (4) and (5),

$$\dot{\hat{\mathbf{x}}}_M = \mathbf{A}\hat{\mathbf{x}}_M + \boldsymbol{\lambda}\mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_M), \qquad (8)$$

$$\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{Q} - \boldsymbol{\lambda}\mathbf{K}\mathbf{R}\mathbf{K}^\top, \qquad (9)$$

where $\hat{\mathbf{x}}_M$ is the estimate of $\mathbf{x}_M$, $\mathbf{P}$ is the estimate covariance matrix, and $\mathbf{z}$, $\boldsymbol{\lambda}$, $\mathbf{R}$ are the concatenated measurement vectors, FoV functions and covariance matrices. The Kalman gain is given by $\mathbf{K} = \mathbf{P}\mathbf{H}^\top\mathbf{R}^{-1}$.

We note that this formulation considers the limit case introduced in [20] where the measurement covariance tends toward infinity when $\lambda = 0$, implying that the measurement has no effect on the filter update.

B. NMPC with the Kalman Filter state

For each agent, we extend the state vector of the NMPC, $\mathbf{x}$, with the internal Kalman filter state $\hat{\mathbf{x}}_M$. In order to propagate the covariance through the receding horizon, we also include in $\mathbf{x}$ a minimal representation of the (symmetric) estimation covariance matrix $\mathbf{P}$. We denote n_M the dimension of $\hat{\mathbf{x}}_M$, and the minimal vector representation $\mathbf{P}_\triangle$, composed of the lower triangular part of $\mathbf{P}$, is defined as the vector

$$\mathbf{P}_\triangle = \left[p_{ij}\right], \quad 1 \leq i \leq j \leq n_M. \qquad (10)$$

The NMPC state vector of each sensing agent is written as

$$\mathbf{x} = [\mathbf{x}_b^\top \quad \mathbf{x}_a^\top \quad \hat{\mathbf{x}}_M^\top \quad \mathbf{P}_\triangle^\top]^\top, \qquad (11)$$

and its time derivative $\dot{\mathbf{x}}$ is obtained from (2), (3), (8) and (9).

In order to reduce the amount of extra state variables in the NMPC, we use the motion model of a constant position, which yields a minimal Kalman filter state size of $n_M = 3$. This also implies that $\mathbf{A} = \mathbf{0}_3$ and $\mathbf{H} = \mathbf{I}_3$, respectively the zero and identity matrices in $\mathbb{R}^{3\times3}$, which simplifies the Riccati equation (9). We note that to further reduce the state, improving the performances at the cost of precision, the cross-variance terms p_{xy}, p_{xz} and p_{yz} could be also neglected.

However, to overcome the imprecision induced by this simple motion model, the predicted state of the feature over the receding horizon is computed using an external estimation/predictor (e.g., a similar Kalman-Bucy filter as introduced in Sec. III-A with a more complete motion model, such as a constant or damped acceleration model). These predictions are used in the NMPC to compute the associated $\mathbf{R}_i$ and λ_i over the horizon, predicting how would evolve the observation uncertainty $\mathbf{P}$. Coupled with a high process noise $\mathbf{Q}$ in the internal Kalman filter, this allows an efficient internal trajectory prediction.

C. Uncertainty Minimization

In order to effectively reduce the feature estimation uncertainty, we need to consider a mathematical tool that represents the 'magnitude' of the covariance matrix $\mathbf{P}$. In information theory, minimizing the *Shannon differential entropy* of a continuous Gaussian variable is equivalent to minimizing the determinant of its covariance matrix. In the case of symmetric definite-positive matrix, the determinant is upper-bounded by the trace, which can be used as alternative metric to optimize. These two approaches are respectively called D-optimal and A-optimal [6], [7]. The first approach aims at reducing the generalized covariance, while the second reduces average covariance. In order to reduce the computation load, we chose to use the latter approach. i.e., to minimize $\mathrm{tr}(\mathbf{P})$.

