

Autonomous Helicopter Formation using Model Predictive Control

Hoam Chung* and S. Shankar Sastry†

University of California, Berkeley, California, 94720, USA

Formation flight is the primary movement technique for teams of helicopters. However, the potential for accidents is greatly increased when helicopter teams are required to fly in tight formations and under harsh conditions. The starting point for safe autonomous flight formations is to design a distributed control law attenuating external disturbances coming into a formation, so that each vehicle can safely maintain sufficient space between it and all other vehicles. In order to avoid the conservative nature inherent in distributed MPC algorithms, we begin by designing a stable MPC for individual vehicles, and then introducing carefully designed inter-agent coupling terms in each performance index. The proposed algorithm works in a decentralized manner, and is applied to the problem of helicopter formations comprised of heterogeneous vehicles. The disturbance attenuation property of the proposed MPC controller is validated throughout a series of computer simulations.

I. Introduction

ROTORCRAFT have revolutionized the offensive, defensive, reconnaissance, and security operations in the battlefield due to their mobility, range, and versatility (including vertical take-off and landing (VTOL) capability). With recent advances in technology, such as aerial refuelling and night vision, helicopters have taken on increasingly important roles in military operations. Formation flight is the primary movement technique for helicopter teams.¹ By maintaining a coordinated formation, it is possible to achieve flight integrity with less fuel consumption than an unstructured flight, increasing the possibility of a mission's success.

Even with such unique flight capabilities, helicopter teams are confronted by very challenging situations. The potential for accidents is increased by requirements to fly in close formation and under harsh conditions including poor weather and extremely low altitudes. The effects of battlefield stress exerted on an aircrew increase dramatically under these adverse circumstances. We propose that computer-assisted autonomous formation flight procedures can be implemented to help to diminish battlefield stress.

Even though helicopter formation flight is of critical importance in various operations, little research has been done on this topic. Since helicopter dynamics are notoriously complex and uncertain, until recently it had not been feasible to design an automatic controller for a single helicopter. However, recent advances in system identification techniques and control of rotorcraft-based unmanned aerial vehicles (RUAVs)^{2,3} have provided insight into autonomous helicopter formation flight. Although several researchers have made efforts on the stable helicopter formation,^{4,5} their applications have been restricted to homogeneous formations in which all the vehicles have identical dynamics.

Model Predictive Control (MPC), also known as moving horizon or Receding Horizon Control (RHC), has been a useful technique for the control of slow dynamic systems such as chemical processes because the scheme requires high computational speed of the control hardware due to its on-line nature. Recently, the rapid development of digital processors, and powerful and inexpensive controllers make it possible to adopt MPC into hard real-time applications.⁶

MPC can provide a better performance in controlling uncertain plants since it can update the gain of the controller based on the current states, whereas fixed-gain control algorithms cannot.⁷ The capability to

*PhD Candidate, Mechanical Engineering, University of California, Berkeley hachung@eecs.berkeley.edu

†Professor, Electrical Engineering and Computer Science, University of California, Berkeley sastry@eecs.berkeley.edu

manipulate the state-dependency of the control weighting matrices and constraints in real-time is the key feature of a model predictive control algorithm. There are excellent survey papers describing the development of MPC theories. See Ref. 8 and 9 for example.

As long as an MPC algorithm is applied to a formation flight problem, a centralized approach is not a feasible choice at all, since it is not scalable from the viewpoint of computation and communication.¹⁰ Thus it is natural to consider a decentralized approach to the formation flight problem. However, with a decentralized MPC scheme, it is recognized that the stability proof becomes very difficult.

Under the assumption that the dynamics of each vehicle are decoupled, a major obstacle in proving the stability of a decentralized MPC scheme arises in predicting neighbors' behaviors over the future horizon. Without considering inter-vehicle constraints, the coupling between agents appears in the performance index as a penalty on relative gap errors. If there is no appropriate predictions of the behaviors of neighbors, it is difficult to set bounds on them. In an attempt to resolve this difficulty, authors of Ref. 10 introduced so called 'compatibility constraints', which restrict the future variations of neighbors' optimal inputs from the previous optimal ones. Using this, it can be proved that the closed-loop states converge to the neighborhood of origin. However, due to the nature of this constraint, once an open-loop control is computed and applied to the system on the current sampling time, the control at the next sampling time is constrained by the previous open-loop control. In nominal situations (no model errors without exogenous disturbances), this may not be a problem, since the open-loop control will predict the system behaviors exactly and the system may stay on the optimal trajectory. However, if the system trajectory starts to deviate from the initial optimal trajectory because of disturbances or model errors, this constraint would limit the effect of feedback, and the robust nature of the feedback system might be lost. Before the stability is affected by uncertainties, the algorithm may have trouble maintaining the feasibility of the optimization problem. Furthermore, since this algorithm is applicable only to (nonlinear) double integrator systems, it is impossible to use this algorithm for formation flight of helicopters, which have unmeasurable hidden state variables related to flapping and stabilizer bar dynamics.

In Ref.11, researchers used the hierarchical decomposition method, which decomposes the original formation graph into overlapping subgraphs with different hierarchy levels. Under this decomposition, the algorithm allows a vehicle at a node with high priority to compute control laws for vehicles with lower priorities, and transmit them to vehicles with lower priorities, assuming one time step communication delay. By doing this, since future behaviors of neighboring vehicles with lower priorities are completely known to vehicles with higher priorities, 'prediction' is no longer needed, and stability can be proved by standard Lyapunov arguments. In theory, this method provides a simple and clear way to prove the stability of a decentralized MPC scheme, minimizes the conservatism and required communication bandwidth. However, in this case, the integrity of the system structure is totally dependent on the communication link, which can be deteriorated easily in battlefields.

Instead of sticking to proving stability of a decentralized MPC, our focus is on designing an MPC-based velocity tracking controller with penalties on relative gap errors, and study the propagation of external disturbances through the formation.

In Section II, helicopter dynamics and kinematics are reviewed. Then, we suggest a carefully designed strategy of defining relative gap errors between neighboring vehicles in Section III-A. Before introducing inter-agent coupling terms into the full scale problem, a stable MPC controller for a single vehicle is designed in Section III-B. Simulation results are shown in Section IV, which is followed by conclusions and future work.

II. Helicopter Dynamics

Since the helicopter dynamics, which can be derived using Newton's law, are represented in the body coordinates system fixed to the center of the mass of a helicopter,³ the kinematic equations between the body coordinates and the spatial coordinates^a are required. The kinematics are further divided into two parts: (1) the position describing translational motion in the spatial coordinates, and (2) the Euler angles describing the vehicle's attitude and heading in the spatial coordinates. It should be pointed out that, even though the dynamics are assumed to be linear at a certain operating point, the entire system of equations becomes nonlinear, since the kinematics involve nonlinear transformations. In hover mode, it is possible to linearize the kinematics and keep the entire system linear. However, in the cruise flight mode, due to the

^aThroughout this paper, the spatial coordinates mean the tangent-plane coordinate system, whose origin is located at a certain point of the earth's surface.

non-zero pitch trim, the validity of the linearized kinematics model may deteriorate easily.

A. Basic Helicopter Dynamics and Kinematics

The following definitions of helicopter dynamics and kinematics are based on Ref. 3 and 12 with slight modifications. The positive directions of x , y , and z axes of the spatial coordinates align, respectively, to the north, east and downward directions. For detailed derivations of the helicopter dynamics, see Ref. 3 and 13.

