Title

On the Perpetual Collision-Free RHC of Fleets of Vehicles

Journal

Journal of Optimization Theory and Applications, 145(1)

ISSN

1573-2878

Authors

Gonzalez, H.
Polak, E.

Publication Date

2010-04-01

DOI

10.1007/s10957-009-9633-1

Peer reviewed

J Optim Theory Appl (2010) 145: 76–92
DOI 10.1007/s10957-009-9633-1

On the Perpetual Collision-Free RHC of Fleets of Vehicles

H. Gonzalez · E. Polak

Published online: 30 October 2009

Abstract Receding horizon control is emerging as a very promising technique for the centralized control of fleets of vehicles on land, sea, and in the air. We present a sufficient condition for collision avoidance within a fleet under receding horizon control, over an indefinite period of operation.

Keywords Collision avoidance · Multi-agent control · Min-max-min algorithm · Receding horizon control

1 Introduction

Airplanes in holding patterns over airports or taxing on runways, ships entering or leaving harbors, drones circling overhead waiting to be directed to a mission, automated container terminals in wharves where autonomous robotic vehicles load and unload ships are all under centralized control, usually under human control, but progressively more often under computer assisted control. The human control is provided by air traffic controllers, harbor masters, drone dispatchers, etc.

Receding horizon control (RHC) is emerging as a very promising technique for more automated and safer centralized control of fleets of vehicles on land, sea, and

Communicated by David Q. Mayne.

H. Gonzalez (✉) · E. Polak
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770, USA
e-mail: hgonzale@eecs.berkeley.edu

E. Polak
e-mail: polak@eecs.berkeley.edu

in the air. A common feature of the above cited examples is that, in all of these situations, collision avoidance is a major issue.[1, 2]

Receding horizon control is an advanced form of sample-data control. But, unlike in classical sample-data control, the control is not constant over the sample intervals and is not determined by a linear compensator and sample-and-hold circuit. In an idealized version which ignores computing time, at time $k\Delta$, where $\Delta > 0$ is the *sampling time* and $k \in \mathbb{N}$, the state of the dynamical system is measured and then the RHC digital computer solves an optimal control problem (using as initial state the measured state) whose cost function expresses the desired goals to be achieved and whose constraints are defined by the dynamical system limitations as well as by external considerations. The time horizon T for the optimal control problem is usually much larger than the sampling time Δ, and in some cases may be free, i.e., it may be a decision variable. The optimal control computed at time $k\Delta$ is applied only for Δ time units and is then recomputed at time $(k+1)\Delta$, using as the initial state the state measured at time $(k+1)\Delta$. The reinitialization of the optimal control problem serves as a feedback mechanism [1].

Typically, the cost function is a weighted sum of desired goals, such as an input energy term, distance to destination term, and penalty terms for the violation of soft constraints. The hard constraints, expressed as inequalities, include input constraints (such as amplitude constraints) and, in the above cited examples, collision avoidance constraints, which are state-space constraints.

For the RHC scheme to be successful, the optimal control problem that determines the control must have feasible solutions at each sampling time. In this context, the control constraints are never an issue, the real issue are the state-space constraints, which are intended to ensure that the minimum distance between various vehicles and between vehicles and obstacles remains larger than some safe tolerance.

Now, for example, consider the situation of a car that must be kept at constant speed within a winding lane. Clearly, under realistic assumptions on the vehicle dynamics and steering angles, if at time $t = k\Delta$ the vehicle is on the boundary of the lane and faces outward, the "remain in-lane" constraint will be violated in the time interval $t \in [k\Delta, (k+1)\Delta]$. Thus, it is clear that constraint violation will occur unless some condition on the state at the times $t = k\Delta$, $k = 0, 1, 2, \ldots$, can be imposed and maintained.

Many authors, see e.g. [2] and [3], deal with collision avoidance in RHC by means of barrier functions that are added to the cost function and which grow to infinity as the distance between vehicles shrinks to zero. One may be lead to the conclusion that this approach provides a sufficient condition for collision avoidance, since the

[1]Quoting from the New York Times, April 26, 2008 *"Where we are most vulnerable at this moment is on the ground. To me, this is the most dangerous aspect of flying."—Mark V. Rosenker, chairman of the National Transportation Safety Board*. For the six-month period that ended March 30, there were 15 serious runway incursions, compared with 8 in the period a year earlier. Another occurred at Dallas-Fort Worth International Airport on April 6 when a tug operator pulling a Boeing 777 along a taxiway failed to stop at a runway as another plane was landing, missing the tug by about 25 feet.

[2]On November 7, 2007, the cargo ship Cosco Busan, sailing in heavy fog from the port of Oakland to the open sea under harbor pilot control, struck the Bay Bridge due to pilot error, resulting in a major oil spill. That would not have happened under computer control.

augmented cost function is minimized. However, this is a rather optimistic conclusion, since it does not take into account the effect of control constraints, which are always present and which can prevent collision avoidance under certain state configurations of the vehicles. So far, it seems, it has not been established that, in a multi-agent moving obstacle environment, such unfavorable state configurations will never arise.

In this paper, we present a sufficient condition for recursive feasibility, which ensures the perpetual collision avoidance for a fleet of vehicles with time-invariant dynamics, operated under centralized RHC and confined to operate within a bounded region of physical space.

The motivation behind our sufficient conditions is that, if the vehicles have a sufficiently large initial separation, then they will maintain a minimum separation for a given time horizon. Hence, our sufficient condition is in the form of two inequalities that must be included in the RHC optimal control problem. The first inequality requires that the minimum separation between individual vehicles is at least $\rho_{\min} > 0$ over the entire *interval* $[0, \Delta]$. The second inequality requires that the minimum spatial separation between individual vehicles at the sampling time $t = \Delta$ is at least $r > \rho_{\min}$. Reasoning by induction, it is easy to conclude that, provided there exists a feasible solution to the resulting augmented optimal control problem for any set of initial states, in a region of interest, that are pairwise at least r apart, the fleet can be kept collision-free perpetually.

An enhancement of our sufficient condition, allowing one to take into account state measurement errors and disturbances can be obtained in a reasonably straightforward manner by making use of the results in [4]. This enhancement leads to a requirement of larger separation.

Since the sufficient condition deals only with feasibility, it should be clear that the cost function, *but not the constraints*, of the centralized RHC optimal control problem can be changed as time goes on without affecting the recursive feasibility of the RHC problem.

