



HAL
open science

Model Predictive Control for Dynamic Quadrotor Bearing Formations

Julian Erskine, Rafael Balderas-Hill, Isabelle Fantoni, Abdelhamid Chriette

► **To cite this version:**

Julian Erskine, Rafael Balderas-Hill, Isabelle Fantoni, Abdelhamid Chriette. Model Predictive Control for Dynamic Quadrotor Bearing Formations. IEEE International Conference on Robotics and Automation, May 2021, Xi'an (virtual), China. hal-03177635

HAL Id: hal-03177635

<https://hal.science/hal-03177635v1>

Submitted on 23 Mar 2021

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.

Model Predictive Control for Dynamic Quadrotor Bearing Formations

Julian Erskine¹, Rafael Balderas-Hill¹, Isabelle Fantoni², and Abdelhamid Chriette¹

Abstract—Formation control of multi-agent systems deals with groups of robots forming specific spatial geometries. Combined with the advancements of unmanned aerial vehicles (UAVs) in the past decade, formation control may potentially be applied to tasks such as search-and-rescue, surveillance, even collaborative manipulation. A key challenge is the decentralization of formation control, where each agent behaves independently using onboard sensors and computation, improving the scalability and robustness of the system.

This paper proposes a decentralized controller based on model predictive control (MPC), for the control of formations of quadrotor UAVs defined by inter-agent bearings. The use of MPC allows the controller to account for attitude kinematics, improving upon the results of existing bearing formation control methods based on rigidity and visual servoing approaches, which typically only consider the quadrotor as a single or double integrator. Furthermore the near-optimality of MPC permits a more optimal use of the quadrotors dynamic capabilities for faster maneuvering. Extensive simulations are performed to demonstrate the improved transient formation convergence and fast maneuvering permitted by this controller. Experiments show that it is indeed a real-time feasible solution for bearing formation control.

Index Terms—Multi-Robot Systems, Formation Control, Model Predictive Control, Quadrotor Swarms, Visual Servoing

I. INTRODUCTION

Multi-agent systems (MAS) are of great interest to the robotics community, due to the potential for system redundancy, cost-effectiveness, and augmented capabilities beyond those of individual robots [1]. A computer able to communicate with all agents can act as a centralized controller, but decentralization is often preferred due to superior scalability and robustness to communication delays and single point failures. Initial work focused on swarm behaviour and flocking [2], generally following Reynolds’ rules [3] for which the collective motion is more important than the positioning of individual agents. Recently however there has been significant work in the field of formation control, where precise inter-agent geometries may be optimal for certain tasks such as intruder searches [4], search-and rescue [5], cooperative mapping [6], and even cinematography [7].

The type of control law for MAS generally depends on the type of information available to the controllers, and on the

J. Erskine is the corresponding author.

This work was supported by the French region Pays de la Loire through the RFI AtlanSTIC2020 project RAPID and by the Natural Sciences and Engineering Research Council of Canada

¹École Centrale de Nantes, Laboratoire des Sciences du Numérique de Nantes (LS2N), UMR CNRS 6004, 1 rue de la Noe, 44321 Nantes, France.

² LS2N, Centre National de la Recherche Scientifique, France.

Emails: {julian.erskine, rafael.balderashill, isabelle.fantoni, abdelhamid.chriette}@ls2n.fr

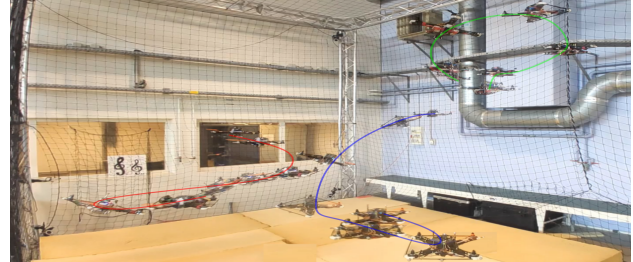


Fig. 1: Formation flight with three quadrotors

constraints of the agents. In situations where the positions of all agents are known, a wide selection of controllers are available [8], [9]. The knowledge of all positions in a common frame however is a best-case scenario. Formations of agents that must evolve in unmapped and GPS-denied environments must be able to self-localise with their neighbours using onboard measurements. Often, this takes the form of distance measurements [10], [11] or bearing (often referred to as visual) measurements [12]–[14] which return incomplete information about the state of the formation. In this paper we only consider bearing measurements, which are extractable from cheap and lightweight cameras.

Most bearing formation controllers (see [14] for survey) only consider the translation and yaw kinematics of quadrotors, neglecting the full attitude kinematics to which they are coupled (due to the well-known under-actuation of quadrotors), and which become significant during aggressive maneuvering. For agents simulated as single or double integrators this is not an issue, but for real quadrotors this limits the reactivity of the system due to the low gains and filtering which are required to avoid highly dynamic attitudes, and which may result in instability. To solve this, we apply model predictive control (MPC) considering the quadrotors thrust and angular velocity as control inputs to track the desired bearing trajectory of the formation. This allows us to account for the (almost) full dynamics of the quadrotor in the control law, permitting near-optimal convergence and maneuvering of the formation. The major contributions of this paper compared to existing work on MPC for formation control [15]–[17] is the use of bearing error in the optimisation cost function, allowing for a weak dependence on the inter-agent distance. We can thus say our controller is “almost-bearing-only” (i.e. functions with only bearing sensors in most situations), while providing a more reactive formation compared to existing bearing-only controllers.

The paper is structured as follows: Section II presents the modelling of quadrotors and bearing formations. In

section III, rigidity based control of bearing formations are presented, along with the concept of model predictive control. The application of MPC to bearing formation control is presented in section IV, and is supported by extensive simulations in section V and by real-time experiments in section VI. Finally in section VII we discuss the successes and drawbacks of this novel MAS control strategy.

