Heading and Position Receding Horizon Control for Trajectory Generation

Eric Gagnon, C. A. Rabbath and Marc Lauzon
Defence R&D Canada - Valcartier
2459 Pie-XI Blvd North
Val-Belair, Qc, G3J 1X5, Canada

Abstract— **This paper presents a comparative study of two formulations of model predictive control with receding horizons for the cooperative control of a team of unmanned aerial vehicles. In the first formulation, the vehicle trajectories are solved dynamically as sequences of vehicle headings over prediction horizons and executed over shorter action horizons. This formulation takes advantage of an implementation of collision avoidance based on vehicle heading constraints. In the second formulation, the vehicle trajectories are solved as sequences of vehicle positions, rather than vehicle headings. This formulation handles collision avoidance with vehicle position constraints. An efficient branch-and-bound algorithm is proposed to support the mixed integer constraints, and a collision avoidance solution based on heading constraints is evaluated. This paper shows that both receding horizon formulations produce exactly the same vehicle trajectories when they are used without collision avoidance constraint. It is shown however that heading receding horizon control requires less computing power than position receding horizon control whether in situations of collision avoidance or not.**

I. INTRODUCTION

Cooperative control represents an important operational requirement of future unmanned aerial vehicles (UAV). Joint operational control of a UAV team comprises a pre-arranged policy of action and inter-team communication, which is decided *a priori* and adhered to by each member of the team [1]. The team is expected to carry out its mission to its fullest extent through autonomous joint decision making. Cooperative control represents an important enabler towards achieving the overarching goal of a cost-effective augmentation of operational capability and flexibility in complex theaters of operations, when compared with the employment of individual UAV platforms. These notions along with the continued need to assign the dull, dangerous and dirty missions to autonomous platforms (individual or team) are main drivers for current research in cooperative control [2],[3].

Several methods have been proposed to obtain cooperative control between UAV. One of these methods is known as model predictive control (MPC) with receding horizons [2],[4]. MPC allows for the control of multivariable systems as easily as for single input single output (SISO) systems. MPC also supports linear and nonlinear control, equality and inequality constraints, and offers extensive flexibility in the formulation of the cost function to be minimized.

In the present paper, MPC with receding horizons is used to determine dynamically the vehicle trajectories through the solution of sequential optimization problems. Real-time performance (fast computing) is critical in actual multi-UAV applications. Designers thus look for formulations that are simple to express and are numerically tractable. Two formulations are studied in this paper. In the first one, vehicle trajectories are solved as sequences of vehicle headings [2]. This formulation is more convenient when collision avoidance is done with vehicle heading constraints. With heading constraints, the optimization problem is convex, which simplifies its solution because no mixed-integer linear or nonlinear programming (MILP or MINLP) is required. In the second formulation, the vehicle trajectories are solved as sequences of vehicle positions rather than vehicle headings, wherein collision avoidance is realized with position constraints. However, position constraints usually lead to non-convex problems that are difficult to solve and require the use of MILP or MINLP solvers as is done in [4],[5],[6]. Nearly all MILP and MINLP problems rely on the successive application of linear and nonlinear programming in Branch-and-Bound (BB) algorithms. BB is the most popular approach and is currently used in virtually all commercial MILP and MINLP softwares [7].

This paper shows that heading and position receding horizon formulations produce exactly the same vehicle trajectories when they are used without collision avoidance constraint. Both formulations are thus equally applicable to cooperative teaming from this perspective. Formulations are also compared in collision avoidance situations with square and circular obstacles. An efficient BB algorithm is proposed to support mixed integer constraints of position receding horizon control (PRHC) with position constraints. Heading receding horizon control (HRHC) is used to evaluate a collision avoidance solution based on heading constraints [8]. Interestingly, HRHC is found to be significantly less computationally intensive than PRHC whether in situations of collision avoidance or not.

In this paper, section II presents the cooperative control problem. Section III describes the concept of MPC with heading and position receding horizons. The collision avoidance methods are compared in section IV. Some numerical examples are presented in Section V and concluding remarks are provided in Section VI.