As mentioned earlier, the overall system dynamics are divided into the kinematics and the system-specific dynamics denoted by superscripts K and D . The state vectors and the control input vector are defined as follows:

$$\mathbf{x}^D = [u \ v \ p \ q \ a_{1s} \ b_{1s} \ w \ r \ r_{fb} \ c \ d]^T \quad (1)$$

$$\mathbf{x}^A = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \mathbf{x}^K = \begin{bmatrix} \mathbf{x}^A \\ \mathbf{x}^S \end{bmatrix}, \quad (2)$$

$$\mathbf{u} = [u_{a1s} \ u_{a1s} \ u_{\theta_M} \ u_{ref}]^T, \quad (3)$$

where

u, v, w : trimmed translational velocities in body coordinates

p, q, r : roll, pitch and yaw rates in the body coordinates

ϕ, θ, ψ : roll, pitch, and yaw in ZYX Euler angle notation in the spatial coordinates

a_{1s}, b_{1s} : longitudinal and lateral flapping angles of the main rotor

c, d : longitudinal and lateral flapping angles of the Bell-Hiller stabilizer bar

r_{fb} : internal state of yaw rate feedback gyro

u_{a1s}, u_{a1s} : inputs to the lateral and longitudinal cyclic pitch

u_{θ_M} : input to the main rotor collective pitch

u_{ref} : reference yaw rate input to the gyro

Let the superscripts S and B denote spatial and body coordinates. \mathbf{x}^S and \mathbf{x}^B denote the position in the spatial coordinates and in the body coordinates, respectively^b.

The kinematics part can be defined as follows:

$$\dot{\mathbf{x}}^S = R^{B \rightarrow S} \dot{\mathbf{x}}^B, \quad \dot{\mathbf{x}}^A = R_{\omega}^{B \rightarrow S} \omega, \quad (4)$$

where $\mathbf{x}^S = [x^S \ y^S \ z^S]^T$, and $\omega = [p \ q \ r]^T$. $R^{B \rightarrow S}(\mathbf{x}^A)$, the rotation matrix from the body to the spatial coordinates, and $R_{\omega}^{B \rightarrow S}(\mathbf{x}^A)$ is the relationship matrix between angular rates in the body and in the spatial coordinates. They are defined by^{14,15}

$$R^{B \rightarrow S} = \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \quad (5)$$

$$R_{\omega}^{B \rightarrow S} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix}. \quad (6)$$

The dynamics part can be written as

$$\dot{\mathbf{x}}^D = f^D(\mathbf{x}^D(t), \mathbf{x}^A(t), \mathbf{u}(t)). \quad (7)$$

^bSubscript vehicle indices are suppressed until the next section for simplicity

Finally, the entire system equation becomes

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x}^D \\ \mathbf{x}^A \\ \mathbf{x}^S \end{bmatrix} = \begin{bmatrix} f^D(\mathbf{x}^D(t), \mathbf{x}^A(t), \mathbf{u}(t)) \\ R_\omega^{B \rightarrow S} \omega \\ R^{B \rightarrow S} \dot{\mathbf{x}}^B \end{bmatrix} = f(\mathbf{x}^S(t), \mathbf{x}^D(t), \mathbf{x}^A(t), \mathbf{u}(t)), \quad (8)$$

or simply,

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \quad \text{with} \quad \mathbf{x} \triangleq \begin{bmatrix} \mathbf{x}^D \\ \mathbf{x}^A \\ \mathbf{x}^S \end{bmatrix}. \quad (9)$$

B. Helicopter Cruise Model

In Ref. 13, a linear cruise flight model of the Yamaha R-50 industrial helicopter trimmed at $u_0 = 49$ ft/sec, $w_0 = 11.2$ ft/sec, and $v_0 = 0$ is introduced and all the coefficients in the dynamics equation are identified through test flights and system identification techniques. If we convert this trim condition into spatial coordinates, then it becomes 30 mi/h forward cruise speed with the pitch trim $\theta_0 = -0.22$ (rad).

In order to use the linear cruise model, we need to define several relationships between spatial variables and variables in the linear dynamics. First, the velocities in the body-fixed frame can be represented by

$$\dot{x}^B = u_0 + u, \quad \dot{y}^B = v_0 + v, \quad \text{and} \quad \dot{z}^B = w_0 + w. \quad (10)$$

Next, the trimmed pitch angle $\bar{\theta}$ can be defined as

$$\bar{\theta} = \theta - \theta_0. \quad (11)$$

From these relationships, the kinematics equations (Eq.(4)) are now well defined. The dynamics can be represented by

$$\dot{\mathbf{x}}^D(t) = f^D(\mathbf{x}^D(t), \mathbf{x}^A(t), \mathbf{u}(t)) = A\mathbf{x}^l(t) + B\mathbf{u}(t), \quad (12)$$

where $A \in \mathbb{R}^{11 \times 13}$, $B \in \mathbb{R}^{11 \times 4}$, and

$$\mathbf{x}^l = \left[u \quad v \quad p \quad q \quad a_{1s} \quad b_{1s} \quad w \quad r \quad r_{fb} \quad c \quad d \quad \phi \quad \bar{\theta} \right]^T. \quad (13)$$

C. Scaling Based on Froude Number

In order to test the algorithm we propose in the next section with a heterogeneous helicopter team, we need to generate a model that is different from our existing Yamaha R-50 model. Since it is extremely difficult to perform an identification flight in a cruise condition without a wind tunnel facility, we used the scaling technique presented by Mettler.¹³

The Froude number is the ratio of inertia to gravitational forces. If two different models have Froude numbers that are close each other, it means that two systems have similar dynamic properties. The number is defined by

$$F = \frac{V^2}{Lg}, \quad (14)$$

where V is the characteristic velocity, L is the characteristic length, and g is gravitational acceleration. In helicopter dynamics, the rotor tip velocity and the rotor radius are used as V and L respectively. The Table 1 shows the comparison of the Froude numbers of Yamaha R-50 and Robinson R22, and they are very close. We have created a virtual model in the region between the Robinson R22 and Yamaha R-50, which has two times the rotor diameter of the R-50 and has a similar Froude number. The scale N refers to a model helicopter with $1/N$ the rotor diameter of the Yamaha R-50. It should be noted that the relationship of the Froude number imposes a relation between time scales,¹³

$$\text{second} \approx \frac{1}{\sqrt{N}}. \quad (15)$$

Based on these comparisons, the scaling of coefficients in A and B (Eq.(12)) can be done using dimensional analysis. The created virtual model is less agile than the original R50 model due to its larger rotor radius and lower rotor speed. Refer to Ref. 18 for the comparison of dynamic properties of two models.

Table 1. Comparison of Froude numbers

	Rotor Radius (ft)	Rotor Speed (rad/s)	N	Froude Number
R22	13	53	0.38	1134
Virtual Model	10	62	0.5	1194
R-50	5	89	1	1230

III. MPC-Based Helicopter Formation Flight

A. Formation Topology and Definitions of Gap Errors

In the following discussion, we examine formations where each agent in the formation has connections that are less than or equal to two. Although cases where one or more agents in the formation has more than three connections can still be accounted for, these are considered as special cases, and are not described here. In real-world operations, most helicopter formations fall into the category with maximum two bidirectional connections on each agent.¹ Some publications^{11,16} on the vehicle formations using distributed MPC algorithms consider arbitrary formation shapes represented by connected graphs. However, an arbitrary formation shape is obviously not used in high-speed cruise formations, especially for manned helicopters. We believe that the research on arbitrarily coupled vehicle formation needs to be developed in the context of behaviors of a ‘swarm’ of unmanned vehicles.¹⁷