II. MODELLING

A. Quadrotor Modelling

Each quadrotor i is considered as a rigid body with frame $\mathfrak{F}_i = \{O_i, \mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i\}$ moving with respect to a common inertial frame $\mathfrak{F}_0 = \{O_0, \mathbf{x}_0, \mathbf{y}_0, \mathbf{z}_0\}$. The primary interest in decentralized bearing control is to act without any external system of localization. As such, the position $\mathbf{p}_i = \overline{O_0 O_i}$ is unknown. The quadrotors are, however, able to measure their velocity \mathbf{v}_i , acceleration $\dot{\mathbf{v}}_i$, and angular velocity $\boldsymbol{\omega}_i$ in \mathfrak{F}_i using onboard optical flow sensors, accelerometers, and gyroscopes respectively.

The attitude of the quadrotors are also known. Indeed, an accurate estimation of the roll and pitch (denoted as ϕ_i and θ_i) is necessary for stable flight, and is often recovered from a sensor fusion of the accelerometer and gyroscope. The yaw (denoted as ψ_i) of each agent i is more difficult as the magnetic field used for yaw estimations is easily perturbed [18], even by the electric current in the UAV itself [19]. In the case where a consistent estimate of the absolute yaw is infeasible, a decentralized bearing estimator such as [20] can be used for each agent to converge to a common (although arbitrarily rotated) reference frame for the yaw. We choose to use the quaternion $\mathbf{q} = [q_w \ q_x \ q_y \ q_z]^T$ representation for the attitude control due to its computational efficiency.

The state of the i^{th} quadrotor evolves according to:

$$\dot{\mathbf{p}}_i = \mathbf{v}_i \quad (1)$$

$$\dot{\mathbf{v}}_i = \frac{f_i}{m_i} \begin{bmatrix} 2(q_w q_y + q_x q_z) \\ 2(q_y q_z - q_w q_x) \\ 1 - 2(q_x^2 + q_y^2) \end{bmatrix} + \mathbf{g} \quad (2)$$

$$\dot{\mathbf{q}}_i = \frac{1}{2} \begin{bmatrix} 0 \\ \boldsymbol{\omega}_i \end{bmatrix} \otimes \mathbf{q}_i \quad (3)$$

$$\dot{\boldsymbol{\omega}}_i = \mathbf{I}_i^{-1} \boldsymbol{\tau}_i - \mathbf{I}_i^{-1} (\boldsymbol{\omega}_i \times \mathbf{I}_i \boldsymbol{\omega}_i) \quad (4)$$

where m_i is the mass, f_i is the thrust along the \mathbf{z}_i axis, and $\boldsymbol{\tau}_i$ is the steering moment of the quadrotor. The inertia tensor \mathbf{I}_i , as well as $\boldsymbol{\omega}_i$ and $\boldsymbol{\tau}_i$ are expressed in \mathfrak{F}_i , while \mathbf{p}_i , \mathbf{v}_i , and \mathbf{q}_i are expressed in \mathfrak{F}_0 . The gravitational acceleration $\mathbf{g} = [0 \ 0 \ -9.81]^T \text{ ms}^{-2}$ is also expressed in \mathfrak{F}_0 .

Each quadrotor is controlled by setting the angular velocity of four co-planar propellers. Each propeller p generates a thrust force $f_p = k_t \Omega_p^2$ and a drag torque $\tau_p = \text{sign}(\Omega_p) k_d \Omega_p^2$, where k_t and k_d are aerodynamic coefficients and Ω_p is the rotational velocity of the propeller. The vector of squared propeller velocities $\boldsymbol{\Omega}_i = [\Omega_1^2 \ \dots \ \Omega_4^2]^T$ can be directly related to the actuation wrench $\mathbf{w}_i = [f_i \ \boldsymbol{\tau}_i^T]^T$ by

$$\mathbf{w}_i = \boldsymbol{\Gamma} \boldsymbol{\Omega}_i \quad (5)$$

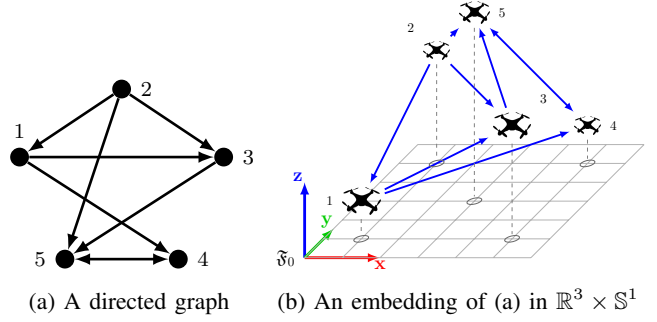


Fig. 2: The graph and framework representations of a directed bearing formation, where an arrow from \mathcal{A}_i to \mathcal{A}_j indicates a measurement β_{ij}

where $\boldsymbol{\Gamma}$ depends on the quadrotor geometry and on the aerodynamic coefficients of the propellers [21]. Generally the propeller velocities themselves behave as first order systems but this is neglected along with drag effects and lower level motor and battery dynamics due to difficulties (although not insurmountable [22]) in accurately identifying them.