Also, on some thought, it becomes clear that if $r > 0$ is a satisfactory separation distance at each sampling time for a fleet of N_v vehicles, it is also a satisfactory separation distance for any subfleet of less than N_v vehicles. This observation leads to the conclusion that vehicles can be assembled into a fleet as time goes on. Thus, we see that our sufficient condition is compatible with sophisticated control schemes.

Finally, to be realistic, for use in an environment where totally unforeseen circumstances can occur, RHC must be used only as a powerful aid for a human operator, with additional features added that permit the human operator to intervene when an emergency condition arises.

In Sect. 2, we present the optimal control problem associated with the RHC scheme. In Sect. 3, we present our main theoretical result, the sufficient condition for perpetual collision avoidance. In Sect. 4, we discuss the numerical evaluation of the function $\psi(r)$, which accounts for the worst-case minimum distance between vehicles given the initial separation was r. In Sect. 5, we present a numerical evaluation of the function $\psi(r)$ for the case of four drones flying at constant speed in a circle. In the concluding Sect. 6, we summarize our findings.

2 Receding Horizon Control Law Formulation

Any RHC law is based on the recursive solution of an associated optimal control problem. We begin by describing a generic optimal control problem for the centralized RHC of a fleet of vehicles.

Suppose that we have N_v vehicles. We assume that, for each $i = 1, \dots, N_v$, the ith vehicle dynamics are given by a time-invariant differential equation of the form

$$\dot{x_i}(t) = h_i(x_i(t), u_i(t)), \quad t \geq 0, \qquad x_i(0) = \zeta_i, \tag{1}$$

where $x_i(t) \in \mathbb{R}^n$ is the state of the vehicle at time t, ζ_i is the initial state, $u_i(t) \in \mathbb{R}^m$ is the input at time t and $h_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.[3] We will denote the solutions of (1) by $x_i^{(\zeta_i,u_i)}(t)$ and we will assume that a part of the state denoted by $x_{iP}^{(\zeta_i,u_i)}(t) \in \mathbb{R}^d$ represents the Cartesian coordinates of the vehicle with respect to a fixed arbitrary origin, where $d \in \{2, 3\}$ will take a value depending on the particular model used. Note that, without loss of generality, we can consider the initial time to be zero since the model is time invariant.

We assume that the inputs are elements of the space

$$\mathcal{U} = \left\{ u(\cdot) \in \mathcal{L}^m_{\infty,2}[0, T] \mid \|u(t)\|_\infty \leq \alpha \right\}, \tag{2}$$

where $\mathcal{L}^m_{\infty,2}[0, T]$ is a pre-Hilbert space with the same elements as $\mathcal{L}^m_\infty[0, T]$, but endowed with the inner product and norm of $\mathcal{L}^m_2[0, T]$, with $\alpha < \infty$ and $T < \infty$ a fixed horizon. We use the space $\mathcal{L}^m_{\infty,2}[0, T]$, because it makes it possible to establish the continuity and differentiability of the solutions of (1) with respect to the controls as well as to relate the optimality conditions for discretizations of the continuous optimal control problem to those of the original optimal control problem; see Sect. 5.4 in [5].

Since most numerical optimization methods require differentiability of the discretized differential equation with respect to the controls, we introduce the following assumption.

Assumption 2.1 (Lipschitz Continuity) Consider the system defined by (1). We assume that there exists $L \in (0, \infty)$ such that, for all $x_1, x_2 \in \mathcal{B}$, a sufficiently large ball in $\mathbb{R}^n$, for all $u_1, u_2 \in \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq \alpha\}$ and, for all $i \in \{1, \dots, N_v\}$,

$$\|h_i(x_1, u_1) - h_i(x_2, u_2)\| \leq L\left(\|x_1 - x_2\| + \|u_1 - u_2\|\right), \tag{3}$$

$$\left\| \frac{\partial h_i}{\partial x}(x_1, u_1) - \frac{\partial h_i}{\partial x}(x_2, u_2) \right\| \leq L\left(\|x_1 - x_2\| + \|u_1 - u_2\|\right), \tag{4}$$

$$\left\| \frac{\partial h_i}{\partial u}(x_1, u_1) - \frac{\partial h_i}{\partial u}(x_2, u_2) \right\| \leq L\left(\|x_1 - x_2\| + \|u_1 - u_2\|\right). \tag{5}$$

[3] To simplify exposition, we assume that the states and inputs of all vehicle model are of the same dimension.

To make our notation more compact, we define

$$\mathcal{N}_v \triangleq \{1, 2, \dots, N_v\}, \tag{6}$$

$$U \triangleq \mathcal{U} \times \mathcal{U} \times \cdots \times \mathcal{U} \quad (N_v \text{ times}), \tag{7}$$

$$R \triangleq \mathbb{R}^n \times \mathbb{R}^n \times \cdots \times \mathbb{R}^n \quad (N_v \text{ times}), \tag{8}$$

$$u \triangleq (u_1, u_2, \dots, u_{N_v}) \in U, \tag{9}$$

$$\zeta \triangleq (\zeta_1, \zeta_2, \dots, \zeta_{N_v}) \in R. \tag{10}$$

We assume that we are given the *initial states* $\zeta = (\zeta_1, \zeta_2, \dots, \zeta_{N_v})$, for the vehicle dynamics in (1), a *time horizon* $T \in \mathbb{R}_+$, a *sample time* $\Delta \in \mathbb{R}_+$ with $\Delta \le T$, a problem specific differentiable cost function $f^0 : R \times U \to \mathbb{R}$ and a set of q problem specific differentiable constraint functions $f^j : R \times U \to \mathbb{R}$, $j = 1, \dots, q$.[4]

Since in this paper we are only interested in collision avoidance, the specific form of the functions $f^j(\cdot)$, $j = 1, \dots, q$, is not relevant to our discussion.

In addition, we include a set of collision avoidance constraints of the form:

$$\left\| x_{iP}^{(\zeta_i,u_i)}(t) - x_{jP}^{(\zeta_j,u_j)}(t) \right\|^2 \ge \rho_{\min}^2, \quad \forall i, j \in \mathcal{N}_v,\ i \ne j,\ \forall t \in [0, \Delta], \tag{11}$$

$$\left\| x_{iP}^{(\zeta_i,u_i)}(\Delta) - x_{jP}^{(\zeta_j,u_j)}(\Delta) \right\|^2 \ge r^2, \quad \forall i, j \in \mathcal{N}_v,\ i \ne j, \tag{12}$$

$$x_i^{(\zeta_i,u_i)}(\Delta) \in \mathcal{S}, \quad \forall i \in \mathcal{N}_v, \tag{13}$$

where $\rho_{\min} \in \mathbb{R}_+$ is the minimum safety distance between the vehicles, $r \ge \rho_{\min}$, and $\mathcal{S} \subset \mathbb{R}^n$ is a compact set with interior.