Most vehicle formation algorithms^{4,5,11,16} use a so-called ‘constant gap’ strategy as described in Figure 1(a). $l_{i-1,i}^r \in \mathbb{R}^3$ denotes the constant relative gap vector from $i-1$ -th vehicle to i -th vehicle in the reference coordinates, which is represented by $x^r - y^r$ (tangential-normal to the reference velocity V_r^S) in the figure. Note that we need the relationship

$$l_{i-1,i}^r = -l_{i,i-1}^r \quad (16)$$

for the unique definition of the formation shape. Also, we can obtain the $l_{i-1,i}^S$ using the rotation matrix $R^{r \rightarrow S}(\psi_r)$ such that

$$l_{i-1,i}^S = R^{r \rightarrow S}(\psi_r) l_{i-1,i}^r. \quad (17)$$

As shown in Figure 1(a), the gap error can be defined as

$$\begin{aligned} \mathbf{e}_{i,i-1}^S &= \mathbf{x}_{i-1}^S - \mathbf{x}_i^S + l_{i-1,i}^S \\ &= \mathbf{x}_{i-1}^S - \mathbf{x}_i^S - l_{i,i-1}^S \end{aligned} \quad (18)$$

The other type of the gap strategy is called a ‘varying gap’ strategy (Figure 1(b)). In this strategy, the i -th vehicle considers the middle point in the line connecting $i-1$ -th vehicle to $i+1$ -th vehicle as the reference point. The error vector becomes

$$\mathbf{e}_i^S = \frac{\mathbf{x}_{i-1}^S + \mathbf{x}_{i+1}^S}{2} - \mathbf{x}_i^S. \quad (19)$$

For vehicles in edges, the constant gap strategy (Eq.(18)) is used, although other vehicles use the varying gap strategy. By using the constant strategy in edges, one can show that the sum of squares of gap errors is zero if and only if all the gap errors are zero, even if we use the varying gap strategy. In the viewpoint of exogenous disturbance attenuation, the varying gap strategy (with constant gap strategy in edges) shows superior performance to the constant gap strategy.¹⁸ Therefore, we use this strategy throughout this paper.

In order to realize real-world helicopter formations like Vee, wedge, left and right echelon, and left and right staggered formations,¹ we need to consider a gap error definition for a vehicle that has followers in left and right sides. In this case, the gap error vector definition becomes more complex, but they can be defined in similar manner. See Ref. 18 for details.

B. Model Predictive Control Law for Helicopter Formations

Recall the system equation of the i -th vehicle

$$\dot{\mathbf{x}}_i(t) = f_i(\mathbf{x}_i(t), \mathbf{u}_i(t)), \quad (20)$$

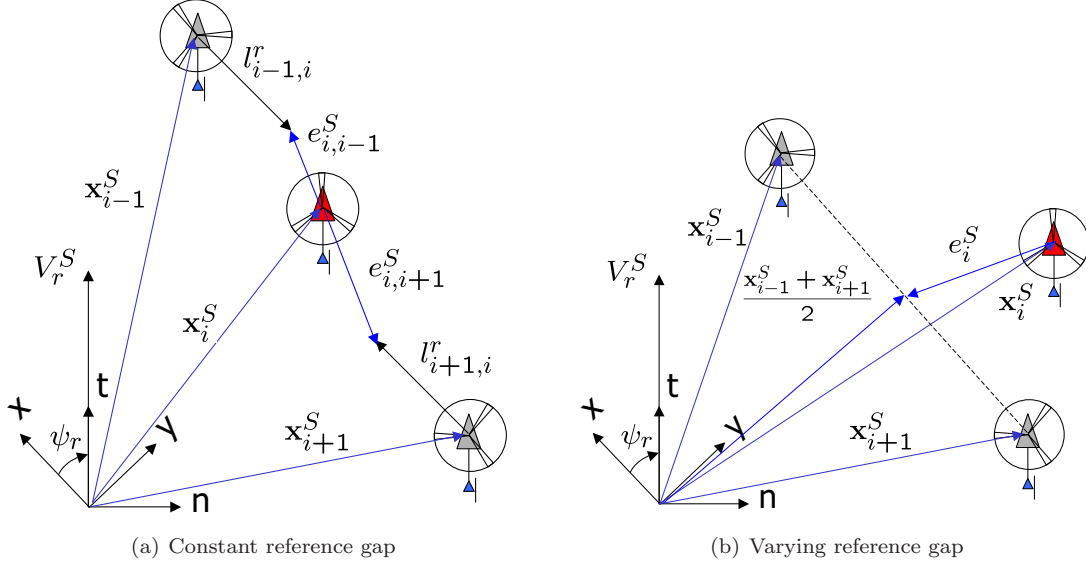


Figure 1. Gap error vector definitions

where

$$\mathbf{x}_i \in \mathcal{X}_i \subset \mathbb{R}^{n_x}, \mathbf{u}_i \in \mathcal{U}_i \subset \mathbb{R}^{n_u}. \quad (21)$$

Here \mathcal{X}_i and \mathcal{U}_i are convex sets. We assume that $\mathbf{u}_i(t)$ is measurable, and $f_i : \mathcal{X}_i \times \mathcal{U}_i \rightarrow \mathbb{R}^{n_x}$ satisfies standard conditions for admitting a unique solution.

The performance index $J_i(\cdot)$ has the form of

$$J_i(\bar{\mathbf{x}}_i, \kappa_i, t) = \int_{\tau}^{\tau+L} \mathcal{L}_i(\mathbf{x}_i(t), \mathbf{x}_{-i}^S(t), \mathbf{u}_i(t), \mathbf{y}_i^{S,r}(t)) dt + V_i^f(\mathbf{x}_i(\tau+L), \mathbf{x}_{-i}^S(\tau+L), \mathbf{y}_i^S(\tau+L)), \quad (22)$$

where the terminal penalty function $V_i^f(\cdot) : \mathcal{X}_i \times \mathcal{X}_{-i} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}$ is positive definite. The subscript $-i$ represents indices of neighboring vehicles following the notation of Ref. 10. $\mathbf{y}_i^{S,r} : \mathbb{R} \rightarrow \mathbb{R}^{n_y}$ is the reference vector in the spatial frame, which has (at least) the reference velocity vector $\dot{\mathbf{x}}_i^{S,r}(t)$ and the reference heading $\psi_i^r(t)$.

The finite horizon optimal control (FHOC) problem with initial condition $\bar{\mathbf{x}}_i = \mathbf{x}_i(\tau)$ and horizon length L is defined as

$$V_i(\bar{\mathbf{x}}_i, t) = \min_{\kappa} J_i(\bar{\mathbf{x}}_i, \kappa_i, t), \quad (23)$$

which is subject to Eq.(20), and Eq.(21).

κ_i is a piecewise continuous time-dependent function in open-loop strategy space such that

$$\kappa_i \in \mathcal{K}_i = \{\kappa : [0, L] \times \mathcal{X}_i \rightarrow \mathcal{U}_i\} \quad (24)$$

$$\mathbf{u}_i(t) = \kappa_i(t - \tau, \bar{\mathbf{x}}_i). \quad (25)$$

If an optimal solution of the FHOC problem exists, let $\kappa_i^*(t - \tau, \bar{\mathbf{x}}_i)$ denote the solution for $t \in [\tau, \tau + L]$. Note that $V_i(\bar{\mathbf{x}}_i, t) = J(\bar{\mathbf{x}}_i, \kappa_i^*, t)$.