D. Continuous Field of View Function

In order to avoid the discontinuity in $\mathrm{tr}(\mathbf{P})$, induced by the λ_i FoV functions, we replace (6) with a continuous approximation [12], which is a function of the FoV angles of each sensing agent:

$$\lambda_i = \frac{1}{1 + \exp\left(-\beta\left(\frac{{}^C\mathbf{p}_M}{\|{}^C\mathbf{p}_M\|} - \cos\alpha\right)\right)}, \tag{12}$$

where α is the angular FoV, and $\beta > 0$ defines the curvature of the sigmoid. The FoV is assumed to be conic in order to simplify (12). Alternatively, a pyramidal FoV could be used by defining α as a function of ${}^C\mathbf{p}_M$, see [16].

This formulation only considers an angular limitation for the FoV. A depth limit can be introduced by adding a second term in (12), as proposed in [12]. This second term can however be omitted since the influence of depth on the uncertainty close is already modeled in (7), driving the UAV to stay inside the FoV range limits.

We note that if the feature is not observed by an agent, the NMPC relies on the predictions from the external Kalman filter (and, in a multi-agent scenario, from the measurements from other agents). However these predictions would have no effect on the update of the internal covariance of the NMPC, since $\lambda_i = 0$. Hence, the agent will tend to recover visibility to effectively reduce $\mathrm{tr}(\mathbf{P})$.

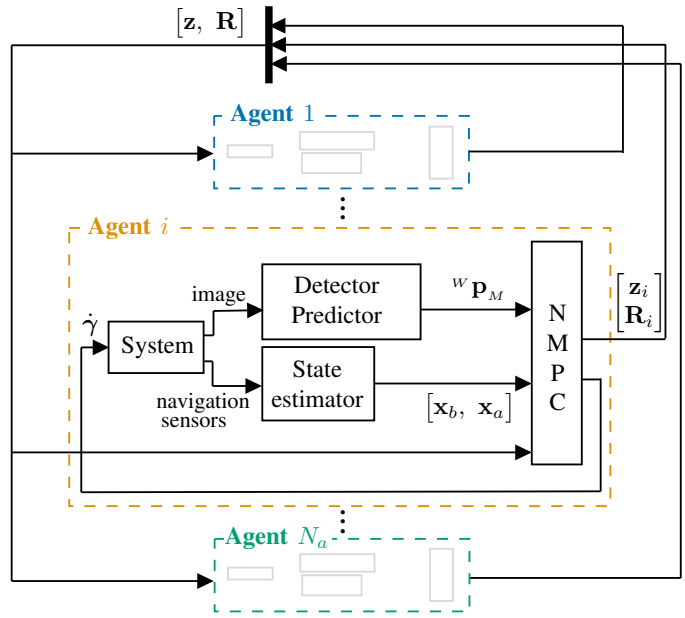

Fig. 2: Block diagram of the decentralized framework. The controller of the *i-th* agent is detailed. All the measurements and covariances, predicted over the receding horizon are communicated to every agent.

IV. Optimal Control Problem Formulation

A. Decentralized Control Policy

The proposed control policy can be formulated either as centralized or decentralized. In the first case, the states of each agent $\left[\mathbf{x}_{b,i}^\top\ \mathbf{x}_{a,i}^\top\right]^\top$ are concatenated into the full system state. The controller has full knowledge of the system and computes the control inputs of each agent accordingly.

However, in order to compute the estimation uncertainty, the agents only needs to know the full measurement vector $\mathbf{z}$, and the uncertainty matrix $\mathbf{R}$. Hence, in a decentralized formulation, each agent is controlled by its own NMPC, estimates the feature position using a Kalman filter, and communicates its predicted measurements to the rest of the team. Then, each NMPC uses the concatenated measurements as external parameters to update their internal Kalman filter, using (8) and (9).

We note that this assumes a perfect instantaneous communication, which is far from being true in real scenarios. In case of failed or delayed communication, the UAVs would still be able to rely on the previous measurements, which is a correct approximation when dealing with relatively slow motions, but needs to be considered when dealing with agile maneuvers.

The decentralized formulation is more generic and scalable w.r.t. the number of agents, hence the OCP presented hereafter corresponds to this formulation. The block diagram in Fig. 2 presents the corresponding controller.

B. Optimal Control Problem

As per [16], we consider realistic physical limitations of the motors in the NMPC. Such limitations are equivalently recast as constraints on $\boldsymbol{\gamma}$ and $\dot{\boldsymbol{\gamma}}$, which lower and upper bounds can be obtained through an identification campaign, as shown in [18].