0-7803-9098-9/05/$25.00 ©2005 AACC

II. COOPERATIVE CONTROL PROBLEM

The cooperative control problem formulation and its solution are presented for a 2D space representation, which can in turn be extended to 3D space in a straightforward manner. The 2D formulation facilitates visualization of the vehicle trajectories, while still representing a 3D problem where vehicles are assumed to operate at a constant altitude. Also, altitude is often determined by mission constraints, such as sensor resolution and radar visibility, simplifying the guidance to a 2D problem [9].

Let matrices $\boldsymbol{N}(t)\in\Re^{2\times n}$ (the ($2\times n$)-dimensional real matrix space) and $\boldsymbol{P}(t)\in\Re^{2\times p}$ represent temporal position sets in a 2D space of n UAV and p target locations defined as follows:

$$\boldsymbol{N}(t)=\begin{bmatrix} a_1(t) & b_1(t) \\ a_2(t) & b_2(t) \\ \vdots & \vdots \\ a_n(t) & b_n(t) \end{bmatrix}, \boldsymbol{P}(t)=\begin{bmatrix} \lambda_1(t) & \phi_1(t) \\ \lambda_2(t) & \phi_2(t) \\ \vdots & \vdots \\ \lambda_p(t) & \phi_p(t) \end{bmatrix} \qquad (1)$$

where $a_1(t)\ldots a_n(t)$ and $\lambda_1(t)\ldots\lambda_p(t)$ represent x-axis positions while $b_1(t)\ldots b_n(t)$ and $\phi_1(t)\ldots\phi_p(t)$ represent y-axis positions, in the inertial flat-earth coordinate system. One target is assigned per UAV as part of the task allocation process and represents a location that must be visited by this UAV, where the distance at all time must be minimized. The dynamics of each UAV and target are represented by a simple point-mass model, also known as the kinematics model of the unicycle-type robot [10]. The UAV kinematics are given as:

$$\dot{\boldsymbol{N}}(t)=\left[\boldsymbol{V}(t)\otimes\cos\boldsymbol{U}(t) \quad \boldsymbol{V}(t)\otimes\sin\boldsymbol{U}(t)\right] \qquad (2)$$

where ⊗ denotes the Hadamard or term-to-term product, and $\dot{\boldsymbol{N}}(t)$ denotes the derivative of $\boldsymbol{N}(t)$. $\boldsymbol{V}(t)$ and $\boldsymbol{U}(t)$ are vehicle velocity and vehicle heading vectors respectively, and can be expressed as:

$$\boldsymbol{V}(t)=\left[v_1(t) \quad v_2(t) \quad \ldots \quad v_n(t)\right]^{\mathrm{T}} \qquad (3)$$

$$\boldsymbol{U}(t)=\left[u_1(t) \quad u_2(t) \quad \ldots \quad u_n(t)\right]^{\mathrm{T}} \qquad (4)$$

where $u_i(t)\in[-\pi,\pi]$, $v_i(t)\in[v_{\min},v_{\max}], i=1,2,\ldots,n$ and T denotes the transpose operator. Given τ, the time interval between successive vehicle positions (update period), the vehicle positions at time $t+\tau$ can be approximated as:

$$\boldsymbol{N}(t+\tau)=\boldsymbol{N}(t)+\tau\cdot\left[\boldsymbol{V}(t)\otimes\cos\boldsymbol{U}(t) \quad \boldsymbol{V}(t)\otimes\sin\boldsymbol{U}(t)\right] \qquad (5)$$

For realism in the vehicle dynamics, the vehicle heading and velocity variations over a time interval τ have been constrained to:

$$\left|\boldsymbol{U}(t+\tau)-\boldsymbol{U}(t)\right|\le\boldsymbol{C} \qquad (6)$$

$$\left|\boldsymbol{V}(t+\tau)-\boldsymbol{V}(t)\right|\le\boldsymbol{E} \qquad (7)$$

where the vectors $\boldsymbol{E}\in\Re_+^n$ (the n-dimensional real positive vector space) and $\boldsymbol{C}\in\Re_+^n$ are constant and defined as:

$$\boldsymbol{C}=\left[c_1 \quad c_2 \quad \ldots \quad c_n\right]^{\mathrm{T}} \qquad (8)$$

$$\boldsymbol{E}=\left[e_1 \quad e_2 \quad \ldots \quad e_n\right]^{\mathrm{T}} \qquad (9)$$