Based on these, the receding horizon control law for i -th vehicle at $t = \tau$ is defined as

$$\mathbf{u}_i(\tau) = \kappa_i^{RH}(\bar{\mathbf{x}}_i) = \kappa_i^*(0, \bar{\mathbf{x}}_i). \quad (26)$$

The running cost $\mathcal{L}_i(\cdot)$ has the form of

$$\mathcal{L}_i(\mathbf{x}_i, \mathbf{x}_{-i}^S, \mathbf{u}_i, \mathbf{y}_i^S) = \mathcal{L}_i^{gap}(\mathbf{x}_i, \mathbf{x}_{-i}^S) + \mathcal{L}_i^y(\mathbf{x}_i, \mathbf{y}_i^S) + \mathcal{L}_i^x(\mathbf{x}_i) + \mathcal{L}_i^u(\mathbf{u}_i). \quad (27)$$

For vehicles in edges,

$$\mathcal{L}_i^{gap}(\mathbf{x}_i, \mathbf{x}_{-i}^S) = \|\mathbf{e}_{i,-i}^S\|_{Q_{i,-i}^e}, \quad (28)$$

where $\|x\|_Q$ denotes a matrix weighted norm ($x^T Q x$). Similarly, using the varying gap strategy, the term can be represented as

$$\mathcal{L}_i^{gap}(\mathbf{x}_i, \mathbf{x}_{-i}^S) = \|\mathbf{e}_i^S\|_{Q_i^e}. \quad (29)$$

$\mathcal{L}_i^y(\cdot)$ penalizes the tracking error, and can be defined as

$$\mathcal{L}_i^y(\mathbf{x}_i, \mathbf{y}_i^S(t)) = \|\mathbf{y}_i^S(t) - C_i^y(\mathbf{x}_i)\|_{Q_i^y}, \quad (30)$$

where $C_i^y: \mathcal{X}_i \rightarrow \mathbb{R}^{n_i^y}$ maps the state vector into corresponding output signals. In order to track the reference velocity vector and heading in the spatial coordinate system, we use the following definition of $C_i^y(\mathbf{x}_i)$ in simulation:

$$C_i^y(\mathbf{x}_i) = \begin{bmatrix} R^{B \rightarrow S} \dot{\mathbf{x}}_i^B \\ \psi_i \end{bmatrix} \quad (31)$$

The term $\mathcal{L}_i^x(\cdot)$ is for remaining terms in the state vector that do not appear in the previous running costs, ϕ , θ , p , q and r , for example. It is noticeable that internal states, a , b , r_{fb} , c , and d , are not penalized, since they are not measurable, and related dynamics are well damped.³ Therefore, we can define $\mathcal{L}_i^x(\cdot)$ such that

$$\mathcal{L}_i^x(\mathbf{x}_i) = \|C_i^x \mathbf{x}_i\|_{Q_i^x}, \quad (32)$$

where $C_i^x \in \mathbb{R}^{n_i^{x'} \times n_i^x}$, and $n_i^{x'}$ denotes the number of terms in the state penalized by $\mathcal{L}_i^x(\cdot)$. $\mathcal{L}_i^u(\cdot)$ penalizes input magnitudes,

$$\mathcal{L}_i^u(\mathbf{u}_i) = \|\mathbf{u}_i\|_{R_i}, \quad (33)$$

with positive definite matrix $R_i \in \mathbb{R}^{n_i^u \times n_i^u}$.

Finally, the terminal penalty $V_i^f(\cdot)$ is defined by

$$V_i^f(\mathbf{x}_i, \mathbf{x}_{-i}^S, \mathbf{y}_i^S) = \begin{cases} \left\| \begin{bmatrix} \mathbf{e}_{i,-i}^S \\ \mathbf{y}_i^S - C_i^y(\mathbf{x}_i) \\ C_i^x \mathbf{x}_i \end{bmatrix} \right\|_{P_i} & \text{vehicles in edges} \\ \left\| \begin{bmatrix} \mathbf{e}_i^S \\ \mathbf{y}_i^S - C_i^y(\mathbf{x}_i) \\ C_i^x \mathbf{x}_i \end{bmatrix} \right\|_{P_i} & \text{otherwise} \end{cases}, \quad (34)$$

where $P_i \in \mathbb{R}^{(n^g + n_i^y + n_i^{x'}) \times (n^g + n_i^y + n_i^{x'})}$ is a positive definite matrix. This terminal penalty plays several important roles in achieving guaranteed stability. In the next section, we discuss a design procedure of an MPC without time dependent terms $\mathcal{L}^{gap}(\cdot)$ and $\mathcal{L}^y(\cdot)$.

C. Design of Nonlinear MPC without Inter-agent Couplings

Due to pioneering research efforts in the 1990s, several design methodologies for stable MPC algorithms are now available.⁸ In most of the stability proofs of MPC algorithms, the ‘tail’ of the value function of an FHO problem plays a very important role, since it is known that, if one can approximate this term properly, the MPC based on FHO problem realizes the virtues of infinite-horizon problems in stability and robustness. The work of Chen and Allgöwer¹⁹ achieves this by taking advantage of the terminal inequality constraint and (virtual) terminal linear controller. Their technique for proving stability is one of the well-regarded among various finite-horizon based online optimization controllers, including a decentralized MPC algorithm.¹⁰ However, it is reported that by introducing a terminal inequality constraint in the FHO problem the numerical computation becomes slow,²⁰ and sometimes the MPC structure nonrobust.²¹

On the other hand, Jadbabaie et al.²² achieved a stable MPC algorithm by using the so-called control Lyapunov function (CLF) as a terminal cost without any terminal constraints. In this case, even though it is not easy to find a proper CLF for a given nonlinear system without conservatism, the scheme effectively minimizes the number of constraints subject to FHO problem, which is quite important in practical implementation of an MPC algorithm.

In the remaining section, we describe the procedure to define a CLF for the nonlinear helicopter cruise model without inter-agent coupling and time dependent terms using semi-definite programming, which appears in Ref. 20.

First, we need to redefine helicopter cruise dynamics (Eq.(12)) without the spatial position vector part as follows

$$\frac{d}{dt} \begin{bmatrix} \mathbf{x}^D \\ \mathbf{x}^A \end{bmatrix} = \begin{bmatrix} A^D & A^A \\ A^{R_\omega^{B \rightarrow S}}(\mathbf{x}^A) & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}^D \\ \mathbf{x}^A \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} \mathbf{u}, \quad (35)$$

where the matrix A in Eq.(12) is separated into A^A and A^D , and $A^{R_\omega^{B \rightarrow S}}(\mathbf{x}^A)$ is rearranged version of $R_\omega^{B \rightarrow S}$ (Eq.(6)) in corresponding order and dimension.

If we set upper and lower limits of ϕ and θ , then we can get bounds of those terms in $A^{R_\omega^{B \rightarrow S}}(\mathbf{x}^A)$. The operational limits of attitude variables are set to

$$-30^\circ \leq \phi \leq 30^\circ, \quad -40^\circ \leq \theta \leq 20^\circ. \quad (36)$$

The corresponding bounds of terms in $A^{R_\omega^{B \rightarrow S}}(\mathbf{x}^A)$ are

$$\begin{aligned} -0.4195 &\leq \sin \phi \tan \theta \triangleq p_1 \leq 0.4195 \\ -0.8391 &\leq \cos \phi \tan \theta \triangleq p_2 \leq 0.3640 \\ 0.8660 &\leq \cos \phi \triangleq p_3 \leq 1 \\ -0.5 &\leq -\sin \phi \triangleq p_4 \leq 0.5 \\ -0.6527 &\leq \sin \phi \sec \theta \triangleq p_5 \leq 0.6527 \\ 0.8660 &\leq \cos \phi \sec \theta \triangleq p_6 \leq 1.3054 \end{aligned}$$

Now we are ready to convert the system matrix in Eq.(35) into an affine parameter varying matrix²³ such that

$$A(p(t)) = A_0^p + \sum_{i=1}^6 p_i(t) A_i^p, \quad (37)$$

where A_i^p is a constant matrix that has only one 1 on the corresponding entry, and zeros otherwise. A_0^p is the matrix that has constant terms in the system matrix of Eq.(35). Finally, the above parameter varying matrix can be represented by a polytopic model

$$A(t) \in \text{Co}\{A_1^v, A_2^v, \dots, A_{n_v}^v\}, \quad (38)$$

where the set $\text{Co}\{\cdot\}$ denotes the set that includes all possible convex combination of its vertex elements. The conversion from Eq.(37) to Eq.(38) can be done by the function `aff2pol` in LMI Toolbox,²³ and it results in a polytopic model with 64 vertices.