B. Formation Modelling

Typically in formation control, the formation is represented as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ composed of a set of vertices \mathcal{V} and a set of edges \mathcal{E} as in Fig. 2a. Each agent \mathcal{A}_i is represented by a vertex $\mathcal{V}_i \in \mathcal{V}$, and each inter-agent measurement corresponds to an edge $\mathcal{E}_{ij} = \mathcal{V}_i \times \mathcal{V}_j \in \mathcal{E}$. Graphs only represent the flow of information, thus a framework $\mathcal{F}(\mathcal{G}, \mathbf{q})$ is defined such that $\mathbf{q} = [\mathbf{q}_1^T \ \dots \ \mathbf{q}_n^T]^T$ (known as the embedding) associates a state \mathbf{q}_i to each \mathcal{V}_i as shown in Fig. 2b. Because of the under-actuation of quadrotors, the embedding is chosen as the state of the quadrotor projected on the $\mathbb{R}^3 \times \mathbb{S}^1$ manifold, thus the embedded state of \mathcal{A}_i is $\mathbf{q}_i = [\mathbf{p}_i^T \ \psi_i]^T$. As the quadrotor frame \mathfrak{F}_i evolves in $SE(3)$, a vector $\mathbf{x}_i \in \mathbb{R}^3$ in \mathfrak{F}_i can be reprojected onto the $\mathbb{R}^3 \times \mathbb{S}^1$ manifold using a ZYX Euler angle convention.

$$\mathbf{x}_i^{\mathbb{R}^3 \times \mathbb{S}^1} = \mathbf{R}_y(\phi_i) \mathbf{R}_x(\theta_i) \mathbf{x}_i^{SE(3)} \quad (6)$$

We assume that each quadrotor (or agent) is equipped with a camera, capable of measuring the direction (or bearing) of other agents in its own frame of reference (see Fig. 3). The bearing of \mathcal{A}_j wrt \mathcal{A}_i expressed in \mathfrak{F}_i can be written as

$$\beta_{ij} = \mathbf{R}_z^T(\psi_i) \frac{\mathbf{p}_{ij}}{d_{ij}} \in \mathbb{S}^2 \quad (7)$$

where $\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i$ is the position of \mathcal{A}_j with respect to \mathcal{A}_i expressed in \mathfrak{F}_0 , $d_{ij} = \|\mathbf{p}_{ij}\|$ is the distance between the two agents, and \mathbf{R}_z is a rotation matrix about the \mathbf{z}_0 -axis. Note that the yaw can be expressed as

$$\psi = \text{atan2}(2q_w q_z + 2q_x q_y, -2q_y^2 - 2q_z^2 + 1) \quad (8)$$

III. CONTROL BACKGROUND

To understand the benefits of MPC in the context of formation control, we discuss existing control methods for formations, beginning at the level of the quadrotor and moving up to the higher level formation controller.

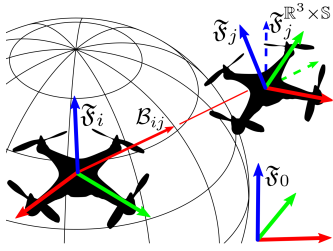


Fig. 3: Agent \mathcal{A}_i measuring the bearing β_{ij} . As \mathcal{A}_i is quasi-static, it has no need to reproject β_{ij} . However \mathcal{A}_j is performing a high-dynamics maneuver, and thus must reproject any measurements in \mathfrak{F}_j onto its $\mathbb{R}^3 \times \mathbb{S}^1$ manifold.

A. Quadrotor Control Structure

The most common approach to deal with the inherent under-actuation of quadrotors is to have multiple layers of control, with an outer translation controller tracking a high-level position or velocity trajectory by generating a thrust and reference attitude for the low-level attitude controller. The attitude controller uses the attitude estimation and body-rate measurements to feed either a kinematic or dynamic controller in order to track the reference attitude by applying actuation moments which, along with the thrust, are then converted to propeller speeds through Eq. (5). As a general rule, the high-level planners run at 1-20 Hz, the translation controllers at 20-200 Hz, and the attitude and motor level controllers run at frequencies greater than 200 Hz.

B. Existing Bearing Controllers

The control of bearing formations consists primarily in driving the formation measurements $\beta = [\beta_1^T, \dots, \beta_{|\mathcal{V}|}^T]^T$ (where $\beta_i(t) = [\beta_{ij}(t)^T \dots \beta_{in}(t)^T]^T$ is the stack of bearings observed by \mathcal{A}_i) toward to a desired set of values β^d . Secondary tasks such as moving the formation may also be imposed. Existing work on quadrotor bearing formations generally uses the first order kinematics of the formation, expressed as

$$\dot{\beta} = \mathbf{B} \begin{bmatrix} \mathbf{v} \\ \dot{\psi} \end{bmatrix} \quad (9)$$

where \mathbf{B} is the bearing rigidity matrix (similar to the Jacobian in other robotic domains) relating the change in bearings to the velocity $\mathbf{v} = [\mathbf{v}_1^T \dots \mathbf{v}_{|\mathcal{V}|}^T]$ and yaw rates $\dot{\psi} = [\dot{\psi}_1 \dots \dot{\psi}_{|\mathcal{V}|}]$ of the agents. The quadrotors are then controlled by their body frame velocities and yaw rates such as to minimize the bearing error $\mathbf{e}_\beta = \beta^d - \beta$. A successful example of this is the rigidity controller from [20] which uses the jacobian transpose control law

$$\begin{bmatrix} \mathbf{v} \\ \dot{\psi} \end{bmatrix} = k_c \mathbf{B}^T \beta^d \quad (10)$$

allowing for a compact, low computation controller which is easily decentralizable and is tuned by a single gain k_c .