Combining these elements, we obtain the following optimal control problem.

Optimal Control Problem 2.1 Given the initial states $\zeta = (\zeta_1, \dots, \zeta_{N_v}) \in R$, of the vehicles at time $t = 0$, compute the set of inputs $\hat{u} = (\hat{u}_1, \dots, \hat{u}_{N_v}) \in U$ as a solution of the problem:

$$\text{(OCP)} \quad \min_{u = \{u_1, \dots, u_{N_v}\} \in U} f^0(\zeta, u), \tag{14}$$

$$\text{s.t.} \quad f^j(\zeta, u) \le 0, \quad j = 1, \dots, q, \tag{15}$$

$$\left\| x_{iP}^{(\zeta_i,u_i)}(t) - x_{jP}^{(\zeta_j,u_j)}(t) \right\|^2 \ge \rho_{min}^2, \quad \forall i, j \in \mathcal{N}_v,\ i \ne j,\ \forall t \in [0, \Delta], \tag{16}$$

$$\left\| x_{iP}^{(\zeta_i,u_i)}(\Delta) - x_{jP}^{(\zeta_j,u_j)}(\Delta) \right\|^2 \ge r^2, \quad \forall i, j \in \mathcal{N}_v,\ i \ne j, \tag{17}$$

$$x_i^{(\zeta_i,u_i)}(\Delta) \in \mathcal{S}, \quad \forall i \in \mathcal{N}_v. \tag{18}$$

[4]Cost functions with an integral term, free time optimal control problems and constraints on the derivatives of the input functions can be transformed into this form. See Sect. 4.1.2 in [5].

Assuming that the computing time required to solve Optimal Control Problem 2.1 is negligible with respect to Δ, the RHC law is defined as follows.

Algorithm 2.1 (Receding Horizon Control Law) Given the states ζ_i, $i = 1, \ldots, N_v$, of the vehicles at time $t = 0$.

Step 1. Set $k = 0$.
Step 2. Set $t_k = k\Delta$.
Step 3. Measure the vehicle states $x_i(t_k)$, $i = 1, \ldots, N_v$, and set $\zeta_i = x_i(t_k)$.
Step 4. Solve Optimal Control Problem 2.1 for the optimal controls $\hat{u}_i(t)$, $i = 1, \ldots, N_v$, with $t \in [0, T]$, and define the controls $u_i(t)$, $i = 1, \ldots, N_v$, with $t \in [t_k, t_k + T]$, by

$$u_i(t) = \hat{u}_i(t - t_k), \quad t \in [t_k, t_k + T], \quad i = 1, \ldots, N_v. \tag{19}$$

Step 5. Apply the control u_i during the time interval $[t_k, t_k + \Delta)$ to the ith vehicle, with $i = 1, \ldots, N_v$.
Step 6. Increase k by one and go back to Step 2.

This is a feedback law because the optimal control problem is resolved every Δ seconds, using the observed states of the vehicles as new initial states ζ_i. In a more sophisticated scheme, the time needed to solve the optimal control problem and disturbances can be accounted for; see [4].

Note that there is nothing in this formulation that ensures that a feasible solution of Optimal Control Problem 2.1 exists and, in particular, that no collisions will occur over the time of operation.

The selection of the cost function $f^0(\cdot, \cdot)$ depends on the particular physical problem. Consider the following examples:

Example 2.1 Control of a fleet of commercial airplanes, each flying to a possibly different destination. In this case, one might wish to minimize a weighted sum of two terms, the first being the fuel consumption and the second being the deviation from the scheduled arrival time. The time horizon T becomes a variable in this application, since as the airplanes get closer to their destinations the time-to-go becomes shorter and shorter. Thus, the function $f^0(\cdot, \cdot)$ might have the following form:

$$f^0(\zeta, u, T) = \sum_{i=1}^{N_v} \int_0^T \|u_i(t)\|^2 dt + w(T - T^*)^2, \tag{20}$$

where T^* is the desired arrival time and $w > 0$ is a weighting factor.

Example 2.2 A set of airplanes confined to fly in a bounded space for a long time (as in a holding pattern over an airport). In this case, minimizing fuel consumption might be the most sensible thing to do, and the cost function $f^0(\cdot, \cdot)$ might have the following form:

$$f^0(\zeta, u) = \sum_{i=1}^{N_v} \int_0^T \|u_i(t)\|^2 dt. \tag{21}$$

In addition, a set of inequalities for ensuring that the airplanes remain in the holding pattern needs to be added:

$$f^j(\zeta,u) = \max_{t\in[0,T]} \left\| x_{jP}^{(\zeta_j,u_j)}(t) - x_c \right\|^2 - \rho^2, \quad j = 1,\dots,N_v, \tag{22}$$

where $x_c \in \mathbb{R}^d$ is the center of the circular holding pattern and $\rho > 0$ is its radius.

Example 2.3 Vehicles joining a holding pattern from arbitrary initial positions. In this case, one would not impose inequalities to keep the vehicles in the holding pattern, but instead one would use a cost function that imposes a penalty for straying outside the holding pattern, e.g.,

$$f^0(\zeta,u) = \sum_{i=1}^{N_v} \int_0^T \|u_i(t)\|^2 dt + w \sum_{i=1}^{N_v} \int_0^T \left(\left\| x_i^{(\zeta_i,u_i)}(t) - x_c \right\|^2 - \rho^2 \right)^2 dt. \tag{23}$$

3 Problem of Perpetual Collision Avoidance

We begin with a definition. Let

$$S \triangleq \mathcal{S} \times \cdots \times \mathcal{S} \quad (N_v \text{ times}), \tag{24}$$

where $\mathcal{S} \subset \mathbb{R}^n$ has a nonempty interior. For any $r > 0$, we define the set of *admissible* initial states, parameterized by r, by

$$\mathcal{I}(r) \triangleq \left\{ \zeta = (\zeta_i)_{i=1}^{N_v} \in S \;\middle|\; \|\zeta_{iP} - \zeta_{jP}\| \ge r \;\forall i,j \in \mathcal{N}_v,\ i \ne j \right\}. \tag{25}$$

The easiest way to visualize the set $\mathcal{I}(r)$ is as a set of elements in $\mathbb{R}^{N_v\times n}$. Thus, suppose that $N_v = 2$ and $n = 1$. In this case, we see from Fig. 1 that $\mathcal{I}(r)$ is $\mathbb{R}^2$, with a diagonal strip cut out (i.e., it consists of two half planes). In general, it is $\mathbb{R}^{N_v\times n}$ with a diagonal cylinder cut out.