In order to ensure the stability of the system, a motion task is defined for each agent by an output $\mathbf{y}$ and its reference $\mathbf{y}_r$, which is provided, over the receding horizon, as a trajectory in position, attitude, and their first and second order derivatives.

The motion task cost is defined as the weighted square Euclidean norm of the difference between $\mathbf{y}$ and its reference, denoted $\|\cdot\|^2_{\mathbf{W}}$, $\mathbf{W}$ being a diagonal weight matrix.

In addition to this motion task, the NMPC cost function contains the observation task, which consists in minimizing the cumulative sum of estimation uncertainties over the receding horizon. This task is weighted by a scalar, denoted $w_{\mathbf{P}}$.

The discrete-time OCP for each sensing agent, over the receding horizon T, sampled in N shooting points, at a given instant t, is expressed as

$$\min_{\substack{\mathbf{x}_0 \dots \mathbf{x}_N \\ \mathbf{u}_0 \dots \mathbf{u}_{N-1}}} \sum_{k=0}^{N} \|\mathbf{y}_k - \mathbf{y}_{r,k}\|^2_{\mathbf{W}} + \sum_{k=0}^{N} w_{\mathbf{P}} \mathrm{tr}(\mathbf{P}_k) \tag{13a}$$

$$\begin{aligned}
s.t. \quad & \mathbf{x}_0 = \mathbf{x}(t) && && (13b)\\
& \mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{p}_k), && k\in\{0,N-1\} && (13c)\\
& \mathbf{y}_k = \mathbf{h}(\mathbf{x}_k), && k\in\{0,N\} && (13d)\\
& \mathbf{P}_k = \mathbf{s}(\mathbf{x}_k), && k\in\{0,N-1\} && (13e)\\
& \underline{\mathbf{x}_b} \leq \mathbf{m}(\mathbf{x}_{b,k}, \mathbf{p}_k) \leq \overline{\mathbf{x}_b}, && k\in\{0,N\} && (13f)\\
& \underline{\boldsymbol{\gamma}} \leq \boldsymbol{\gamma}_k \leq \overline{\boldsymbol{\gamma}}, && k\in\{0,N\} && (13g)\\
& \underline{\dot{\boldsymbol{\gamma}}_k} \leq \mathbf{u}_k \leq \overline{\dot{\boldsymbol{\gamma}}_k}, && k\in\{0,N-1\} && (13h)
\end{aligned}$$

where $\mathbf{x}(t)$ is the measurement of the current state and $\mathbf{f}$ synthetically denotes the dynamics of the agent, expressed in (2), (3), (8) and (9). The function $\mathbf{h}$ is the system output map, and $\mathbf{s}$ denotes the function that reconstructs the $\mathbf{P}$ matrix from of $\mathbf{P}_{\triangle}$ in $\mathbf{x}$. Potential motion constraints on the body state $\mathbf{x}_b$ are expressed using a selection function $\mathbf{m}$. The lower and upper bounds of the inequality constraints are denoted using respectively $\underline{\cdot}$ and $\overline{\cdot}$. Finally, $\mathbf{p}_k$ are the external parameters passed to the NMPC, namely the predicted measurements and covariances.

For safety reasons, we added collision avoidance between the agents. This is done by imposing linear constraints on the distance between each agent, along the 3 position coordinates, expressed in (13f). Thus, each UAV forbids the others to enter a region of the workspace, assimilating agents as boxes.

V. Validation

A. Experimental and Simulation Setup

The framework is implemented in C++, using GenoM [24] which is a middleware-independent component generator, that can be compiled for a given middleware, e.g., ROS. The NMPC implementation is the one introduced in [16], based on [25]. It uses a 4th order explicit Runge-Kutta integrator. The hardware interface as well as the state estimation and path planning are done using the TeleKyb3 software, available on the OpenRobots platform[1].

Fig. 3: One of the quadrotors used in the presented experiment.

The software framework can later be connected to the actual platform, or to a Gazebo simulated system that emulates the platform interface. Details on how to use this software can be found in the provided git repository[2].

The experiments presented in this section are performed using two standard quadrotors, each equipped with an onboard Intel NUC (Intel Core i7-8565U and 8GB of DDR3 RAM). One of these quadrotors is shown in Fig.3.