It is important to discuss the previously mentioned steering of the bearing formation. It has been proven in [20] that any (rigid) bearing formation in $\mathbb{R}^3 \times \mathbb{S}^1$ has a five-dimensional

matrix in the kernel of the bearing rigidity matrix. This corresponds to the five trivial motions

$$\mathfrak{M} = \text{span}(\ker(\mathbf{B})) = [\mathbf{v}_{\mathcal{F}} \in \mathbb{R}^3 \ \dot{\psi}_{\mathcal{F}} \in \mathbb{S}^1 \ \dot{s}_{\mathcal{F}} \in \mathbb{R}^1] \quad (11)$$

corresponding to a spatial translation $\mathbf{v}_{\mathcal{F}}$ of the formation, a rotation $\dot{\psi}_{\mathcal{F}}$ of the formation about \mathbf{z}_0 , and a change of scale¹ $\dot{s}_{\mathcal{F}}$ of the formation. Any trivial motion can thus be expressed in the body-frame velocity of \mathcal{A}_i as

$$\mathbf{v}_i = \mathbf{R}_i^T (\mathbf{v}_{\mathcal{F}} + \dot{\psi}_{\mathcal{F}} (\mathbf{c}_{\mathcal{F}} - \mathbf{p}_i) \times \mathbf{z}_0 + \dot{s}_{\mathcal{F}} (\mathbf{c}_{\mathcal{F}} - \mathbf{p}_i)) \quad (12a)$$

$$\dot{\psi}_i = \dot{\psi}_{\mathcal{F}} \quad (12b)$$

where $\mathbf{c}_{\mathcal{F}}$ is a finite position in space about which the formation rotates and expands, taken in this paper as the centroid of the agents. The fact that the motions in Eq. (12) lie in the kernel of the rigidity matrix decouples them from the first order bearing control problem.

Other control laws exist treating agents either as double integrators or as non-holonomic mobile robots, but few formation control works consider the coupling between the attitude and translational dynamics of the quadrotors. In fact, this coupling poses a significant problem for accurate bearing control as the quadrotor is not able to instantly generate the velocity or acceleration set by the formation controller, due to the need to reorient itself. This requires that either the formation control gains are low or that the control signal is heavily filtered, which results in slow convergence to the desired formation and less reactive steering.

C. Model Predictive Control

MPC was initially pioneered to control complex systems with constraints. Recently it has become a popular tool in robotics due to its near optimality (it is truly optimal only for an infinite control horizon [23]) and constraint handling capabilities. It has in fact been used in various quadrotor applications trajectory planning and tracking [24], [25], as well as for multi-objective tasks such as tracking a trajectory while keeping and object in view [26].

MPC is performed by predicting the state \mathbf{q} of the system over a prediction horizon T_p in the future, discretized into N_p samples, as a function of the current state \mathbf{q}_0 , a desired trajectory $\mathbf{q}^d(t)$, and the future control inputs $\mathbf{u}(t)$. The control input is optimized to minimize an objective function

$$\begin{aligned} \min_{\mathbf{u}} \quad & \sum_{i=1}^{N_p} \mathcal{O}(\mathbf{q}, \mathbf{q}^d, \mathbf{u}, \Delta t) \\ \text{s.t.} \quad & \underline{\mathbf{u}} \leq \mathbf{u} \leq \bar{\mathbf{u}} \\ & \mathcal{C}_{eq}(\mathbf{q}, \mathbf{u}) = 0 \\ & \mathcal{C}(\mathbf{q}, \mathbf{u}) \leq 0 \end{aligned} \quad (13)$$

where \mathcal{O} is the objective function to be minimized, \mathcal{C}_{eq} and \mathcal{C} are constraint functions which must be respected, and $\underline{\mathbf{u}}$ and $\bar{\mathbf{u}}$ are the lower and upper bounds of the control signal. Generally once the optimization converges, the first values of the computed control input sequence $\mathbf{u}(t_0)$ are applied to

¹The scale $s_{\mathcal{F}}$ is defined as the mean distance between each quadrotor, and the centroid of the formation $\mathbf{c}_{\mathcal{F}}$

the system, and the optimization is run again using updated state information.

IV. MPC FOR BEARING FORMATIONS

The first decision is to choose a control variable that may be accurately applied to the quadrotor so that it will evolve as close to the prediction as possible. We choose to use $\mathbf{u}_i = [f_i \ \boldsymbol{\omega}_i^T]^T$ as the quadrotor control, which will be tracked by a low-level control attitude rate controller setting the motor speeds. This is because $\boldsymbol{\omega}_i$ is measured directly, can be integrated to recover the future attitude, and the attitude rate dynamics of quadrotors are very fast allowing rapid tracking by the low-level rate controller. Motor control is not used because the required rate for stable, vibration-free flight may be difficult to reliably achieve with onboard real-time optimization, together with potential computation or image processing tasks.

A. Prediction

Given the choice of control signal, we choose the state

$$\mathbf{q}_i = [\mathbf{q}_i, \mathbf{v}_i, \mathbf{p}_i] \quad (14)$$

for which the prediction $\mathbf{q}_i(t)$ may be calculated from a control signal $\mathbf{u}_i(t)$, $t \in T_p$ and the state \mathbf{q}_i at t_0 (up to an unknown translation) by integrating Eq. (1-3) with respect to time. Each predicted bearing $\beta_{ij}(t)$ is then defined by Eq. (7-8). The position of \mathcal{A}_j relative to \mathcal{A}_i expressed in \mathfrak{F}_i given the bearing measurement β_{ij} at time t_0 is predicted as

$$\hat{\mathbf{p}}_j(t) = \beta_{ij} \hat{d}_{ij} + \mathbf{v}_{\mathcal{F}} t \quad (15)$$

where \hat{d}_{ij} is an approximate distance and the $\mathbf{v}_{\mathcal{F}} t$ term acts as the MPC equivalent of a velocity feed-forward term. Note that this requires that the inter-agent relative yaw is well estimated, which requires communication between connected nodes in the formation graph [20]. The distance may be taken as either the desired distance corresponding to a desired formation scale, an estimated distance (using such methods as [27], [28] for scale estimation), or even a unitary distance as is often done in 2D visual servoing [29]. As there is no known initial position \mathbf{p}_i in an inertial frame, the predicted position $\mathbf{p}_i(t)$ is therefore relative to \mathfrak{F}_i at the time t_0 , projected onto the $\mathbb{R}^3 \times \mathbb{S}^1$ manifold.