Proposition 3.1 *Suppose that there exists an $r > \rho_{\min} > 0$ such that, for every $\zeta = (\zeta_i,\dots,\zeta_{N_v}) \in \mathcal{I}(r)$, there exists a set of* feasible *controls $u = (u_1^f,\dots,u_{N_v}^f) \in U$ such that the states $x_i^{(\zeta_i,u_i^f)}(t)$, $t \in [0,T]$, $i \in \mathcal{N}_v$, satisfy the constraints* (15)–(18).

If the set of the initial states $\zeta = (x_1(0),\dots,x_{N_v}(0))$ is in $\mathcal{I}(r)$, then for all $k = 0,1,2,\dots$. Algorithm 2.1 will construct a set of optimal controls $u_i(t)$, $t \in [k\Delta,(k+1)\Delta]$, that satisfy the constraints (15)–(18).

Proof Since by assumption the set of initial states $\zeta = (x_1(0),\dots,x_{N_v}(0)) \in \mathcal{I}(r)$, and since we have assumed that, for every $\zeta \in \mathcal{I}(r)$, there exists a *feasible* control $u_i^f \in \mathcal{U}$ such that the states $x_i^{(\zeta_i,u_i^f)}$, $i \in \mathcal{N}_v$, satisfy the constraints (15)–(18), Algorithm 2.1 will construct a set of *optimal* controls $u_i(t)$, $i \in \mathcal{N}_v$, $t \in [0,T]$, which satisfy the constraints (15)–(18).

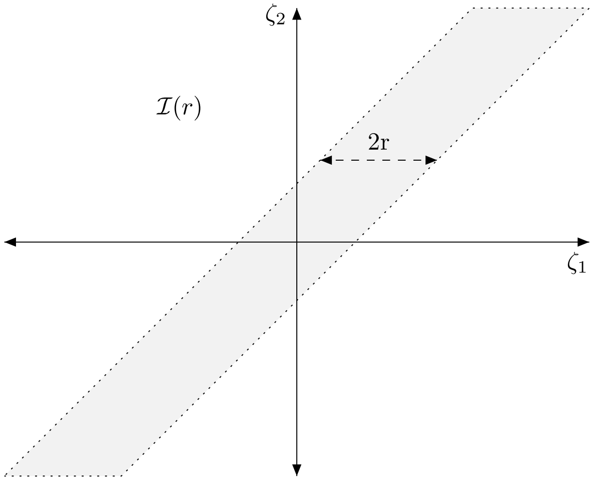


Fig. 1 Set of initial states $\mathcal{I}(r)$ for $N_v = 2$ and $n = 1$

Since by (17) and (18) the final states $(x_1(\Delta), \ldots, x_{N_v}(\Delta)) \in \mathcal{I}(r)$, it follows by induction, that, for all $k = 1, 2, 3, \ldots$, the optimal control problem faced by Algorithm 2.1 will have feasible solutions; hence, its optimal solutions will be feasible.[5] $\square$

Corollary 3.1 (Perpetual Collision Avoidance) *Suppose that the assumptions of Proposition 3.1 are satisfied and that the fleet of N_v vehicles is controlled by the centralized RHC Algorithm 2.1. Then*:

(a) *The vehicles will never collide.*
(b) *The state of the vehicles will remain bounded.*

The proof is straightforward by induction since both properties are satisfied at each interval $[k\Delta, (k+1)\Delta]$, with $k = 0, 1, 2, \ldots$.

We now turn to the task of developing a function $\psi : \mathbb{R} \to \mathbb{R}$ that can be used for testing whether a particular value $r \geq \rho_{\min}$, of the minimum initial physical separation between vehicles is large enough to ensure perpetual collision avoidance.

For any $i, j \in \mathcal{N}_v$, $i \neq j$, we define the functions $d_{ij}^2 : S \times U \times [0, \Delta] \to \mathbb{R}$ by

$$d_{ij}^2(\zeta, u, t) \triangleq \left\| x_{iP}^{(\zeta_i, u_i)}(t) - x_{jP}^{(\zeta_j, u_j)}(t) \right\|^2. \tag{26}$$

The value $d_{ij}^2(\zeta, u, t)$ is the squared distance between the vehicles i and j at time t for the given sets of initial states ζ and controls u.

Next, we define the function $\phi_1 : S \times U \times [0, \Delta] \to \mathbb{R}$ by

$$\phi_1(\zeta, u, t) \triangleq \min_{i, j \in \mathcal{N}_v, i \neq j} d_{ij}^2(\zeta, u, t). \tag{27}$$

The value $\phi_1(\zeta, u, t)$ is the shortest distance squared between any two vehicles at time $t \in [0, \Delta]$.

[5] This argument is known in the literature as *recursive feasibility* (see [6]) or *robust feasibility* (see [7]).

Next, we define the function $\phi_2 : S \times U \to \mathbb{R}$ by

$$\phi_2(\zeta, u) \triangleq \min_{t\in[0,\Delta]} \phi_1(\zeta, u, t). \tag{28}$$

The value $\phi_2(\zeta, u)$ is the shortest distance squared between any two vehicles during the time interval $[0, \Delta]$.

Next, we define the function $\phi_3 : \mathbb{R}_+ \times S \to \mathbb{R}_+$ by

$$\phi_3(r, \zeta) \triangleq \max_{u\in U} \Big\{ \phi_2(\zeta, u) \,\big|\, f^j(\zeta, u) \le 0,\ j = 1, \dots, q, \\ \phi_1(\zeta, u, \Delta) \ge r^2;\ x_i^{(\zeta_i, u_i)}(\Delta) \in \mathcal{S},\ \forall i \in \mathcal{N}_v \Big\}, \tag{29}$$

if there exists a u that satisfies the constraints in (29), and we define $\phi_3(r, \zeta) = 0$ otherwise.