In the case where collision avoidance is applied, the vehicles must avoid square and circular static obstacles. Use of pre-defined obstacle shapes rather than generic multi-faceted ones simplifies the formulation of the control problem. The m square static obstacles are implemented by considering their lower left corners $(x_{\mathrm{low},i}, y_{\mathrm{low},i})$ and upper right corners $(x_{\mathrm{high},i}, y_{\mathrm{high},i})$, $i=1,2,\ldots,m$. The obstacles are represented by the following matrix $\boldsymbol{M}\in\Re^{m\times 4}$:

$$M=\begin{bmatrix} x_{\mathrm{low},1} & y_{\mathrm{low},1} & x_{\mathrm{high},1} & y_{\mathrm{high},1} \\ x_{\mathrm{low},2} & y_{\mathrm{low},2} & x_{\mathrm{high},2} & y_{\mathrm{high},2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{\mathrm{low},m} & y_{\mathrm{low},m} & x_{\mathrm{high},m} & y_{\mathrm{high},m} \end{bmatrix} \qquad (10)$$

The q circular static obstacles are implemented by considering their center positions (σ_i, γ_i) and radii r_i, $i=1,2,\ldots,q$. The circular obstacles are represented by a matrix $\boldsymbol{R}\in\left[\Re^{q\times 2} \quad \Re_+^{q\times 1}\right]$ defined as:

$$\boldsymbol{R}=\begin{bmatrix} \sigma_1 & \gamma_1 & r_1 \\ \sigma_2 & \gamma_2 & r_2 \\ \vdots & \vdots & \vdots \\ \sigma_q & \gamma_q & r_q \end{bmatrix} \qquad (11)$$

The distances between the vehicle and target positions are represented by a matrix $\boldsymbol{D}(t)$, where each of its elements provides the distance between the j^{th} vehicle and the i^{th} target as:

$$d_{ij}(t)=\left[\left(a_j(t)-\lambda_i(t)\right)^2+\left(b_j(t)-\phi_i(t)\right)^2\right]^{1/2} \qquad (12)$$

where $j=1,2,\ldots,n$ and $i=1,2,\ldots,p$. In addition, time-dependent target location weights are employed to customize the cooperative control problem. These weights ensure that critical targets receive a higher priority, through a dynamic priority management. The target weights are specified through the vector $\boldsymbol{G}(t)\in\Re_+^p$, defined as:

$$\boldsymbol{G}(t)=\left[g_1(t) \quad g_2(t) \quad \ldots \quad g_p(t)\right]^{\mathrm{T}} \qquad (13)$$

where each $g_i(t)$ element, $i=1,2,\ldots,p$, represents a specific target weight. The mission objective consists of minimizing the distance at all time between the vehicle positions $\boldsymbol{N}(t)$ and target positions $\boldsymbol{P}(t)$ based on the target weights $\boldsymbol{G}(t)$. From (12) and (13), the cost function minimized is:

$$J = \sum_{j=1}^{n} \min_{i} \left[d_{ij}(t) / g_i(t) \right]; \quad i = 1,2,\ldots,p \qquad (14)$$

A nonlinear programming (NLP) solver is required to minimize this nonlinear cost function.

III. MPC WITH RECEDING HORIZONS

In a 2D space, a receding horizon can be represented as a circle centered on the vehicle. Two such temporal receding horizons are used here. The prediction horizon ($h_i, i = 1,2,\ldots,n$), which is unique for each vehicle, sets the temporal window in the future over which the cost function (14) is minimized, starting from the current time. The action horizon ($\vartheta_i, i = 1,2,\ldots,n$), which is also unique to each vehicle, sets the temporal window in the future over which the minimizers of (14) will be applied, starting from the current time. The action horizon of each vehicle must evidently be equal to or shorter than the prediction horizon. In addition, one spatial receding horizon ($H_i, i = 1,2,\ldots,n$) is considered here for each vehicle, which represents the limitation in range of the UAV on-board sensors, but also to constrain inter-vehicle communication. This means that two vehicles cannot communicate any information between themselves if their spatial horizons do not overlap. Also, the obstacle positions are assumed unknown outside the spatial horizons while the target positions are known.