B. Objective Function

The objective function of \mathcal{A}_i may be a function of the predicted state $\mathbf{q}_i(t)$, bearings $\beta_i(t)$, and inputs $\mathbf{u}_i(t)$. As discussed in section III-B, there are two primary tasks of formation control:

- 1) Achieve the desired inter-agent geometry
- 2) Steer the formation by it's trivial motions

We therefore formulate the objective function as a multi-objective optimization problem of the shape

$$\mathcal{O}_i = \mathcal{O}_{\beta_i} + w \mathcal{O}_{\mathfrak{M}_i} \quad (16)$$

where \mathcal{O}_{β_i} is a function minimizing the bearing error, and $\mathcal{O}_{\mathfrak{M}_i}$ is a function minimizing the trivial motion error, and w is an inter-objective weight.

The bearing error $\mathbf{e}_{\beta_i} = \beta_i^d - \beta_i$ minimization is achieved using the function

$$\mathcal{O}_{\beta_i} = \sum_{n=1}^{N_p} \mathbf{e}_{\beta_i}^T(n\Delta t) \mathbf{Q}_{\beta_i} \mathbf{e}_{\beta_i}(n\Delta t) \quad (17)$$

where \mathbf{Q}_{β_i} is a positive diagonal matrix.

The second task minimizes the trivial motion error

$$\mathbf{e}_{\mathfrak{M}_i} = [\mathbf{v}_i^d(t)^T \psi_i^d(t)]^T - [\mathbf{v}_i(t)^T \psi_i(t)]^T \quad (18)$$

where $\mathbf{v}_i^d(t)$ and $\psi_i^d(t)$ come from the mapping of a desired trivial motion \mathfrak{M}^d (given by a pilot or a higher level control algorithm for example) to the quadrotor motion by Eq. (12). The objective function is thus

$$\mathcal{O}_{\mathfrak{M}_i} = \sum_{n=1}^{N_p} \mathbf{e}_{\mathfrak{M}_i}^T(n\Delta t) \mathbf{Q}_{\mathfrak{M}_i} \mathbf{e}_{\mathfrak{M}_i}(n\Delta t) \quad (19)$$

where $\mathbf{Q}_{\mathfrak{M}_i}$ is a positive diagonal matrix. Because $\mathfrak{M} \in \ker(\mathbf{B})$ applies only to the first order model whereas we use a third order model (i.e. the input $\boldsymbol{\omega}_i$ must be thrice integrated to get the bearing output). An individual minimization of the two objectives may yield conflicting inputs, thus w is used to shape the Pareto frontier of the two tasks.

C. Constraints

The optimization of the objective function may account for constraints that must necessarily be satisfied. There are many possible constraints for this system, but we only briefly discuss them for the sake of completeness.

1) *Operational Constraints*: These are constraints relating to the quadrotor itself, its actuation limits, operational stability, or similar criteria. The only critical one in our case is the input bounds from Eq. (13), which are defined (for our quadrotor) as

$$\underline{f}_i \leq f_i \leq \bar{f}_i \quad (20a)$$

$$\underline{\boldsymbol{\omega}}_{xy,i} \leq \boldsymbol{\omega}_{xy,i} \leq \bar{\boldsymbol{\omega}}_{xy,i} \quad (20b)$$

$$\underline{\boldsymbol{\omega}}_{z,i} \leq \boldsymbol{\omega}_{z,i} \leq \bar{\boldsymbol{\omega}}_{z,i} \quad (20c)$$

where \underline{x} and \bar{x} are respectively the lower and upper limits of x . These constraints are chosen such that there is no motor saturation during an aggressive manual flight respecting these limits. Other constraints could include limitations on the velocity, and on the maximum tilt angle of the quadrotor.

2) *Task Constraints*: There are no task constraints implemented currently, however we may easily envisage constraints ensuring that all observed quadrotors remain in the field of view of the observing quadrotor as in [30], which require the persistent observation of a target outside the formation [31], or ensuring collision avoidance with environmental obstacles [32].

V. SIMULATIONS

Because of the difficulty in proving the stability of decentralized MPC in this application, this section shows the results of extensive simulations to both demonstrate the stability of the proposed MPC control strategy, as well as

TABLE I: Relationship between bearing error norm $\|e_\beta\|$ and the equivalent angular error e_α between the measured and desired values of a single bearing measurement

e_α (deg)	180	90	45	30	15	10	5	1
$\ e_\beta\ $	2.0	1.41	0.77	0.52	0.26	0.17	0.09	0.02

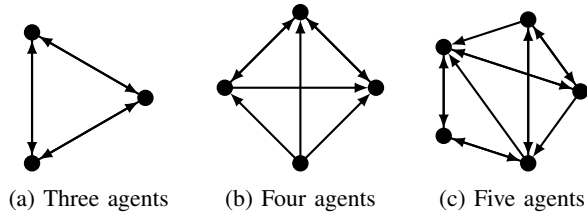


Fig. 4: Test Formation Graphs

to compare its performance relative to other state of the art bearing control methods. We compare our controller with the rigidity-based controller in [20], which considers only first-order quadrotor kinematics, but follows a locally optimal gradient due to the controller of \mathcal{A}_i considering not only it's observations β_{ij} , but also all observations β_{ki} of itself. In all simulations and experiments, in order to compare results between formations of different sizes, the bearing error magnitude is scaled by $1/\sqrt{|\mathcal{E}|}$, and thus always has a maximum value of 2.0, regardless of the number of edges. Table I can be referred to for a comparison between bearing magnitude error and the more interpretable angular error.