The value $\phi_3(r, \zeta)$ is the largest among the shortest squared distances between any two vehicles during the time interval $[0, \Delta]$ achievable with the available controls, in the presence of the constraints in the optimal control problem OCP.

Finally, we define the function $\psi : \mathbb{R}_+ \to \mathbb{R}_+$ by

$$\psi(r) \triangleq \min_{\zeta\in\mathcal{I}(r)} \phi_3(r, \zeta). \tag{30}$$

The following result should be obvious.

Proposition 3.2 *If there exists an $r \ge \rho_{\min}$ such that $\psi(r) \ge \rho_{\min}^2$ then for every set of initial states $(\zeta_i)_{i=1}^{N_v} \in \mathcal{I}(r)$ there exist a set of inputs $(u_i)_{i=1}^{N_v} \in U$ such that the assumptions of Proposition 3.1 are satisfied.*

The questions that still remain are: (a) whether the function $\psi(r)$ is computable; (b) whether there is an $r \ge \rho_{\min}$ such that $\psi(r) \ge \rho_{\min}^2$. We will address the first of these questions in Sect. 4. In Sect. 5, we will show an example for which such an r exists under a reasonable selection of the parameter Δ.

4 Numerical Evaluation of $\psi(r)$

To evaluate $\psi(r)$, we must discretize the differential equation (1). Although pseudospectral methods, such as those described in [8], are currently favored for solving optimal control problems over classical approaches such as Euler discretization, in the case of multi-agent optimal control problems with large number of collision avoidance constraints fast results can be obtained using forward Euler discretization together with the accelerator technique based on outer approximation presented in [9].

Thus, consider the use of the forward Euler method. Let $N \in \mathbb{N}$ be the number of steps to be used to discretize the interval $[0, T]$ and let $\sigma = \frac{T}{N}$ be the step size. We will assume that the values of Δ and N were chosen so that $N_\Delta = \frac{\Delta}{\sigma}$ is a positive integer.

Next, following the approach outlined in Sect. 5.4 of [5], we introduce the finite-dimensional control subspace of $\mathcal{U}_N \subset \mathcal{U}$ defined by

$$\mathcal{U}_N \triangleq \left\{ u(\cdot) \in \mathcal{U} \,\middle|\, u(t) = \sum_{k=0}^{N-1} \bar{u}(k)\pi_k(t) \right\}, \tag{31}$$

where $\bar{u}(k) \in \mathbb{R}^m$, for each $k = 0, \dots, N-1$, is a discrete-time control and

$$\pi_k(t) = \begin{cases} 1, & \text{if } t \in [k\sigma, (k+1)\sigma), \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

By analogy with (7) and (9), we let

$$\bar{u} \triangleq \big(\bar{u}(0), \dots, \bar{u}(N-1)\big) \in \mathbb{R}^{N\times m}, \tag{33}$$

$$\bar{\mathcal{U}}_N \triangleq \big\{\bar{u} = \big(\bar{u}(0), \dots, \bar{u}(N-1)\big) \,\big|\, \|\bar{u}(k)\|_\infty \le \alpha,\ k = 0, \dots, N-1\big\}, \tag{34}$$

$$\bar{U}_N \triangleq \bar{\mathcal{U}}_N \times \cdots \times \bar{\mathcal{U}}_N \quad (N_v \text{ times}), \tag{35}$$

$$\bar{u}_N \triangleq (\bar{u}_1, \dots, \bar{u}_{N_v}) \in \bar{U}_N. \tag{36}$$

We approximate the differential equation (1) by the difference equation obtained using the Euler forward method,

$$z_i(k+1) = z_i(k) + \sigma h_i\big(z_i(k), \bar{u}_i(k)\big), \quad k \in \mathbb{N}, \qquad z_i(0) = \zeta_i, \tag{37}$$

where ζ_i is the initial state, $z_i(k) \in \mathbb{R}^n$ is the discretized state, and $\bar{u}_i(k) \in \mathbb{R}^m$ is the discretized control, for all $k \in \mathbb{N}$.

We will denote by $(z_i^{(\zeta_i,\bar{u}_i)}(k))_{k=0}^N$ the solution of (37), which approximates the solution of (1), at the times $t = k\sigma$, for $u_i(t) = \sum_{k=0}^{N-1} \bar{u}_i(k)\pi_k(t)$ and the same initial state.

Finally, we need to define appropriate approximations $\bar{f}^j : R \times \bar{U}_N \to \mathbb{R}$ for the functions $f^j(\cdot,\cdot)$, $j = 1, \dots, q$. This needs to be handled on a case by case basis. For example, a suitable approximation of the constraint defined in (22) would be

$$\bar{f}^j(\zeta, u) = \max_{k=0,\dots,N} \left\| z_{jP}^{(\zeta_j,\bar{u}_j)}(k) - x_c \right\|^2 - \rho^2, \quad j = 1, \dots, N_v. \tag{38}$$

We now define approximations for the functions d_{ij}^2, ϕ_l, $l = 1, 2, 3$, and ψ, as follows. For any $i, j \in \mathcal{N}_v$, let $\bar{d}_{ij}^2 : S \times \bar{U}_N \times \{0, 1, \dots, N_\Delta\} \to \mathbb{R}$ be defined by

$$\bar{d}_{ij}^2(\zeta, \bar{u}_N, k) \triangleq \left\| z_{iP}^{(\zeta_i,\bar{u}_i)}(k) - z_{jP}^{(\zeta_j,\bar{u}_j)}(k) \right\|^2. \tag{39}$$

Let $\bar{\phi}_1 : S \times \bar{U}_N \times \{0, 1, \dots, N_\Delta\} \to \mathbb{R}$ be defined by

$$\bar{\phi}_1(\zeta, \bar{u}_N, k) \triangleq \min_{i,j\in\mathcal{N}_v, i\ne j} \bar{d}_{ij}^2(\zeta, \bar{u}_N, k). \tag{40}$$

Let $\bar{\phi}_2 : S \times \bar{U}_N \to \mathbb{R}$ be defined by

$$\bar{\phi}_2(\zeta, \bar{u}_N) \triangleq \min_{k=0,\ldots,N_\Delta} \bar{\phi}_1(\zeta, \bar{u}_N, k), \tag{41}$$

and let $\bar{\phi}_3 : \mathbb{R}_+ \times S \to \mathbb{R}_+$ be defined by

$$\begin{aligned}\bar{\phi}_3(r, \zeta) \triangleq \max_{\bar{u}_N \in \bar{U}_N} \Big\{ \bar{\phi}_2(\zeta, \bar{u}_N) \,\Big|\, \bar{f}^j(\zeta, \bar{u}_N) \le 0,\ j = 1, \ldots, q, \\ \bar{\phi}_1(\zeta, \bar{u}_N, N_\Delta) \ge r^2,\ z_i^{(\zeta_i, \bar{u}_i)}(N_\Delta) \in \mathcal{S},\ \forall i \in \mathcal{N}_v \Big\}\end{aligned} \tag{42}$$

if there exists a $\bar{u}_N$ that satisfies the constraints in (42), and we define $\bar{\phi}_3(r, \zeta) = 0$ otherwise.