The UAVs are equipped with a downfacing monocular camera, an Intel RealSense T265 (or is simulated through Gazebo). This camera is chosen for its lightweight, wide FoV and practicality, though none of the odometry functionalities are used. The camera frequency is set to 30Hz. The observed feature is an AruCo fiducial marker [22], both in simulations and experiments. The tag is either static or moving on the ground, and is tracked with a Kalman filter using a damped acceleration model, and running at 1kHz in order to provide the measurements $\mathbf{z}_i$ to the NMPCs.

The UAV state $[\mathbf{x}_b]$ is estimated using a combination of internal IMU and external motion capture, however the motion capture could be replaced by any visual-inertial odometry method, such as the ones presented in [26].

In the experiments and simulations presented in this section, the UAVs are controlled using the proposed NMPC. Once the UAVs are in position, the observation task is enable (by changing the weights in (13a)) for each agent simultaneously. At the same time, the weights on the position task are set to zero, and small weights are applied to maintain constant roll/pitch and small velocities, in order to penalize large motions and ensure the stability of the system. Therefore, the motions observed are driven only by the observation task in the cost function.

In the multi-agent case, the UAVs exchange data directly through wifi. The communication frequency is 10Hz.

Videos of all the reported experiments and simulations can be found in the attached multimedia file. Additional videos of experiments and simulations achieved using a moving feature can also be found therein.

The computational time of the NMPCs on the onboard computers have been recorded during the various experiments,

[1] `git.openrobots.org/projects/telekyb3`
[2] `redmine.laas.fr/projects/active-perception-nmpc`

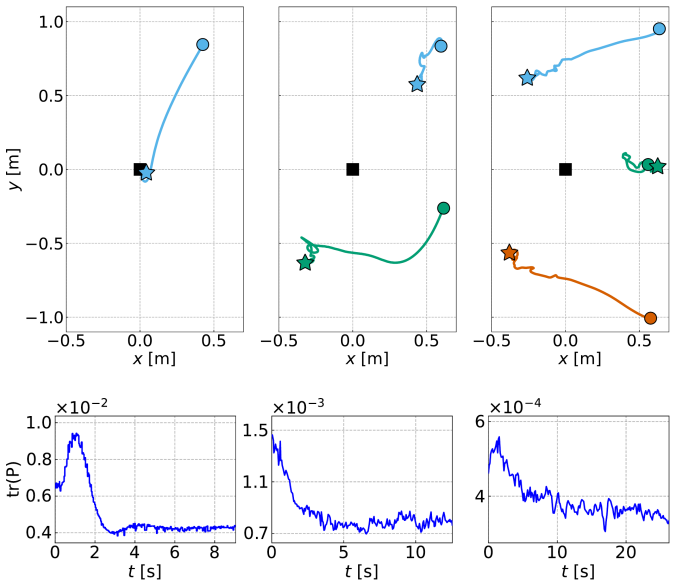

Fig. 4: The (x, y) position over time (top) and the evolution of $\mathrm{tr}(\mathbf{P})$ over time (bottom) for the 1 UAV (left), 2 UAVs (middle) and 3 UAVs (right) cases. The initial positions of the UAVs (when the observation task is enabled) and the final positions are respectively denoted by the circles and stars. The feature position in $(0, 0)$ is denoted by the black square.

both with 1 and 2 robots, for a total flight duration of about 10 minutes. The corresponding boxplot is reported in Fig. 5.

B. Static Feature Sensing

This section presents the behavior of a system of $N_a = 1$, 2 or 3 quadrotor(s), assigned to reduce the estimation uncertainty of a static feature. Various behaviors emerge when more sensing capabilities are added to the system and are reported hereafter. The cases $N_a = 1$ and 2 are performed on real quadrotors, while the case $N_a = 3$ is performed in simulation.

The (x, y) positions of the agents as well as the value of the estimation uncertainty $\mathrm{tr}(\mathbf{P})$ over time are reported in order to illustrate how the system configuration is affected by the observation task to yield a lower overall observation uncertainty. We note that $\mathrm{tr}(\mathbf{P})$ is computed using an external Kalman filter, which aggregates the measurements from all the agents, and is thus agnostic to the internal NMPC processes.