A. Heading Control

Using the proposed set of receding horizons, the objective of the first MPC control formulation, namely heading control, is to compute the future vehicle headings that will minimize (14) over the prediction horizons, with periodic application of the solution within the action horizons. From (14), the selected MPC formulation with heading receding horizons can be written as:

$$\Psi = \min_{\substack{\boldsymbol{U}(t+k\tau) \\ \boldsymbol{V}(t+k\tau)}} \left[\sum_{j=1}^{n} \min_{i} \left[d_{ij}(t + h_j \tau) / g_i(t + h_j \tau) \right] \right] \qquad (15)$$

where $k = 1,2,\ldots,h_i$, $i = 1,2,\ldots,p$ and $\tau \in \Re_+$. Equation (15) shows that when the vehicles fly at constant velocities, the number of parameters optimized for each vehicle is equal to the size of the prediction horizon. If vehicle velocities are not constant, but are limited to a range $[v_{\min}, v_{\max}]$, the number of parameters optimized for each vehicle equals twice the size of the prediction horizon.

B. Position Control

Using the same receding horizons, another MPC control objective considered in this paper is to find the vehicle positions that will minimize (14) over the prediction horizons, with periodic action taken over the action horizons. From (14), the MPC formulation with position receding horizons can be written as:

$$\Psi = \min_{\boldsymbol{N}(t+k\tau)} \left[\sum_{j=1}^{n} \min_{i} \left[d_{ij}(t + h_j \tau) / g_i(t + h_j \tau) \right] \right] \qquad (16)$$

Here, the number of parameters optimized for each vehicle equals twice the size of the prediction horizon, no matter the vehicle velocities. Based on the number of parameters optimized, when the UAV travel at constant velocities without obstacle, HRHC is expected to require less computing power than PRHC.

IV. COLLISION AVOIDANCE

Vehicles must react rapidly when obstacles are detected as they appear within their sensor range, to avoid any collision. Collision avoidance, based on MPC with receding horizons, is implemented through specific constraints in the formulation of the cooperative control problem. Heading constraints are naturally more convenient to use with HRHC and position constraints are better suited to PRHC.

A. Heading Constraints

Heading constraints specify directions which must be avoided by the vehicles. One approach for setting heading constraints in the optimization of (15), in the case of circular obstacles, is proposed in [8]. Specifically, inequality constraints are computed using the intersection points between the vehicle spatial horizon h_i and the obstacle circular border. When a vehicle spatial horizon h_i overlaps an obstacle, a target heading constraint is chosen between two possible headings s_1 and $s_2 \in [-\pi, \pi]$ resulting from the line joining the intersection points. Based on the current vehicle heading and the heading $f_i \in [-\pi, \pi]$ resulting from the line projected from the vehicle position to the obstacle center position, the set $\boldsymbol{Q}$ of inequality constraints, which is used in the optimization problem, is built as:

$$\begin{aligned}
&\bar{u}_i(t + k\tau) \geq \left.\begin{cases} u_i(t) + kc_i & \text{if } u_i(t) + kc_i < s_1 \\ s_1 & \text{if } u_i(t) + kc_i \geq s_1 \end{cases}\right\} \begin{matrix} \text{if } u_i(t) - f_i \geq 0 \\ ; k = 1,2,\ldots,h_i \end{matrix} \\
&\bar{u}_i(t + k\tau) \leq \left.\begin{cases} u_i(t) - kc_i & \text{if } u_i(t) - kc_i > s_2 \\ s_2 & \text{if } u_i(t) - kc_i \leq s_2 \end{cases}\right\} \begin{matrix} \text{if } u_i(t) - f_i < 0 \\ ; k = 1,2,\ldots,h_i \end{matrix}
\end{aligned} \qquad (17)$$

where c_i represents the vehicle heading variation constraint and $\bar{u}_i(t + k\tau)$ the future vehicle heading over the prediction horizon h_i. It is noted that the solution proposed in [8], can be applied to the case of irregular obstacle shapes, as long as the intersection points with the circular vehicle spatial horizon are available from the vehicle sensors.