For a given weighting matrices (only for internal states, heading, and attitude variables extracted from Eq.(27)), the minimum upper bound of the value function is the optimal value of the following convex optimization problem:²⁰

$$\min \text{tr}(Z) \quad (39)$$

$$\begin{aligned} & Y > 0 \\ & \begin{bmatrix} Y A_i^{vT} + A_i^v Y - B X - X^T B^T & Y Q^{1/2} & X^T R^{1/2} \\ Q^{1/2} Y & -I & 0 \\ R^{1/2} X & 0 & -I \end{bmatrix} < 0 \\ & \begin{bmatrix} Z & I \\ I & Y \end{bmatrix} > 0 \\ & Y^T = Y, \quad Z^T = Z, \end{aligned} \quad (40)$$

where Z is a slack variable, $Y = P^{-1}$, and $X = KY$ are the change of variables. From this result, the terminal cost (for the system without external time-dependent signals) is defined by $V^f(\mathbf{x}) = \mathbf{x}^T P \mathbf{x}$.

Due to the size of the given cruise dynamics, the dimension of the above convex optimization problem is prohibitively huge. The problem has $2(n^x)^2 - (n^x - 1)(n^x - 2) + n^x n^i$ (292 in our case) variables and $64+1+1$ LMI constraints whose dimensions are $(2n^x + n^u)^2$, $(2n^x)^2$, and $(n^x)^2$, respectively. Most of contemporary personal computers are based on a 32 bit structure, and their maximum allowable memory block size is limited to four giga bytes, which is marginal for our problem. Using LMITOOL²⁴ and SeDuMi,²⁵ we tried to solve the full-scale problem, but a solution was not complete even after 96 hours. Instead, we sampled 32 vertices from 64 vertices in the definition of the polytopic model, and used them for CLF computation. In this case, the solver converges after 15 hours. The magnitudes of elements in the obtained matrix are shown in Figure 2.

For the given CLF $V^f(\mathbf{x})$, it is known that there exist $r \in \mathbb{R}_+$ such that

$$\min_{\mathbf{u}} (\dot{V}^f(\mathbf{x}) + \mathbf{x}^T Q \mathbf{x} + \mathbf{u}^T R \mathbf{u}) \leq 0 \quad \text{for } \mathbf{x} \in \Omega_r, \quad (41)$$

where $\Omega_r = \{\mathbf{x} \in \mathbb{R}^{n_D+n_A} | V^f(\mathbf{x}) \leq r\}$, and n_D and n_A are dimensions of \mathbf{x}^D and \mathbf{x}^A , respectively.

Let $\mathbf{x}^*(t; \bar{\mathbf{x}})$ be the optimal trajectory at $t \in [\tau, \tau + L]$ starting from $\bar{\mathbf{x}} = \mathbf{x}(\tau)$. If $\mathbf{x}^*(\tau + L; \bar{\mathbf{x}}) \in \Omega_r$, then the trajectory starting from $\bar{\mathbf{x}}$ converges to the origin under the RHC scheme.²²

To complete the design procedure, we need to choose a proper horizon length L so that we can have a sufficiently large invariant set that includes Ω_r . However, in high-dimensional systems, it is very difficult to compute an invariant set corresponding to an L analytically or numerically. In our research, we set $L = 0.5$ (s) based on several simulation results with various values of L . The sampling frequency is set to 50Hz ($\delta = 0.02$ sec) so that it is identical with that of our existing UAV control system.³

In order to show the validity of the CLF computed from the sub-sampled polytopic set, computer simulations are performed with several initial conditions. As shown in Figure 3(a) – 3(c), the designed MPC scheme successfully stabilizes all the initial conditions in Table 2.

It is noticeable that all the states converge to the origin in spite of control input saturations in the beginning of simulations (Figure 3(c)). Even though the design procedures we use here are originally for unconstrained MPC, the controller works well with input constraints in our case.

	$\phi(0)$ (degree)	$\theta(0)$ (degree)
IC1	20	30
IC2	20	-30
IC3	-40	30
IC4	-40	-30

Table 2. Initial conditions used for single helicopter simulations with the designed MPC controller

D. Interagent Information Structure and Communication

Provided that the inter-agent communication happens only one time per sampling instance,^c a decentralized algorithm can be implemented with lower bandwidth communication channels, whereas a centralized setup requires high bandwidth communication channel on the central agent which solves the optimal control problem for every agent. However, in a decentralized setup, if inter-agent communications are required during a numerical iteration, the total amount of information transferred between agents can be more than that of a centralized case.²⁶ This scheme falls into a category of algorithms using *cooperative iteration*.²⁷ Since this

^cThis means that there is no communication while solving a local optimization problem.

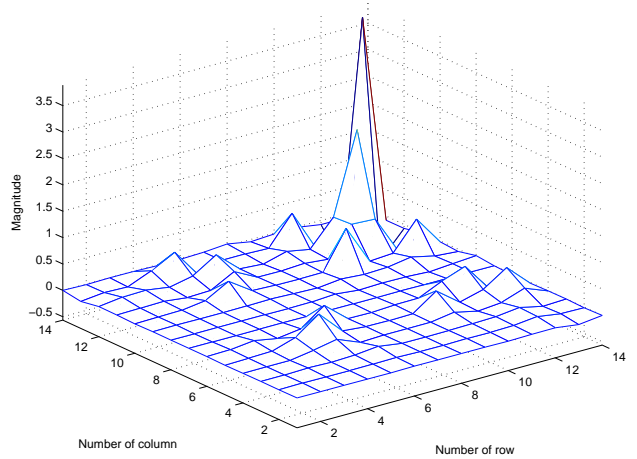


Figure 2. Magnitudes of elements in matrix P

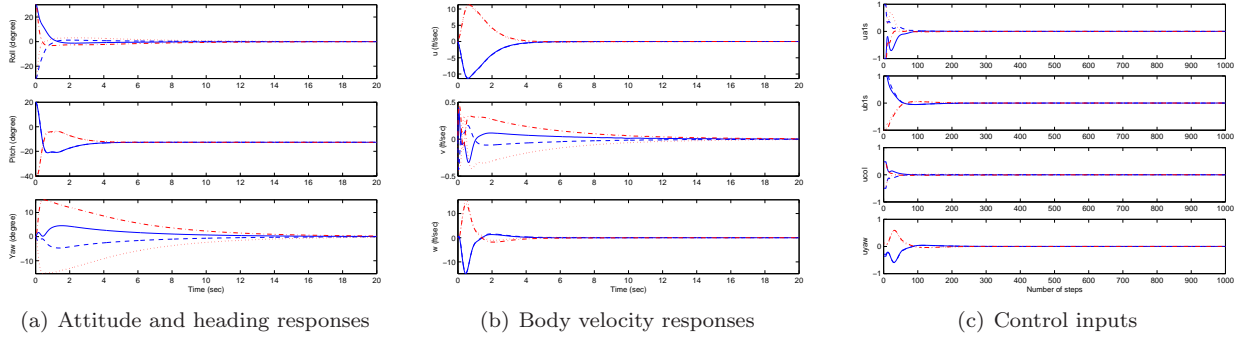


Figure 3. Regulation performance of the designed MPC controller, IC1: blue solid, IC2: blue dashed, IC3: red dash-dotted, IC4: red dotted

cooperative iteration needs stable and high bandwidth communication channels and strict synchronization between agents, it is more challenging to implement one on a hard real-time system. In the proposed setup, we use one inter-agent communication per every sampling instance, and there is no communication while solving an FHOC problem.