1) Case $N_a = 1$: In this scenario, minimizing $\mathrm{tr}(\mathbf{P})$ is equivalent to minimizing $\mathrm{tr}(\mathbf{R})$. Given the model presented in II-D, $\mathrm{tr}(\mathbf{R})$ is minimal when the feature is aligned with the principal axis of the sensor, at a distance d_{ref}. Indeed, the UAV driven by the NMPC, whose motion is reported in Fig. 4 (left column), converges to the (x, y) position of the feature. The UAV motion is faster at the beginning because the cost gradient is steeper, as it can bee seen in the multimedia file. The NMPC also makes the quadrotor move upward, since tilting at the initial height would induce a loss of visibility.

The bottom graph shows the evolution of $\mathrm{tr}(\mathbf{P})$ over time. The system is reducing this uncertainty w.r.t. its value at the initial position. We note that the estimation first increases (since the UAV tilts and goes up) before converging to a smaller value, showing that the proposed algorithm is non greedy and is able to overcome the initial local minimum.

	MPC driven config.			naive config.	
	1 UAV	2 UAVs	3 UAVs	2 UAVs	3 UAVs
mean($\mathrm{tr}(\mathbf{P})$)	5.04e-3	8.55e-4	3.91e-4	3.9e-3	7.2e-4
std($\mathrm{tr}(\mathbf{P})$)	1.54e-3	1.64e-4	5.21e-5	8.13e-4	7.25e-4
mean($\mathrm{tr}(\mathbf{R})$)	4.4e-3	6.8e-3	6.6e-3	4.0e-3	1.08e-3
std($\mathrm{tr}(\mathbf{R})$)	6.7e-3	9.8e-3	9.79e-5	5.2e-3	7.2e-4

TABLE I: Position covariances (measurement and estimation) Mean and Standard Deviation (std) using 1, 2 or 3 UAV(s).

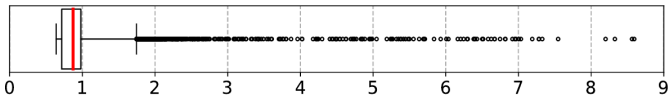

Fig. 5: Boxplot of the computation time (in milliseconds) of the onboard NMPC, recorded using an horizon of $T = 0.75$ s, sampled in $N = 20$ points. The mean (0.87 ms) is depicted in red, and the black dots are the slowest 1% control cycles recorded, ranging from 1.74 to 8.6 ms.

2) Case $N_a \geq 2$: With more UAVs, the sensing configuration that is optimal for one robot is no longer optimal. The system exploits the extra sensing capabilities to observe the feature more efficiently, by placing each agents on opposite sides w.r.t. the feature, hence improving the overall sensing, as shown in the second and third columns of Fig.4. In particular, in the 3 UAVs case, the achieved configuration is similar to the one reported in [13], [23], where the agents are positioned around the feature, spaced by 120°. However, we remark that the convergence in the 3 UAVs case is slower since the initial estimation is already good, and the benefit of moving to a different position is smaller.

3) Estimation improvement: We validate the pertinence of the proposed framework by effectively measuring the estimation improvement yielded by the sensing configuration. We report in Tab. I some quantitative statistics on the estimation (trace of $\mathbf{P}$) and measurement (trace of $\mathbf{R}$) uncertainties for the position of the feature, using 1, 2 and 3 UAVs.

The last two columns report the statistics gathered with respectively 2 and 3 UAVs, manually driven right above the feature (referred hereafter as 'naive configuration, since it is the one observed with 1 UAV and naively extrapolated). As expected, adding additional measurements improves the estimation uncertainty (of about an order of magnitude), compared to the 1 UAV case (first column). However, both with 2 and 3 UAVs, the NMPC driven configurations yield much smaller uncertainties than the respective naive configuration,.

Indeed, in this naive configuration, the agents observe the feature from about the same angle. Instead, because of the shape of the uncertainty ellipsoids (see Fig. 1), observing from different angles allows to compensate the poor *range* estimation of both sensors by exploiting the better *bearing* estimations (or vice-versa if $\sigma_{xy} > \sigma_z$).