B. *Position Constraints*

In a 2D space, the position constraints determine the plane surfaces that must be avoided by the vehicles. In the case of rectangular obstacles, four constraints are required to define each obstacle. Each vehicle cannot satisfy the four constraints of an obstacle. But, each vehicle needs to satisfy only one such constraint to obtain collision avoidance with an obstacle. Accordingly, to include collision avoidance with rectangular obstacles in the optimization of (16), a set of constraints among (18) needs to be satisfied before each displacement of the vehicles over their action horizons.

$$\left.\begin{array}{l} a_i(t+k\tau)\le x_{\text{low},j} \\ b_i(t+k\tau)\le y_{\text{low},j} \\ a_i(t+k\tau)\ge x_{\text{high},j} \\ b_i(t+k\tau)\ge y_{\text{high},j} \end{array}\right\} \begin{array}{l} ;\, k=1,2,\ldots,h_i \\ ;\, i=1,2,\ldots,n \\ ;\, j=1,2,\ldots,m \end{array} \qquad (18)$$

Equation (18) assumes that the obstacles are aligned with the XY coordinate system. In this way, through the displacements and the spatial horizons of the vehicles, some constraints are turned off whereas others are turned on. With this addition of binary constraints in the optimization of (16), the cooperative control problem becomes a MINLP problem. An efficient BB approach, described next, is proposed to solve this problem. This approach has been proposed because unspecialized BB algorithms are not recommended to minimize computing time.

C. *Branch-and-Bound*

Branch and bound (BB) is a class of methods for linear and nonlinear mixed-integer programming. If carried to completion, it is guaranteed that an optimal solution is found, however it is a time consuming process. The requirement of the BB implementation is to choose rapidly the proper constraints among the set given by (18) as a function of the displacements and the spatial horizons of the vehicles. The method proposed consists of computing, for each predicted trajectory point, the distances $\partial_{l,i,j,k}, l=1,2,\ldots,4$ which can be seen inside the spatial horizon, as:

$$\left.\begin{array}{l} \partial_{1,k,i,j}=x_{\text{low},j}-a_i(t+k\tau) \\ \partial_{2,k,i,j}=y_{\text{low},j}-b_i(t+k\tau) \\ \partial_{3,k,i,j}=a_i(t+k\tau)-x_{\text{high},j} \\ \partial_{4,k,i,j}=b_i(t+k\tau)-y_{\text{high},j} \end{array}\right\} \begin{array}{l} ;\, k=1,2,\ldots,h_i \\ ;\, i=1,2,\ldots,n \\ ;\, j=1,2,\ldots,m \end{array} \qquad (19)$$

In (19), positive distances indicate satisfied constraints. Also, the distances that require immediate attention are given by the maximum values of (19), which are defined as:

$$\wp_{k,i,j}=\max[\partial_{1,k,i,j} \quad \partial_{2,k,i,j} \quad \partial_{3,k,i,j} \quad \partial_{4,k,i,j}] \qquad (20)$$

Based on (20), the set of constraints requiring immediate concern is given as:

$$\left.\begin{array}{l} a_i(t+k\tau)\le x_{\text{low},j};\, \mathit{if}\ \wp_{k,i,j}=\partial_{1,k,i,j} \\ b_i(t+k\tau)\le y_{\text{low},j};\, \mathit{if}\ \wp_{k,i,j}=\partial_{2,k,i,j} \\ a_i(t+k\tau)\ge x_{\text{high},j};\, \mathit{if}\ \wp_{k,i,j}=\partial_{3,k,i,j} \\ b_i(t+k\tau)\ge y_{\text{high},j};\, \mathit{if}\ \wp_{k,i,j}=\partial_{4,k,i,j} \end{array}\right\} \begin{array}{l} ;\, k=1,2,\ldots,h_i \\ ;\, i=1,2,\ldots,n \\ ;\, j=1,2,\ldots,m \end{array} \qquad (21)$$

From the vehicle trajectory optimized over the prediction horizon without collision avoidance constraint, if the nearest predicted trajectory point to the vehicle satisfies its constraints (21), these constraints are activated and the next trajectory point is analyzed similarly. This procedure is repeated until a predicted trajectory point is found which does not satisfy one of its immediate constraints (21). In this case, this constraint is activated, but a new vehicle trajectory is optimized over the prediction horizon with the constraints activated. Using the updated predicted trajectory, a new set of immediate constraints (21) is obtained and the predicted trajectory points are reanalyzed similarly. This strategy is repeated for each vehicle until all the predicted trajectory points over the prediction horizon respect their constraints (21), which is the solution of the cooperative control problem at current time.