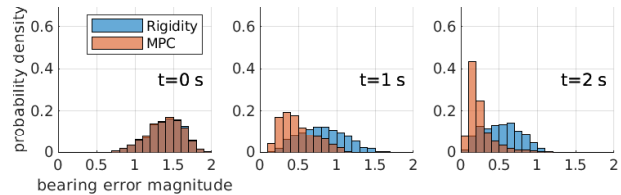
The quadrotors are modelled in Simulink using Eqns. 1-4 with noise and first order propeller dynamics. The velocity (P), attitude (PD) [33] and body rate (P) controllers run at 200 Hz, while the outer loop bearing formation controllers run at 30 Hz. We use the well-known Matlab function *fmincon* to solve the optimization, while the prediction uses a fourth-order Runge-Kutta (RK4) algorithm.

With all agents starting at random positions and yaw angles, and the desired bearings based on another set of random position and yaw angles, the simulation is run for five seconds using both the proposed MPC controller and the rigidity controller. The results of 1110 simulations of the three, four, and five-agent formations (Fig. 4) are shown in Fig. 5a-b, with only formations considered as potentially converging (an average bearing error of $e_\alpha \leq 45^\circ$ at 5 s) are kept, with the others being assumed to have either crashed (due to controller instability) or fallen into local minimas. It is seen that despite the lack of formal proof, the system shows good convergence properties as only 4% of MPC simulations failed to converge, while 12% of Rigidity controllers failed.

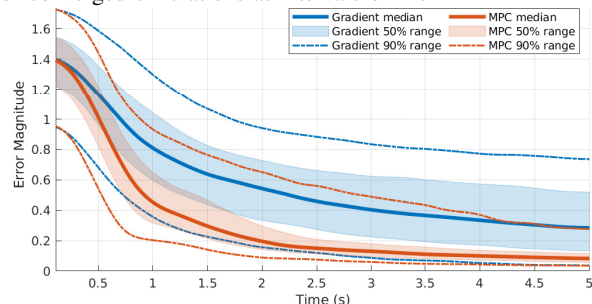
VI. EXPERIMENTS

A. Real-time MPC Implementation

We used the ACADO toolbox [34] to generate efficient C-code for an SQP (sequential quadratic programming) minimization of the objective function with multiple shooting discretization, RK4 integration, and using the QPoeses library



(a) Probability density histograms of the bearing error magnitude of converged simulations at intervals of 1 s



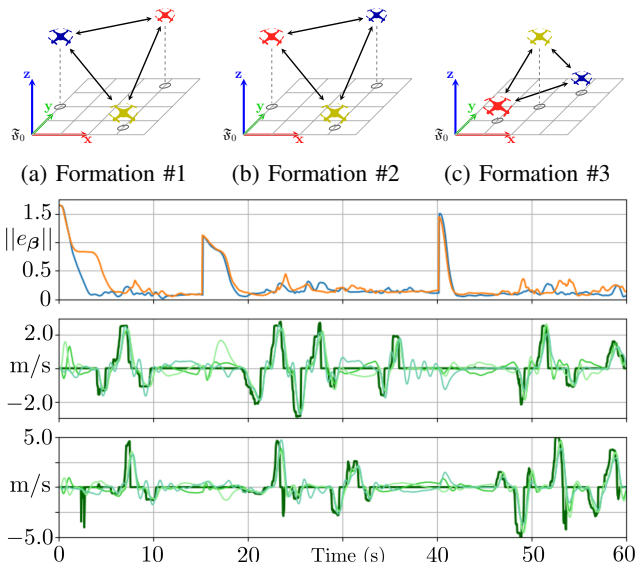
(b) A continuous-time box and whisker plot of the bearing error in successful simulations

Fig. 5: The results extensive simulations comparing the MPC (red) and rigidity (blue) controllers.

for solving sparse quadratic programming problems. Since ACADO is pre-compiled, it lacks the runtime flexibility to re-dimension the MPC problem for the variable number of graph edges leaving each vertex (as in Fig. 4b-c). Rather than pre-compile separate code for each agent, we generated code based on a maximum number of observations per agent (we used four) and set the weights of unused measurements to zero at runtime. After testing, the best results were obtained when running the MPC at 50 Hz, performing two steps of the SQP algorithm (about 8 ms per step) each loop and a prediction horizon of 1.5 s discretized into $N_p = 15$ steps. An inter-robot distance estimate is required (see Eq. (15)) and we use a constant value of $\hat{d}_{ij} = 2.0$ m for all measurements.

B. Hardware

The quadrotors are based on the Lynxmotion Crazy2fly 34 cm frame with MT2208 1100 kV brushless motors, 12 A ESCs, 8" x 4" plastic propellers, a 3-cell LiPo battery, and have masses of around 1.0 kg. They are controlled by a Pixhawk running PX4v1.10.1 [35] which performs sensor fusion and tracks the commands (thrust, body rate, etc...) sent by the formation controller. The PX4 controller was tuned for reactive body rate control with aggressive PD gains, and the integrator removed to reduce phase shift. The formation controller for a given quadrotor runs on its onboard Raspberry Pi 4B computer with 4GB of RAM running Ubuntu 18.04. The ROS Melodic Mavros package is used



(d) The top plot shows the evolution of the normalized bearing error for E3.1 (blue) and E3.2 (orange). The middle and bottom plots shows the desired formation velocity $\mathbf{v}_{\mathcal{F},y}^d$ (dark green line) and the measured velocities $\mathbf{v}_{i,y}$ (light green lines) expressed in \mathfrak{F}_0 for E3.1 and E3.2, respectively. The bearing error peaks correspond to new desired formations (a)-(c).