In addition, we note that this estimation improvement is achieved by degrading the individual measurement, demonstrating a emergent collaborative behavior, despite the decentralization.

C. Asymmetric Sensing Team

This section presents a simulation using an asymmetric heterogeneous sensing team. The two agents are one quadrotor

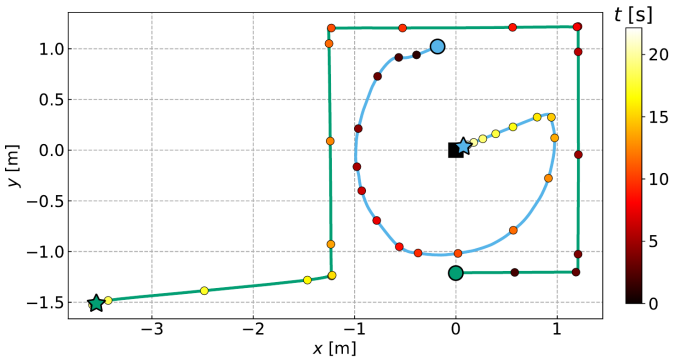

Fig. 6: The (x, y) position of both UAVs (blue for hexarotor, green for quadrotor). The colored dots represent their respective position each second, while the color gradient indicates the time at which the positions were recorded. The circles and stars are respectively the initial and final positions of the UAVs, while the black square is the feature position.

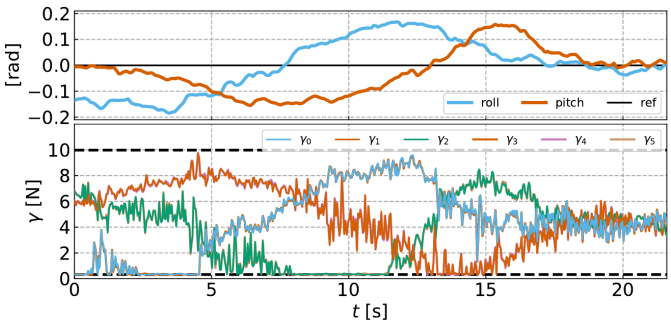

Fig. 7: Above, the roll and pitch of the tilted-propeller hexarotor over time, with their reference value (0 rad). Below, the thrusts exerted by the $n = 6$ propellers, with the lower and upper bounds marked by the black dashed lines.

and one tilted-propeller hexarotor, showing the capability of the framework to handle various types of UAV using different dynamics. In addition, the tasks assigned to the agents are asymmetrical: the hexarotor is tasked as in Sec.V-B (minimal motion task and observation task), while the quadrotor has a given motion task to achieve (reach a set of successive waypoints) but no observation task ($w_{\mathbf{P}} = 0$). The quadrotor however participates to the observation..

The behavior of the system is reported in Fig. 6, where the (x, y) configuration of the system is depicted over time. When the quadrotor moves to reach a waypoint, the hexarotor positions itself on the opposite side, improving the overall estimation, achieving a configuration similar to the 2 UAVs case presented in V-B2.

The roll and pitch of the hexarotor are reported in Fig. 7 (top plot). The hexarotor stays stable in a tilted configuration, hence reducing its measurement uncertainty, while maintaining a configuration that minimizes the overall estimation. Figure 7 (bottom) also shows that the propeller thrusts $\boldsymbol{\gamma}$ actually reach their bounds during this motion. This demonstrates that the NMPC is able to exploit the full actuation span of the platform to perform the task.

When the quadrotor reaches its final waypoint (after 15 s), it looses sight on the feature. The hexarotor then handles the observation burden alone and moves as in the 1 UAV case, i.e. goes on top of the feature, since this configuration minimizes both the measurement uncertainty and the attitude error. This demonstrates the effect of the FoV function λ, which, when becoming 0, indicates that the agent has no influence on the overall estimation, allowing the system to reconfigure accordingly.

VI. Discussion

In this section we propose a qualitative comparison with similar approaches in the literature. As mentioned in the introduction, this work is the first, to the best of our knowledge, to use NMPC to solve AIA problems.