V. Numerical Examples

A. *Moving Targets Without Obstacle*

The first example compares HRHC and PRHC with constant vehicle velocities, moving targets and without obstacle. In Fig. 1, two vehicles with initial positions $(-200,-100)$ and $(-80,-100)$ and initial headings of 0 radian, represented as two small circles, travel at constant velocities $v_i(t)$ of 60 km/h. Constraints c_i have been set to $10\pi/180$ radian. Two small square targets with initial positions $(-200,100)$ and $(100,50)$ and headings of –0.4636 radian and 2.8966 radians move linearly at 134.2 km/h and 123.7 km/h respectively. Even if it is impossible for the vehicles to reach these fast targets, this situation is interesting for the study of receding horizon control performance, in particular, computational requirement. The targets have constant weighting values $g_1(t)$ and $g_2(t)$ of 1 and 2 respectively. H_i, ϑ_i and h_i, have been arbitrarily set to 20 km, 1-step and 20-steps respectively. The cooperative control problem has been simulated over 150 min with an update period τ of 1 min, which produces a 1 min action horizon and a 20 min prediction horizon. Fig. 1 shows the vehicle trajectories obtained with both HRHC and PRHC. The trajectory differences are negligible and not distinguishable in the figure.

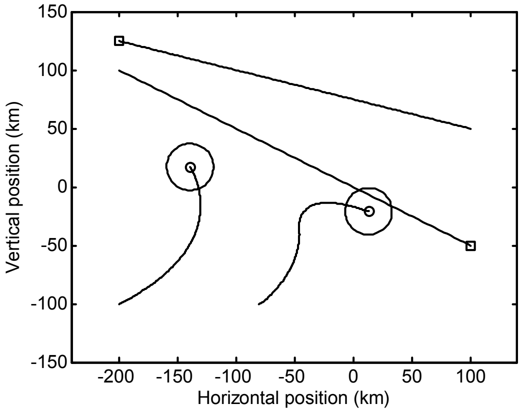

Fig. 1 Cooperative control of two UAV with target linear displacements

Fig. 2 shows bar charts of the processing time required by each iteration of the simulations (150 bars are shown). The top chart of Fig. 2 shows the HRHC simulation and the bottom one shows the PRHC results. The mean value (Mean) and standard deviation (Std) of each bar chart are shown as additional information.

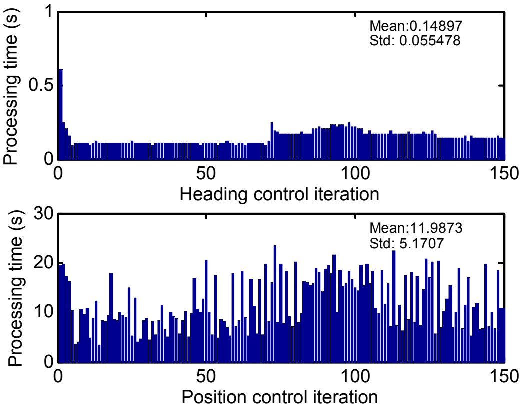

Fig. 2 Processing time with target linear displacements

The same solver (medium-scale optimization with the Fmincon Matlab function) with a 2.4 GHz Intel Xeon processor has been used with both receding horizon formulations. The optimization starting trajectories were straight lines based on the current time vehicle headings. As anticipated, PRHC consistently requires a larger processing time than HRHC and the difference observed is very important.