As shown in Eq.(27), the proposed MPC scheme requires the predicted trajectory of neighbors for $t \in [\tau, \tau + L]$ at every sampling time. In Ref. 10, neighboring agents interchange their predicted trajectories and use them as estimations of neighbors' trajectories. This appears to be a reasonable choice, but it requires higher communication bandwidth than the scheme that uses only current neighbors' states and extrapolates them for prediction. Moreover, few research is done about the cases that these predictions are not accurate due to external disturbances and/or model uncertainties.

In our research, the transferred spatial positions of neighboring agents are extrapolated over the finite horizon. The predicted positions of neighboring vehicles are represented by

$$\mathbf{x}_{-i}^S(\tau + t) = \mathbf{x}_{-i}^S(\tau) + \dot{\mathbf{x}}_{-i}^S(\tau)t \quad \text{for } 0 \leq t \leq L. \quad (42)$$

As shown in the following section, acceptable performance can be achieved by this scheme. However, since the prediction error increases as the prediction horizon extends, the extension of the length of prediction horizon does not mean the enlargement of domain of attraction. In addition, the velocity information transferred to neighbors should be properly filtered so that the effects of noisy measurements can be minimized.

IV. Simulations

The MPC algorithm for autonomous helicopter formations as formulated and described above was implemented in Matlab/Simulink environment. The core of the implementation involves a solution to the FHOC problem, and the following section is devoted to a discussion of numerical algorithms for FHOC problems.

A. Numerical Solver for Finite-Horizon Optimal Control Problem

In general, the nonlinear FHOC problem can be numerically solved in two ways: indirect and direct approaches. Indirect approaches utilize the necessary conditions given by the Pontryagin Minimum Principle. Then, it can be viewed as a multi-point boundary value problem, and an optimal solution is obtained by boundary value problem solvers like shooting methods and finite-element methods. However, these indirect methods are known to be very sensitive to their initial conditions, and as a result, lack robustness. See Ref. 28 and references therein. Therefore, it is not practical to use indirect methods in solving online optimization problems.

Direct methods, on the other hand, discretize continuous dynamics and cost functions using a high-order Runge-Kutta method^{28,29} or direct collocation,³⁰ convert it into a finite-dimensional nonlinear optimization problem, and obtain an optimal solution through nonlinear programming techniques. These provide approximate solutions, but they are robust against arbitrary initial conditions, and optimal solutions with reasonable accuracy can be achieved using less intensive numerical procedures than indirect methods.

In the following simulations, we use the DynOpt package.²⁹ This package uses the 4th order Runge-Kutta method for discretization of the continuous-time dynamics and cost functions, and achieves an optimal solution using the sequential unconstrained minimization technique (SUMT). Since the package contains the SUMT algorithm and it is tightly integrated with discretization procedures, DynOpt allows for compact and versatile implementation. Although other solvers using direct method like NTG³¹ and DIRCOL³⁰ require an external commercial nonlinear programming solver, it is worthwhile to use them in that they provide more user-friendly options and detailed error messages for debugging. In the case of NTG, it was reported that the package is used as an MPC engine in a hard real-time application.¹⁰

B. Simulation Setup

Figure 4 shows the configuration of the right echelon formation¹ used in simulations. We chose this configuration because we want to investigate the propagation of disturbances through connected vehicles.

In the case of the constant gap strategy (vehicles in edges, Vehicle 0 and 7 in Figure 4), gap vectors are defined as

$$l_{i,i-1}^r = \begin{bmatrix} 30 \\ -30 \\ 30 \end{bmatrix} \quad l_{i,i+1}^r = \begin{bmatrix} -30 \\ 30 \\ -30 \end{bmatrix}, \quad (43)$$

and all units are in ft.

When we defined the FHOc problem (Eq.(23)), we assumed that there are constraints on states such that $\mathbf{x}_i \in \mathcal{X}_i$. In recent publications,^{21,32} it is reported that the state-constrained MPC scheme is possibly not robust due to the discontinuity in the value function induced by state constraints. In accordance with this observation, we use only input constraints in our application, even though our FHOc problem solver allows state constraints in the formulation. We use the following admissible input set for all vehicles in our simulations

$$\mathcal{U}_i = \{\mathbf{u} \in \mathbb{R}^{n_i^u} \mid -1 \leq u^j \leq 1, 1 \leq j \leq n_i^u\}, \quad (44)$$

where u^j denotes the j -th element of the vector \mathbf{u} .

A final consideration here is the weighting on the attitude states, ϕ and θ (roll and pitch angles). It is well known in the field of aircraft control that the stabilization of attitude dynamics is a key to good controller design. This is due to the coupling between the translational dynamics and the attitude dynamics. In order to keep the attitude variation at a minimum, the terms related with ϕ and θ in Eq.(32) should be more heavily penalized than other terms.

C. Performance with a Homogeneous Formation

In order to investigate disturbance attenuation performances of the proposed controller, we exert negative longitudinal wind gust on the leading vehicle. The longitudinal acceleration induced by the wind gust is shown in Figure 5.

As shown in Figure 6, the MPC algorithm with the varying gap strategy successfully damps out the relative gap errors caused by the disturbance as they propagate into the homogeneous formation, i.e., the maximum gap errors caused by the disturbance between vehicles become smaller and smaller.

D. Performance with a Heterogeneous Formation

In order to test our algorithm in a heterogeneous setup, we put virtual models from Section II in locations of 1, 2, 4, 5, and 6 in the right echelon formation (Figure 4). For Vehicles 0, 3, and 7, the original R-50 cruise model is used.

As shown in Figure 7, the proposed algorithm successfully damps out external disturbance as it propagates through this heterogeneous formation.

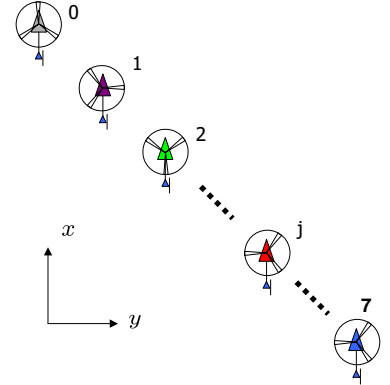


Figure 4. Right echelon formation with eight helicopters used in simulations

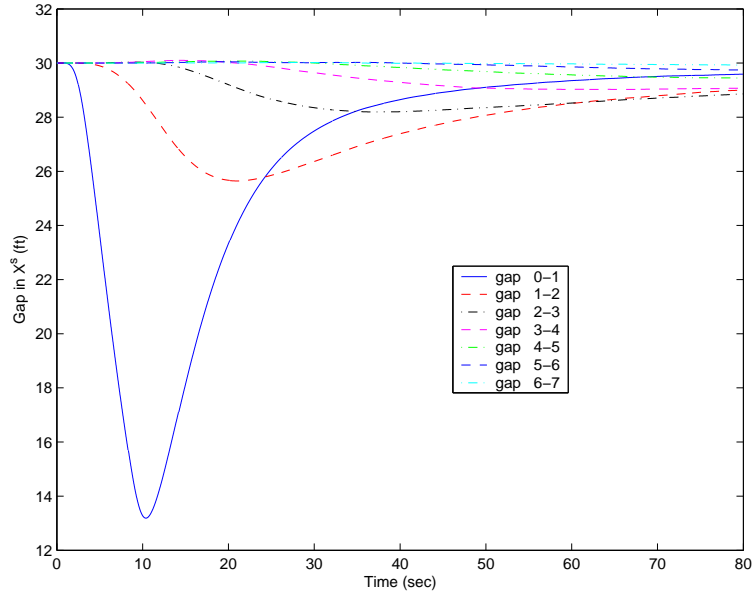


Figure 6. Gaps in x-direction, homogeneous formation under negative gust in longitudinal direction