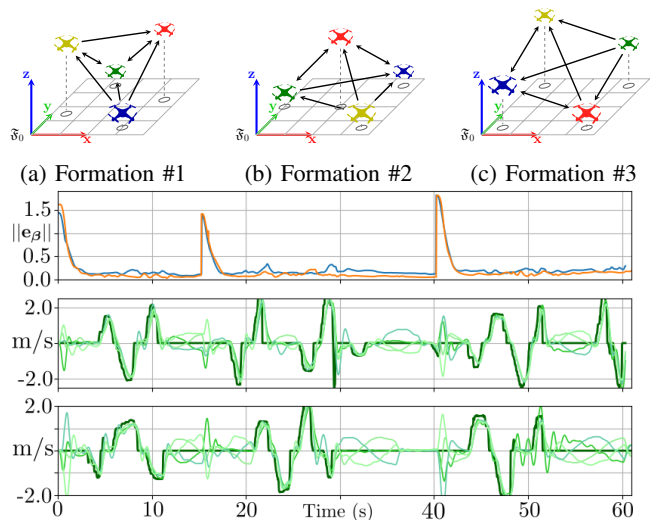
Fig. 6: Two experimental results with the graph of Fig. 4a

to handle message exchanges between the computer and the Pixhawk. Each quadrotor receives desired bearing references from the ground station computer over a 5 GHz wireless network. A Qualisys motion capture (MoCap) system with eight cameras is used to send the pose of the quadrotors to the other quadrotors through the ground station. This information is **only** used for extracting bearing information where needed (using Eq. (7), mimicking an onboard monocular camera), and to generate some nullspace commands to assist the pilot.

C. Experimental Procedures and Results

All experiments began with the quadrotors hovering at pre-specified positions, and the formation controller takes over at time $t = 0$ s as seen in Figures 6-7. As experiments were performed in a small (relative to the formation footprint) $4 \times 6 \times 4$ m flying arena, an autopilot algorithm running at 20 Hz provides the $v_{\mathcal{F}x}$, $v_{\mathcal{F}z}$, and $\dot{s}_{\mathcal{F}}$ trivial motions in \mathfrak{F}_0 to keep the formation centered along the short axes or the arena and at a scale of 1 m. The other trivial motions $v_{\mathcal{F}y}$ and $\dot{\psi}_{\mathcal{F}}$ were given manually at 50 Hz using a joystick.

Two experiments each are shown for the three (E3.1-2) and four (E4.1-2) agent formations in Fig. 4. We see in figures 6-7 that despite differing pilot inputs, the experiments generally share very similar convergence properties. The difference in E3.1-2 for the initial formation shows that due to slight differences in the initial configuration, the two tests followed different convergence paths and were still successful. We can furthermore see that the maximum steering velocity (2.5 ms^{-1} in E3.1 and 5.0 ms^{-1} in E3.2) had some effect on the bearing control, as the peak bearing error for E3.2 at 8 s, 24 s,



(d) The top plot shows the evolution of the normalized bearing error for E4.1 (blue) and E4.2 (orange). The middle and bottom plots shows $\mathbf{v}_{\mathcal{F},y}^d$ (dark green line) and the measured $\mathbf{v}_{i,y}$ (light green lines) expressed in \mathfrak{F}_0 for E4.1 and E4.2, respectively. The bearing error peaks correspond to new desired formations (a)-(c).

Fig. 7: Two experimental results with the graph of Fig. 4b

49 s and 52 s occur directly after the peak velocity times. This is possibly due to propeller saturation that occurs when attempting to re-orient the quadrotor while maintaining high thrust to reach accelerations of $\dot{v}_i > 12 \text{ ms}^{-2}$.

The second pair of experiments show that similar results may be obtained when scaling up the formation and using variable numbers of edges. Experiments E4.1-2 had a one agent with a single bearing measurement, two agents with two bearing measurements, and a single agent with three bearing measurements. The steering was less aggressive (with peak velocities of 2.5 ms^{-1} and peak accelerations of 6 ms^{-2}) as the formation occupied more volume making it harder to pilot in the arena, and there was more aerodynamic interference between quadrotors. We can see in Fig. 7 the yaw rate of the formation (e.g. 30-36 s in E4.2 among others) visible due to the distinctive sine wave induced by $\omega_{z\mathcal{F}}$. A video of the experiments can be found online².

VII. CONCLUSION

This paper showed that model predictive control (MPC) may be applied to bearing formations of quadrotors, improving the transient rate of formation convergence and allowing for high-speed maneuvering of the formation. Extensive simulations have shown that it is able to increase the performance, and possibly the reliability, of bearing-based formations. Future work will include the implementation of MPC formation controllers in constrained environments requiring dynamic maneuvering, and with field of view handling (to which MPC is well suited). Work is underway to integrate real-time visual detection using monocular fish-eye cameras on the quadrotors for the extraction of bearings.