Finally, we define $\bar{\psi} : \mathbb{R}_+ \to \mathbb{R}_+$ by

$$\bar{\psi}(r) \triangleq \min_{\zeta \in \mathcal{I}(r)} \bar{\phi}_3(r, \zeta). \tag{43}$$

By definition, the evaluation of $\bar{\phi}_3(r, \zeta)$ involves solving a constrained max-min problem. This can be done either by converting this max-min problem into a constrained nonlinear programming problem, i.e.,

$$\begin{aligned}\bar{\phi}_3(r, \zeta) = -\min_{\bar{u}_N \in \bar{U}_N,\ \sigma \in \mathbb{R}} -\Big\{ \sigma \,\Big|\, \bar{d}_{ij}^2(\zeta, \bar{u}_N, k) \ge \sigma, \\ \forall i, j \in \mathcal{N}_v,\ i \neq j,\ \forall k = 0, \ldots, N_\Delta, \\ \bar{d}_{ij}^2(\zeta, \bar{u}_N, N_\Delta) \ge r^2,\ \forall i, j \in \mathcal{N}_v,\ i \neq j, \\ \bar{f}^j(\zeta, \bar{u}_N) \le 0,\ j = 1, \ldots, q, \\ z_i^{(\zeta_i, \bar{u}_i)}(N_\Delta) \in \mathcal{S},\ \forall i \in \mathcal{N}_v \Big\},\end{aligned} \tag{44}$$

which can be solved by means of any number of nonlinear programming algorithms, or it can be naturally formulated as a constrained min-max problem, i.e.,

$$\begin{aligned}\bar{\phi}_3(r, \zeta) = -\min_{\bar{u}_N \in \bar{U}_N} \Big\{ \max_{\substack{i,j \in \mathcal{N}_v,\, i \neq j, \\ k=0,\ldots,N_\Delta}} -\bar{d}_{ij}^2(\zeta, \bar{u}_N, k) \,\Big|\, \bar{d}_{ij}^2(\zeta, \bar{u}_N, N_\Delta) \ge r^2,\ \forall i, j \in \mathcal{N}_v,\ i \neq j \\ \bar{f}^j(\zeta, \bar{u}_N) \le 0,\ j = 1, \ldots, q \\ z_i^{(\zeta_i, \bar{u}_i)}(N_\Delta) \in \mathcal{S},\ \forall i \in \mathcal{N}_v \Big\},\end{aligned} \tag{45}$$

which can be solved using a specialized min-max algorithm, such as the Polak-He algorithm (see Sect. 2.6.1 of [5]).

The function $\phi_3(r, \zeta)$ is continuous, but may not be differentiable in ζ. Hence the evaluation of $\bar{\psi}(r)$ requires the use of a derivative-free algorithm such as the Hooke-Jeeves algorithm [10].

5 Example: Drones under Centralized Control

Consider the case of N_v small drones in a holding pattern that is confined to a planar disk (i.e., they are allowed to fly in two dimensions only). We will assume that all the drones have the same dynamical model, with their state defined, for all $i \in \mathcal{N}_v$, by

$$x_i(t) = \begin{bmatrix} p_{xi}(t) \\ p_{yi}(t) \\ v_i(t) \\ \theta_i(t) \end{bmatrix}, \tag{46}$$

$$u_i(t) = \begin{bmatrix} a_i(t) \\ \delta_i(t) \end{bmatrix}, \tag{47}$$

$$h_i(x_i(t), u_i(t)) = \begin{bmatrix} v_i(t) \cdot \cos(\theta_i(t)) \\ v_i(t) \cdot \sin(\theta_i(t)) \\ a_i(t) \\ \delta_i(t) \end{bmatrix}, \tag{48}$$

where $p_{xi}(\cdot)$ and $p_{yi}(\cdot)$ are the Cartesian x-coordinate and y-coordinate, respectively, of the position of drone i, for a given arbitrary origin, $v_i(\cdot)$ is the speed, $\theta_i(\cdot)$ is the heading angle, $a_i(\cdot)$ is the acceleration, and $\delta_i(\cdot)$ is the yaw rate. In spite of its simplicity, this model describes quite well the dynamics of a fixed-wing aircraft, whose attitude, altitude, and forward speed are stabilized by an autopilot. It has been used previously in other applications, including air traffic management [11] and UAV trajectory planning [12]. Recall that, in this example, the position vector is

$$x_{iP}(t) = \begin{bmatrix} p_{xi}(t) \\ p_{yi}(t) \end{bmatrix} \in \mathbb{R}^2, \quad \forall i \in \{1, \dots, N_v\}. \tag{49}$$

For each $i = 1, \dots, N_v$, the speed $v_i(\cdot)$ was constrained to lie in the range $[v_{\min}, v_{\max}]$ with $v_{\min} = 13$ (m/s) and $v_{\max} = 16$ (m/s), to ensure the stability of the drones in the air. Also, the yaw rate was required to satisfy the inequality $|\delta_i(t)| \le 30$ (deg/s), and the acceleration was required to satisfy $|a_i(t)| \le 0.5$ (m/s^2).