Compared to other non-MPC approaches to AIA [7]–[9], our approach couples the UAV low-level control with the observation task. Considering perception-aware controllers has been an active field of study over the recent years, in particular using NMPC [14], [15], [17]. Therein, the perception can be expressed alongside objectives and constraints from different semantics in a single OCP. In particular, this control/perception coupling allows the controller to exploit the platform actuation span at its best to, e.g., simultaneously achieve a given motion while maintaining perception [16], or to maximize the perception accuracy, as depicted in Sec. (V-C).

However, considering the active observation as an AIA problem allows more versatility than the perception-aware NMPC previously introduced, since the uncertainty minimization can account for external measurements of the feature. In particular, it allows the decentralization of the observation as a shared objective rather than constraining each individual agent. More generally, any additional measurement source can be considered, such as static cameras that would provide additional measurements to the NMPC.

There exists, however, a corpus of works tackling such problems with linear MPC [8], [12], [13]. But considering linear models does not come without drawbacks. In particular, for underactuated UAVs (e.g., collinear quadrotors), there is a nonlinear coupling between the translational and rotational states, which implies that the orientation of the camera is dependent on the UAV motion in a nonlinear way. Disregarding such nonlinearity could lead to a loss of visibility over a feature while performing a motion (in particular agile maneuvers). The fact that, using NMPC, a quadrotor is commanded to move upward to maintain visibility while moving toward a feature, as in the attached multimedia file, is a consequence of considering its complete nonlinear dynamics.

Additionally, there is a nonlinear coupling between the agent and feature states in both the measurement uncertainty matrix $\mathbf{R}$ and the FoV function $\boldsymbol{\lambda}$. The propagation of these quantities over the horizon is not possible without simplifications. In [12], this problem is solved by considering an arbitrary constant value for $\mathbf{R}$; and relaxing (12) by precomputing the nonlinear orientation terms in a first step. These two simplifications have some impact on the accuracy of the propagation. Yet the main improvement of our approach is that considering the nonlinear coupling in $\mathbf{R}$ makes the NMPC able to not only maintain the visibility, but also effectively reduce the observation uncertainty. In particular, the collaborative emerging behaviors observed in V-B are a consequence of

considering the full stochastic observation model.

Finally, other works [13], [23] have an 'offline' approach of the AIA problem. Therein, the controller is tasked to maintain a predetermined formation that is proven to yield optimal measurements. Such approach is more rigid, considers a fixed number of agents and does not allow the system to reconfigure according to additional tasks given to the various agents. We rather propose an approach where finding the compromise between minimizing the uncertainty and accounting for a separate motion task is devoted the NMPC, according to a set of weights and parameters (which can be changed online by a supervisor to prioritize one or the other).

VII. Conclusion

In this work we have introduced a NMPC-based cooperative control framework to solve AIA for a system with multiple multirotor UAVs equipped with onboard sensors. We proposed a time-continuous formulation of the intermittent measurement Kalman filter and a strategy to exploit the predictive aspect of this filter in the receding horizon. This allows to propagate the estimation uncertainty. The controller considers a full nonlinear constrained model of the multirotors at the motor-level rather than a simplified approximation of it. Thus, it allows a direct application without an intermediate low level controller and full exploitation of the UAV capability without unnecessary abstraction layers. The framework allows and exploits heterogeneity in the actuation of the sensing agents. Simulations and experiments have shown how the proposed implementation is able to effectively yield to emergent optimal sensing configuration, adapted to the number of agents.

Possible extensions of this work include, for instance, an improvement of the scalability of the framework to handle multiple features without enlarging even more the NMPC state vector. This objective could be perhaps achieved by adopting a PHD filter in place of the Kalman filter. Additionally, the framework could be extended to make it fully robust to severe communication issues that may appear in some cases, such as packet drops. This extension could be achieved, e.g., by exploiting the intermittent aspect of the proposed Kalman filter. Finally, the collision between agents should also be considered using a suitable modification of the cost function.

References

[1] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, 2020.

[2] A. Alcántara, J. Capitán, A. Torres-González, R. Cunha, and A. Ollero, "Autonomous execution of cinematographic shots with multiple drones," *IEEE Access*, vol. 8, pp. 201 300–201 316, 2020.

[3] P. Petracek, V. Kratky, and M. Saska, "Dronument: System for reliable deployment of micro aerial vehicles in dark areas of large historical monuments," *IEEE Robotics and Automation Letters*, vol. 5, pp. 2078–2085, Apr. 2020.