In the case of mesh stability algorithm,³³ the gap error induced by the leader motion was amplified between a normal vehicle and a more agile vehicle. Since the algorithm uses position information of the leader as well as neighboring vehicles, and an agile vehicle tends to maintain the relative position from the leader even when the relative gap errors between neighbors become large, there exists a ‘jump’ in gap error propagation. However, in our MPC-based formulation, since vehicles share only reference velocities and heading, the dilemma of the global connection to the leader does not appear, and the space between vehicles can be safely maintained.

Figure 8 shows comparisons of gaps between homogeneous and heterogeneous formations. Note the figure related with Vehicle 0 and 3. Although there exists slight performance degradation comparing with the homogeneous case, the proposed algorithm shows comparable disturbance attenuation capability in heterogeneous formation.

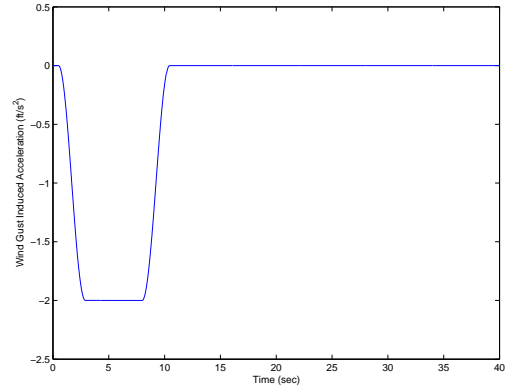


Figure 5. Disturbance induced by wind gust, negative x direction

V. Conclusions and Future Work

In this paper, the problem of autonomous helicopter formations is considered. A stable MPC-based controller for a single helicopter was implemented first, and then carefully designed inter-vehicle coupling terms were added in order to maintain safe space between helicopters. In Section III, we showed the proposed algorithm successfully damps out exogenous disturbances via a series of simulations. The algorithm was also applied to a heterogeneous formation, and it showed a good attenuation property.

The current issue of the proposed scheme is that the FHOC problem solver, DynOpt, is too slow for real-time applications. It takes about an hour to perform an 8-vehicle formation simulation for 80 seconds. Our algorithm will be tested with different solvers such as the gradient descent method.³⁴ We are optimistic because we already have performed successful MPC experiments using the gradient decent method.⁶ The enhancement of the performance of our existing solver is also now being pursued. As soon as the development of the enhanced FHOC solver is completed, the proposed algorithm will be implemented on Berkeley AeRobot (BEAR) testbed,^{3,18} and a series of experiments will be performed.

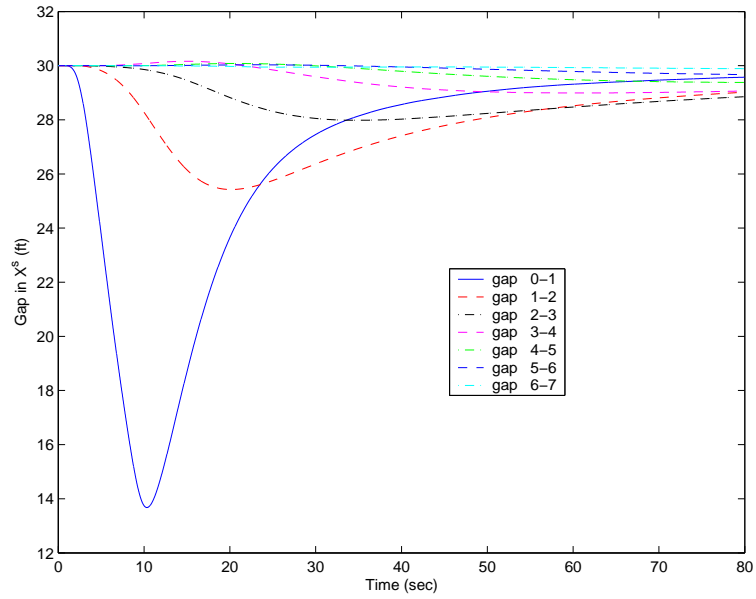


Figure 7. Gaps in x-direction, heterogeneous formation under negative gust in longitudinal direction

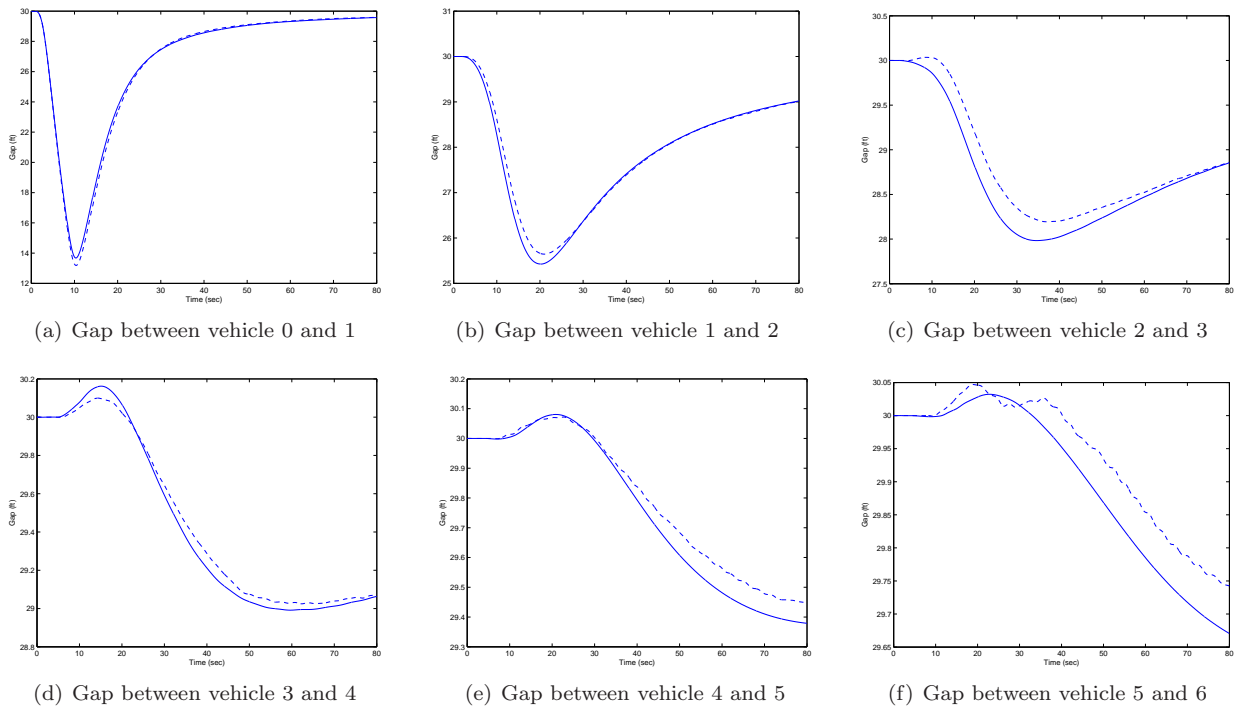


Figure 8. Comparison of gaps in x-direction, homogeneous formation: dashed lines, heterogeneous formation: solid lines

In this work, collision avoidance between neighboring vehicles has not been addressed explicitly. The MPC-based formation controller copes with it only in an implicit manner, in terms of disturbance attenuation. In addition to collision avoidance, individual vehicles in the team may face various emerging situations such as pop-up threats, strong wind gusts, communication failures, and lack of thrust power. Under these unfavorable circumstances, one or more vehicles may separate from the formation until adversities are overcome. Once normal flight conditions are restored, the separated vehicle will rejoin the formation and continue its mission. This means that we need a high-level agent on top of the proposed MPC formation controller that facilitates complex logistics allowing more dynamic and flexible autonomous formation flights. In this context, we introduced the concept of the “formation manager”,¹⁸ and it will be implemented on the BEAR avionics as a part of the autonomous formation flight system.