Let the set $\mathcal{S}$ be defined by

$$\mathcal{S} \triangleq \left\{ \begin{pmatrix} p_x \\ p_y \\ v \\ \theta \end{pmatrix} \in \mathbb{R}^4 \;\middle|\; \left\| \begin{pmatrix} p_x \\ p_y \end{pmatrix} \right\| \le \rho \text{ and } v \in [v_{\min}, v_{\max}] \right\}, \tag{50}$$

Table 1 Results for arbitrary values of r

r (m)	$\bar{\psi}(r)$ (m)
23	3.6306
25	4.3654
27	4.6679

where $\rho > 0$ has to be chosen so that the drones have enough space to maneuver. To enforce the speed bounds, we defined $2N_v$ constraint functions by

$$f^j(\zeta, u) = \max_{t\in[0,T]} v_j(t) - v_{\max}, \tag{51}$$

$$f^{(N_v+j)}(\zeta, u) = \max_{t\in[0,T]} -v_j(t) + v_{\min}, \tag{52}$$

with $j = 1, \dots, N_v$, where $v_j(t)$ is the speed of j-th drone at time t. These constraints where approximated by

$$\bar{f}^j(\zeta, u) = \max_{k=0,\dots,N} \hat{v}_j(k) - v_{\max}, \tag{53}$$

$$\bar{f}^{(N_v+j)}(\zeta, u) = \max_{k=0,\dots,N} -\hat{v}_j(k) + v_{\min}, \tag{54}$$

where $\hat{v}_j(\cdot)$ is the approximation of the speed $v_j(\cdot)$ obtained by solving the difference equation (37).

The function $\bar{\phi}_3$ was computed using a modified version of Polak-He algorithm using ε-active sets (see Sect. 2.6.1 and Algorithm 2.6.23 of [5]), and the function $\bar{\psi}$ was computed using Hooke-Jeeves algorithm [10], with a new general step defined by random choice of a vector from $\mathcal{I}(r)$ randomly. The coordinate search in Hooke-Jeeves algorithm can be parallelized, and hence we ran the simulations using four cores of a 2 GHz Intel Xeon computer using the Parallel Computing Toolbox in MATLAB. We used LSSOL in [13] as the QP-solver for the internal iterations in Polak-He algorithm.

The parameters of the problem were chosen as $N_v = 4$, $\Delta = 7$ (s), $\rho_{\min} = 4$ (m), $\rho = 150$ (m), and $N_\Delta = 35$. The parameters for Polak-He algorithm, following the notation in Sect. 2.6.1 of [5], were chosen as $\delta = 1$, $\gamma = 10$, $\alpha = 0.1$, $\beta = 0.85$, and a value of $\varepsilon = 1000$ for the ε-active sets algorithm. Values of $\psi(r)$ for three values of r are shown in Table 1, where we can see that for $r = 23$ (m) the airplanes can get closer than the minimum distance allowed $\rho_{\min}$. Each simulation used 5000 vectors from $\mathcal{I}(r)$, chosen randomly, and took approximately 7 hours to complete.

In the worst-case scenario, i.e. at the initial conditions that achieve the minimum pairwise distance between drones, the trajectories for each drone are plotted at Fig. 2 in the case $r = 23$ (m). Note that the initial conditions of the drones shown in Fig. 2 are not symmetrical, with two of the drones (top and bottom of the figure) starting as close as possible and facing each other.

To test our results, we carried out a simulation using 4 drones flying in a bounded space, three of them flying in circles and a fourth drone joining them. We implemented the RHC Algorithm 2.1 using the models and parameters described above,

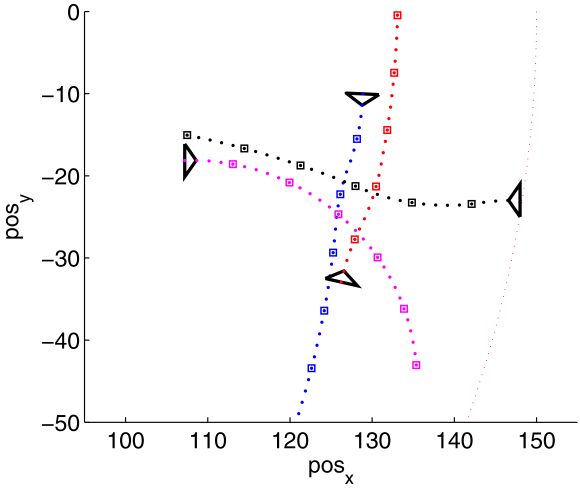


Fig. 2 Worst-case scenario when $r = 23$ (m). Drones are shown at their initial positions. Trajectories represent 3 (s) of flight time. Flight time between dots is 0.1 (s)

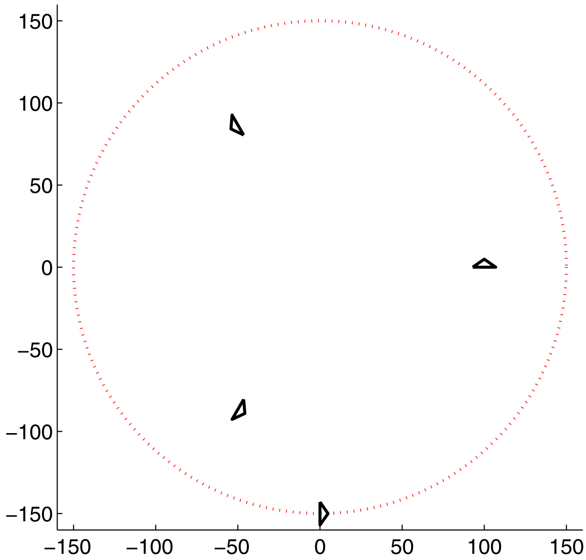


Fig. 3 Drones at $t = 0$ (s)

and added the following cost function:

$$f^0(\zeta, u) = \frac{1}{2}\sum_{i=1}^{N_v}\int_0^T \left(\|u_i(s)\|^2 + \left(\left\|x_{iP}^{(\zeta_i,u_i)}(s)\right\|^2 - \hat{r}^2\right)^2\right) ds, \tag{55}$$

where $\hat{r} = 100$ (m) is the radius of the formation circle. The positions of each drone and their trajectories are plotted for different times in Figs. 3 to 6. A video with the simulation is also available at http://www.eecs.berkeley.edu/~hgonzale/formation_flight.