[4] P. Zhang, Y. Zhong, and X. Li, "Slimyolov3: Narrower, faster and better for real-time uav applications," in *2020 IEEE/CVF Int. Conf. on Computer Vision*, 2019.

[5] Y. Akbari, N. Almaadeed, S. Al-maadeed, and O. Elharrouss, "Applications, databases and open computer vision research from drone videos and images: a survey," *Artificial Intelligence Review*, vol. 54, no. 5, pp. 3887–3938, 2021.

[6] P. Yang, R. A. Freeman, and K. M. Lynch, "Distributed cooperative active sensing using consensus filters," in *2007IEEE Int. Conf. on Robotics and Automation*, Apr. 2007.

[7] F. Morbidi and G. L. Mariottini, "Active target tracking and cooperative localization for teams of aerial vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 5, pp. 1694–1707, Sep. 2013.

[8] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Information acquisition with sensing robots: Algorithms and error bounds," in *2014 IEEE Int. Conf. on Robotics and Automation*, 2014, pp. 6447–6454.

[9] P. Dames, P. Tokekar, and V. Kumar, "Detecting, localizing, and tracking an unknown number of moving targets using a team of mobile robots," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1540–1553, 2017.

[10] J. Chen and P. Dames, "Collision-free distributed multi-target tracking using teams of mobile robots with localization uncertainty," in *2020 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 6968–6974.

[11] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, "Asymptotically optimal planning for non-myopic multi-robot information gathering." in *Robotics: Science and Systems*, 2019, pp. 22–26.

[12] C. Liu and J. K. Hedrick, "Model predictive control-based target search and tracking using autonomous mobile robot with limited sensing domain," in *2017 American Control Conference*, 2017, pp. 2937–2942.

[13] R. Tallamraju, E. Price, R. Ludwig, K. Karlapalem, H. H. Bülthoff, M. J. Black, and A. Ahmad, "Active perception based formation control for multiple aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4491–4498, 2019.

[14] B. Penin, P. R. Giordano, and F. Chaumette, "Minimum-time trajectory planning under intermittent measurements," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 153–160, 2018.

[15] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct. 2018.

[16] M. Jacquet and A. Franchi, "Motor and perception constrained NMPC for torque-controlled generic aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 518–525, Apr. 2021.

[17] G. Li, A. Tunchez, and G. Loianno, "PCMPC: Perception-constrained model predictive control for quadrotors with suspended loads using a single camera and imu," in *2021 IEEE Int. Conf. on Robotics and Automation*, Jun. 2021.

[18] D. Bicego, J. Mazzetto, R. Carli, M. Farina, and A. Franchi, "Nonlinear model predictive control with enhanced actuator model for multi-rotor aerial vehicles with generic designs," *Journal of Intelligent & Robotics Systems*, vol. 100, no. 3, pp. 1213–1247, 2020.

[19] G. Michieletto, M. Ryll, and A. Franchi, "Fundamental actuation properties of multirotors: Force-moment decoupling and fail-safe robustness," *IEEE Trans. on Robotics*, vol. 34, no. 3, pp. 702–715, 2018.

[20] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[21] C. Yang, J. Zheng, X. Ren, W. Yang, H. Shi, and L. Shi, "Multi-sensor kalman filtering with intermittent measurements," *IEEE Trans. on Automatic Control*, vol. 63, no. 3, pp. 797–804, 2017.

[22] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[23] T. H. Chung, V. Gupta, J. W. Burdick, and R. M. Murray, "On a decentralized active sensing strategy using mobile sensor platforms in a network," in *2004IEEE Conf. on Decision and Control*, vol. 2, 2004, pp. 1914–1919.

[24] A. Mallet, C. Pasteur, M. Herrb, S. Lemaignan, and F. Ingrand, "GenoM3: Building middleware-independent robotic components," in *2010 IEEE Int. Conf. on Robotics and Automation*, 2010, pp. 4627–4632.

[25] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, "MATMPC - a MATLAB based toolbox for real-time nonlinear model predictive control," in *2019 European Control Conference*, Jun. 2019, pp. 3365–3370.

[26] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 2502–2509.