References

- ¹Headquarters, Department of the army, *Attack helicopter operations*, FM 1-112.
- ²Mettler, B., Kanade, T., and Tischler, M., “System Identification Modeling of a Model-Scale Helicopter,” Tech. rep., Robot Institute at Carnegie Mellon University, December 2000.
- ³Shim, D. H., *Hierarchical Flight Control System Synthesis for Rotorcraft-based Unmanned Aerial Vehicles*, Ph.D. thesis, University of California, Berkeley, 2000.
- ⁴Pant, A. G., *Mesh Stability of Formations of Unmanned Aerial Vehicles*, Ph.D. thesis, University of California, Berkeley, 2002.
- ⁵Seiler, P. J., *Coordinated Control of Unmanned Aerial Vehicles*, Ph.D. thesis, University of California, Berkeley, 2001.
- ⁶Shim, D., Chung, H., Kim, H. J., and Sastry, S., “Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles,” *AIAA Conference on Guidance, Navigation and Control*, 2005.
- ⁷Kothare, M., Banlakrishnan, V., and Morari, M., “Robust Constrained Model Predictive Control using Linear Matrix Inequalities,” *Automatica*, Vol. 32, 1996, pp. 1361–1379.
- ⁸Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., “Survey Paper: Constrained model predictive control: Stability and Optimality,” *Automatica*, Vol. 36, 2000, pp. 789–814.
- ⁹Bemporad, A. and Morari, M., *Robust model predictive control: A survey*, Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 31–45.
- ¹⁰Dunbar, W. B., *Distributed receding horizon control of multiagent systems*, Ph.D. thesis, California Institute of Technology, 2004.
- ¹¹Keviczky, T., Borrelli, F., and Balas, G. J., “Hierarchical design of decentralized receding horizon controllers for decoupled systems,” *IEEE Conference on Decision and Control*, Vol. 2, 2004, pp. 1592–1597.
- ¹²Shim, D. H., Kim, H. J., and Sastry, S., “Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots in Dynamic Environments,” *IEEE Conference on Decision and Control*, Vol. 4, 2003, pp. 3621–3626.
- ¹³Mettler, B., *Identification modeling and characteristics of miniature rotorcraft*, Kluwer Academic Publishers, 2003.
- ¹⁴Craig, J. J., *Introduction to Robotics*, Addison-Wesley, 1989.
- ¹⁵Kim, H., *Multiagent pursuit-evasion games: algorithms and experiments*, Ph.D. thesis, University of California, Berkeley, 2001.
- ¹⁶Dunbar, W. D. and Murray, W., “Distributed receding horizon control with application to multi-vehicle formation stabilization,” *Automatica*, (to appear).
- ¹⁷<http://www.swarms.org/>.
- ¹⁸Chung, H., *Autonomous Formation Flight of Helicopters: Model Predictive Control Approach*, Ph.D. thesis, University of California, Berkeley, 2006.
- ¹⁹Chen, H. and Allgöwer, F., “A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability,” *Automatica*, Vol. 34, No. 10, 1998, pp. 1205–1217.
- ²⁰Jadbabaie, A., *Receding Horizon Control of Nonlinear Systems: A Control Lyapunov Function Approach*, Ph.D. thesis, California Institute of Technology, 2000.
- ²¹Grimm, G., Messina, M. J., Tuna, S. E., and Teel, A., “Examples when nonlinear model predictive control is nonrobust,” *Automatica*, Vol. 40, 2004, pp. 1729–1738.
- ²²Jadbabaie, A., Yu, J., and Hauser, J., “Unconstrained receding horizon control of nonlinear systems,” *IEEE Transactions on Automatic Control*, Vol. 37, 2001, pp. 1971–1978.
- ²³Gahinet, P., Nemirovski, A., Laub, A. J., and Chilali, M., *LMI Control Toolbox*, Mathworks, Inc., 1995, Now merged to Robust Control Toolbox.
- ²⁴Ghaoui, L. E., Nikoukchah, R., and Delebecque, F., “LMITool: a Package for LMI Optimization,” *IEEE Conference on Decision and Control*, 1995.
- ²⁵Sturm, J. F., “Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones,” *Optimization Methods and Software*, Vol. 11-12, 1998, pp. 625–653.
- ²⁶Raffard, R. L., Tomlin, C. J., and Boyd, S. P., “Distributed Optimization for Cooperative Agents: Application to Formation Flight,” *IEEE Conference on Decision and Control*, December 2004, pp. 2453–2459.
- ²⁷Camponogara, E., Jia, D., Krogh, B. H., and Talukdar, S., “Distributed Predictive Control,” *IEEE Control Systems Magazine*, February 2002, pp. 44–52.
- ²⁸Schwartz, A. L., *Theory and Implementation of Numerical Methods Based on Runge-Kutta Integration for Solving Optimal Control Problems*, Ph.D. thesis, University of California, Berkeley, 1996.

²⁹Fabien, B. C., “Some Tools for the Direct Solution of Optimal Control Problems,” *Advances in Engineering Software*, Vol. 29, 1998, pp. 45–61.

³⁰von Stryk, O., “Numerical solution of optimal control problems by direct collocation,” *Optimal Control - Calculus of Variations, Optimal Control Theory and Numerical Methods*, edited by J. S. R. Bulirsch, A. Miele and K.-H. Well, International Series of Numerical Mathematics 111, Birkhuser, 1993, pp. 129–143.

³¹Milam, M. B., Mushambi, K., and Murray, R. M., “A Computational Approach to Real-Time Trajectory Generation for Constrained Mechanical Systems,” *IEEE Conference on Decision and Control*, Vol. 1, 2000, pp. 845 – 851.

³²Grimm, G., Messina, M. J., Tuna, S. E., and Teel, A., “Model predictive control: for want of a local control lyapunov function, all is not lost,” *IEEE Transactions on Automatic Control*, Vol. 50, 2005, pp. 546–558.

³³Shaw, E., Chung, H., Hedrick, J., and Sastry, S., “Unmanned Helicopter Formation Flight Experiment for the Study of Mesh Stability,” *Advances in Cooperative Control and Optimization*, edited by D. Grundel, R. Murphey, P. Pardalos, and O. Prokopyev, Kluwer Academic, 2005.

³⁴Sutton, G. J. and Bitmead, R. R., “Performance and Computational Implementation of Nonlinear Model Predictive Control on a Submarine,” *Nonlinear Model Predictive Control*, edited by F. Allgöwer and A. Zheng, Vol. 26 of *Progress in Systems and Control Theory*, Birkhäuser, 2000, pp. 461–471.