Fig. 4 Drones at $t = 7$ (s)

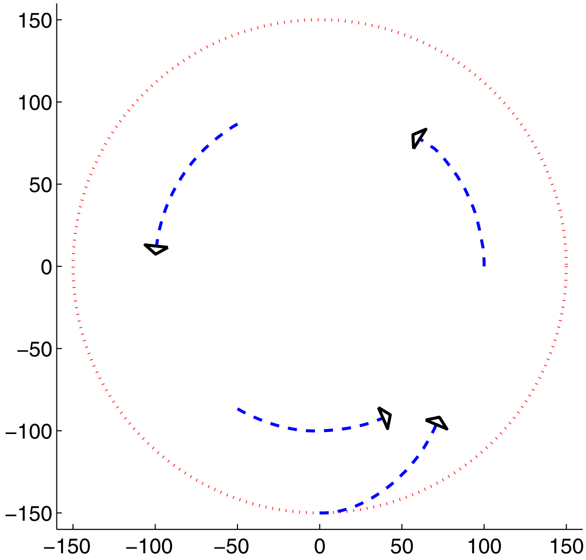
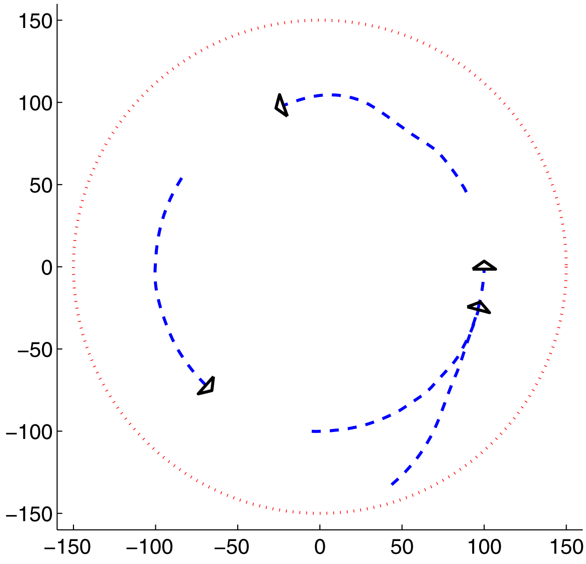


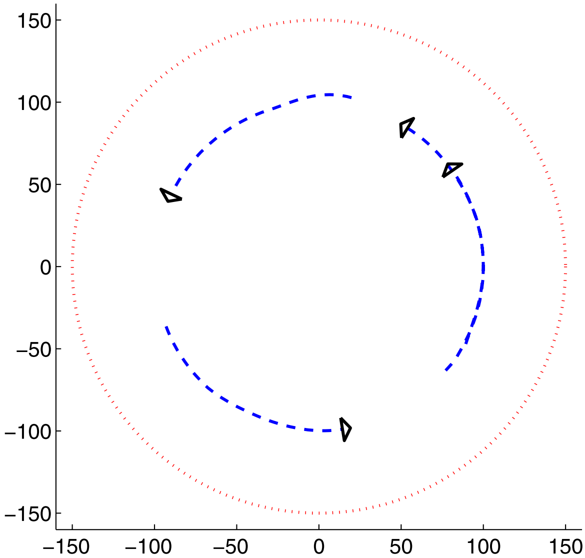
Fig. 5 Drones at $t = 14$ (s)



6 Conclusions

We have presented a sufficient condition that guarantees perpetual collision avoidance for a fleet of vehicles under centralized control. The sufficient condition can be phrased as follows: if the minimum initial distance between vehicles is greater than r, then there exists a control such that the vehicles do not collide for the next Δ seconds, and the minimum distance between vehicles at time Δ will be no less than r. Resorting to recursive feasibility arguments, we then concluded that the vehicles

Fig. 6 Drones at $t = 21$ (s)



would never collide. Although finding a satisfactory separation r requires many hours of computing time, this is not a serious drawback, since the computation need not be done in real time. Rather, once computed, the value r can be used at any time after that.

The results in this paper must be seen as a first step in developing a practical collision free, centralized control strategy. To allow for variations in ambient conditions, such as rising and receding tides, or wind conditions, different values of the minimum separation r will have to be precomputed for a variety, or ranges, of ambient conditions. In addition, model uncertainty will have to be accounted for. Finally, to allow for the unscheduled appearance of small craft, such as private planes, or small boats, any centralized control scheme will have to be supplemented with a freedom for individual vehicles in the fleet to carry out limited evasive maneuvers.

Acknowledgements We wish to thank Professor David Q. Mayne for his comments on the draft of this paper. This research was supported by ARO MURI SWARMS (W911NF-0510219).

References

1. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. Automatica **36**(6), 789–814 (2000)
2. Ge, S.S., Cui, Y.J.: Dynamic motion planning for mobile robots using potential field method. Auton. Robots **13**(3), 207–222 (2002)
3. Shim, D.H., Kim, H.J., Sastry, S.: Decentralized nonlinear model predictive control of multiple flying robots. In: Proceedings of 42nd IEEE Conference on Decision and Control, vol. 4, pp. 3621–3626, 9–12 Dec. 2003

4. Yang, T.H., Polak, E.: Moving horizon control of nonlinear systems with input saturation, disturbances and plant uncertainty. Int. J. Control **58**(4), 875–903 (1993)
5. Polak, E.: Optimization: Algorithms and Consistent Approximations. Springer, Berlin (1997)
6. Sun, J., Kolmanovsky, I.V., Ghaemi, R., Chen, S.: A stable block model predictive control with variable implementation horizon. Automatica **43**(11), 1945–1953 (2007)
7. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, New York (2002)
8. Kang, W., Ross, I.M., Gong, Q.: Pseudo-spectral optimal control and its convergence theorems. In: Analysis and Design of Nonlinear Control Systems, Part 2, pp. 109–124. Springer, Berlin (2008)
9. Chung, H., Polak, E., Sastry, S.S.: An accelerator for packages solving discrete-time optimal control problems. In: Proc. 17th IFAC World Congress, pp. 14925–14930, 6–11 July 2008
10. Hooke, R., Jeeves, T.A.: "Direct search" solution of numerical and statistical problems. J. ACM **8**(2), 212–229 (1961)
11. Mitchell, I.M., Bayen, A.M., Tomlin, C.J.: A time-dependent Hamilton-Jacobi formulation of reachable sets for continuous dynamic games. IEEE Trans. Automat. Contr. **50**, 947–957 (2005)
12. Kang, Y., Hedrick, J.: Design of nonlinear model predictive controller for a small fixed-wing unmanned aerial vehicle. In: AIAA Guidance, Navigation, and Control Conference (2006)
13. Holmström, K., Göran, A.O., Edvall, M.M.: User's guide for TOMLAB. Tomlab Optimization Inc., December 2006