B. Static Targets With Obstacles

Example 2 compares HRHC and PRHC with circular and square obstacles respectively. Two vehicles with initial positions $(-50,-100)$ and $(230,-100)$ and initial headings of $\pi/2$ radian, represented again by small circles, travel at constant velocities $v_i(t)$ of 60 km/h. Two static and small square targets at position $(50,100)$ and $(170,110)$ with both a constant weighting value of 1 are also included in this scenario. Three static square obstacles have been added and defined as:

$$\boldsymbol{M} = \begin{bmatrix} -10 & -30 & 50 & 30 \\ -50 & 60 & 30 & 140 \\ 150 & -50 & 230 & 30 \end{bmatrix} \tag{22}$$

Also, three static circular obstacles have been added over the square obstacles and defined as:

$$\boldsymbol{R} = \begin{bmatrix} 20 & 0 & 33.851 \\ -10 & 100 & 45.135 \\ 190 & -10 & 45.135 \end{bmatrix} \tag{23}$$

Each circular obstacle has the same surface as its twin square obstacle. The constraints c_i on the vehicle heading variations, and the vehicle spatial, action and prediction horizons have been set to $10\pi/180$ radian, 15 km, 1-step and 15-steps respectively. Here, the cooperative control problem has been simulated over 220 min with an update period τ of 1 min, which produces a 1 min action horizon and a 15 min prediction horizon.

Fig. 3 shows HRHC results with heading constraints (17) for circular obstacle collision avoidance. Each vehicle tracked its nearest target, which is as expected since the targets have the same weighting value. Also, the vehicles have reached the targets while avoiding any collision with the circular obstacles, the avoidance maneuvers taking place when their horizons H_i intersected the obstacles.

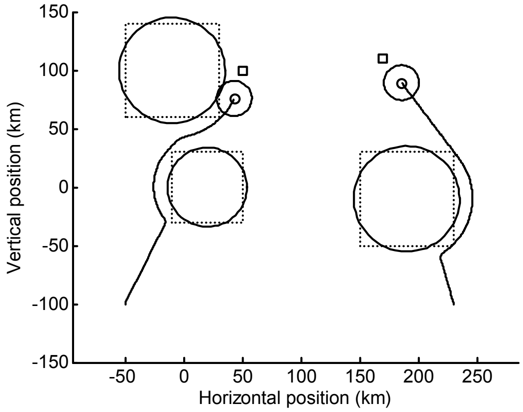

Fig. 3 HRHC with collision avoidance based on heading constraints

Fig. 4 presents the PRHC results with the position constraints (18). The proposed BB solution based on (19), (20) and (21) has been used to obtain collision avoidance with the square obstacles. In Fig. 4, the vehicle on the right has followed a trajectory which did not reach its target after 220 iterations as opposed to the same vehicle in Fig. 3. This is a result of flying in an unknown environment with a finite spatial horizon. Also, with larger circular obstacles, HRHC could avoid the square obstacle corners, and with larger square obstacles, PRHC could avoid the circular obstacles.

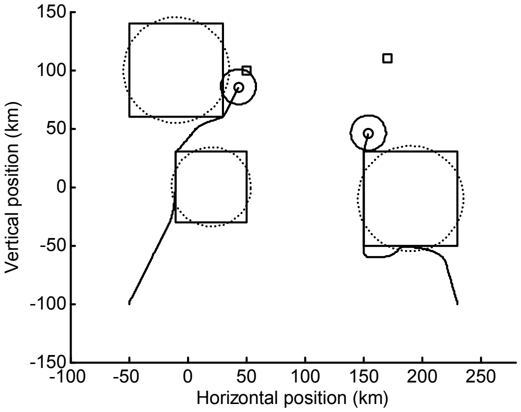

Fig. 4 PRHC with collision avoidance based on position constraints

From Fig. 5, PRHC requires yet larger processing time than HRHC as anticipated, but the difference becomes more significant when collision avoidance is required because of the MINLP problem generated by PRHC with position constraints. In Fig. 5, collision avoidance situations appear roughly between the 40th and 200th iteration in the PRHC case and from the 40th until the end in the HRHC case.