²<https://youtu.be/KCpXjPI6eyU>

REFERENCES

- [1] M. Coppola, K. N. McGuire, C. De Wagter, and G. C. de Croon, "A Survey on Swarming With Micro Air Vehicles: Fundamental Challenges and Constraints," *Frontiers in Robotics and AI*, vol. 7, no. February, 2020.
- [2] R. Olfati-Saber, "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [3] C. W. Reynolds, "Flocks, Herds, and Schools: A Distributed Behavioral Model," *Computer Graphics*, vol. 21, no. 4, 1987.
- [4] K. Vencatasamy, L. Jaulin, and B. Zerr, "Secure a Zone from Intruders with a Group Robots," in *Marine Robotics and Applications*, I. L. Jaulin, A. Caiti, M. Carreras, V. Creuze, F. Plumet, B. Zerr, and A. Billon-Coat, Eds., Springer, 2018, pp. 101–116.
- [5] R. D. Arnold, H. Yamaguchi, and T. Tanaka, "Correction to: Search and rescue with autonomous flying robots through behavior-based cooperative intelligence," *Journal of International Humanitarian Action*, vol. 4, no. 1, 2019.
- [6] J. Kim, S. Bonadies, C. D. Eggleton, and S. A. Gadsden, "Cooperative Robot Exploration Strategy Using an Efficient Backtracking Method for Multiple Robots," *Journal of Mechanisms and Robotics*, vol. 10, no. 6, p. 064502, 2018.
- [7] I. Mademlis, V. Mygdalis, N. Nikolaidis, M. Montagnuolo, F. Negro, A. Messina, and I. Pitas, "High-Level Multiple-UAV Cinematography Tools for Covering Outdoor Events," *IEEE Transactions on Broadcasting*, vol. 65, no. 3, pp. 627–635, 2019.
- [8] M. Fernando and L. Liu, "Formation control and navigation of a quadrotor swarm," *2019 International Conference on Unmanned Aircraft Systems, ICUAS 2019*, pp. 284–291, 2019.
- [9] S. M. Kang and H. S. Ahn, "Design and Realization of Distributed Adaptive Formation Control Law for Multi-Agent Systems with Moving Leader," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, 2016.
- [10] D. Zelazo, A. Franchi, H. H. Bühlhoff, and P. Robuffo Giordano, "Decentralized rigidity maintenance control with range measurements for multi-robot systems," *International Journal of Robotics Research*, vol. 34, no. 1, pp. 105–128, 2015.
- [11] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *International Journal of Control*, vol. 82, no. 3, pp. 423–439, 2009.
- [12] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Sagués, "Vision-Based Distributed Formation Control Without an External Positioning System," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 339–351, 2016.
- [13] R. Tron, J. Thomas, G. Loianno, K. Daniilidis, and V. Kumar, "A distributed optimization framework for localization and formation control," *IEEE Control Systems Magazine*, vol. 36, no. 4, 2016.
- [14] S. Zhao and D. Zelazo, "Bearing Rigidity Theory and its Applications for Control and Estimation of Network Systems: Life Beyond Distance Rigidity," *IEEE Control Systems Magazine*, 2019.
- [15] I. K. Erunsal, R. Ventura, and A. Martinoli, "Nonlinear Model Predictive Control for 3D Formation of Multirotor Micro Aerial Vehicles with Relative Sensing in Local Coordinates," 2019.
- [16] T. T. Ribeiro, A. G. S. Conceição, I. Sa, and P. Corke, "Nonlinear Model Predictive Formation Control for Quadcopters," *IFAC-PapersOnLine*, vol. 48, no. 19, pp. 39–44, 2015.
- [17] M. Saska, V. Vonásek, T. Krajník, and L. Přeučil, "Coordination and navigation of heterogeneous MAV-UGV formations localized by a 'hawk-eye'-like approach under a model predictive control scheme," *International Journal of Robotics Research*, vol. 33, no. 10, pp. 1393–1412, 2014.
- [18] J. Svacha, K. Mohta, M. Watterson, G. Loianno, and V. Kumar, "Inertial Velocity and Attitude Estimation for Quadrotors," *IEEE International Conference on Intelligent Robots and Systems*, pp. 7810–7816, 2018.
- [19] I. Sa and P. Corke, "System Identification, Estimation and Control for a Cost Effective Open-Source Quadcopter," in *2012 IEEE International Conference on Robotics and Automation*, 2012.
- [20] F. Schiano, A. Franchi, D. Zelazo, and P. Robuffo Giordano, "A Rigidity-Based De-centralized Bearing Formation Controller for Groups of Quadrotor UAVs," in *IEEE International Conference on Intelligent Robots and Systems*, Daejeon, South Korea, 2016, pp. 5099–5106.
- [21] D. Six, S. Briot, J. Erskine, and A. Chriette, "Identification of the Propeller Coefficients and Dynamic Parameters of a Hovering Quadrotor from Flight Data," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1063–1070, 2020.
- [22] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential Flatness of Quadrotor Dynamics Subject to Rotor Drag for Accurate Tracking of High-Speed Trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.
- [23] R. Findeisen and F. Allgoewer, "An Introduction to Nonlinear Model Predictive Control," in *21st Benelux Meeting on Systems and Control*, Veldhoven, 2002.
- [24] M. Kamel, T. Stastny, K. Alexis, and R. Siegwart, "Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system," *Studies in Computational Intelligence*, vol. 707, pp. 3–39, 2017.
- [25] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, "Fast nonlinear Model Predictive Control for unified trajectory optimization and tracking," in *IEEE International Conference on Robotics and Automation*, 2016.
- [26] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "PAMPC: Perception-Aware Model Predictive Control for Quadrotors," in *IEEE International Conference on Intelligent Robots and Systems*, IEEE, Ed., Madrid, Spain, 2018.
- [27] F. Schiano and R. Tron, "The Dynamic Bearing Observability Matrix Nonlinear Observability and Estimation for Multi-Agent Systems," pp. 1–8, 2018.
- [28] R. Spica and P. R. Giordano, "Active decentralized scale estimation for bearing-based localization," *IEEE International Conference on Intelligent Robots and Systems*, pp. 5084–5091, 2016.
- [29] G. Allibert, E. Courtial, and F. Chaumette, "Predictive control for constrained image-based visual servoing," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 933–939, 2010.
- [30] F. Schiano and P. R. Giordano, "Bearing rigidity maintenance for formations of quadrotor UAVs," in *Proceedings - IEEE International Conference on Robotics and Automation*, Singapore, 2017, pp. 1467–1474.
- [31] N. Lissandrini, G. Michieletto, R. Antonello, M. Galvan, A. Franco, and A. Cenedese, "Cooperative optimization of UAVs formation visual tracking," *Robotics*, vol. 8, no. 3, 2019.
- [32] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-Based Reactive Planning for Aggressive Target Tracking while Avoiding Collisions and Occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.
- [33] D. Brescianini, M. Hehn, and R. D'Andrea, "Nonlinear Quadcopter Attitude Control," ETH Zurich, Tech. Rep. 2, 2013.
- [34] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit - An open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [35] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6235–6240, 2015.