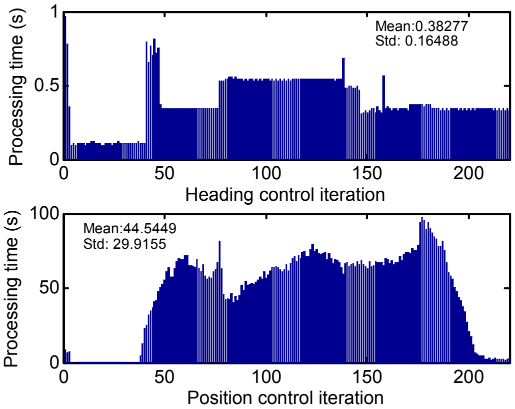

Fig. 5 Processing time with constant vehicle velocities

With the objective of making more complex the cooperative control problem, a second scenario repeats the first one but without constant vehicle velocities. The vehicle velocities have been limited between 30 km/h and 60 km/h with initial values of 30 km/h for both vehicles. Constraints e_i on the vehicle velocity variations over the time interval τ have been set to 6 km/h.

The results show trajectories similar to Fig. 3 and Fig. 4, and vehicle velocity variations were observed, the vehicles increasing rapidly their velocities to the maximum value in the first simulation steps. Fig.6 shows the processing time results. As anticipated, the HRHC has increased its processing time, but so did the PRHC, which still shows significantly larger processing time than HRHC.

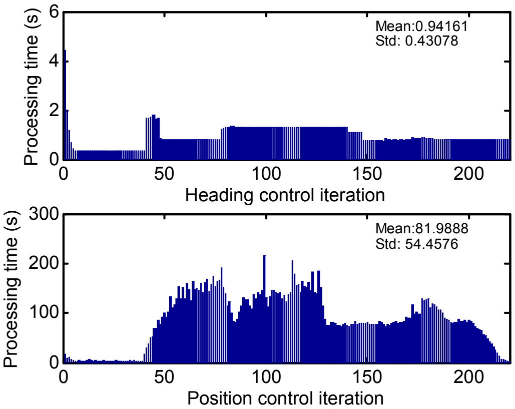

Fig. 6 Processing time with varying vehicle velocities

VI. CONCLUSION

This paper compared heading and position receding horizon for the cooperative control of unmanned aerial vehicles. It was shown that these two receding horizon formulations produce very similar vehicle trajectories when they are used in a simplified context without collision avoidance constraint. However, the problem formulated as a HRHC required significantly less computational effort than that formulated as a PRHC. With collision avoidance, PRHC with position constraints increases greatly its processing time compared with HRHC with heading constraints because of the MINLP problem generated. Based on these results, HRHC with heading constraints is more appropriate for embedded applications in unknown environments, where the threats and obstacles are discovered progressively as the vehicles move forward.

REFERENCES

[1] C.A. Rabbath, E. Gagnon and M. Lauzon, “On the Cooperative Control of Multiple Unmanned Aerial Vehicles”, *IEEE Canadian Review*, Spring 2004, 46, pp.15-19.

[2] C.G. Cassandras and W. Li, “A Receding Horizon Approach for Solving Some Cooperative Control Problems”, *IEEE Conf. on Decision and Control*, 2002, pp.3760-3765.

[3] D.A. Schoenwald, “AUVs: In space, air, water and on the ground”, *IEEE Control Systems Magazine*, December 2000.

[4] J.S. Bellingham, A.G. Richards and J.P. How, “Receding Horizon Control of Autonomous Aerial Vehicles”, *American Control Conf.*, 2002, pp.3741-3746.

[5] J. Bellingham, Y. Kuwata and J. How, “Stable Receding Horizon Trajectory Control for Complex Environments”, *AIAA Guidance, Navigation, and Control Conf.*, 2003, Paper 2003-5635.

[6] A. Richards, Y. Kuwata and J. How, “Experimental Demonstrations of Real-time MILP Control”, *AIAA Guidance, Navigation, and Control Conf.*, 2003, Paper 2003-5802.

[7] T.F. Edgar and D.M. Himmelblau, *Optimization of Chemical Processes*, McGraw-Hill, New York, 2001.

[8] E. Gagnon, C.A. Rabbath and M. Lauzon, “Receding Horizons with Heading Constraints for Collision Avoidance”, *Submitted to AIAA Guidance, Navigation, and Control Conf.*, 2005.

[9] A. Richards and J.P. How, “Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming”, *Proc. of the American Control Conf.*, may 2002.

[10] J.A. Marshall, M.E. Broucke and B.A. Francis, "A Pursuit Strategy for Wheeled-vehicle Formations", *IEEE Conf. on Decision and Control*, 2003, pp.2555-2560.