Receding Horizon Control for Free-Flight Path Optimisation

Xiao-Bing Hu and **Wen-Hua Chen**

Department of Aeronautical and Automotive Engineering

Loughborough University

Loughborough

Leicestershire LE11 3TU

UK

Tel.: +44 1509 227230

Fax: +44 1509227275

Email: X.Hu@lboro.ac.uk, W.Chen@lboro.ac.uk

Abstract: This paper presents a Receding Horizon Control (RHC) algorithm to the problem of on-line flight path optimization for aircraft in a Free Flight (FF) environment. The motivation to introduce the concept of RHC is to improve the robust performance of solutions in a dynamic and uncertain environment, and also to satisfy the restrictive time limit to the real-time optimization of this complicated air traffic control problem. Firstly, the mathematical model for the on-line FF path optimization problem is set up and discussed. Then, the proposed RHC algorithm is described in details. Simulation results illustrate that the new algorithm is very efficient and promising for practical applications. While achieving almost the same optimal solution as an existing algorithm in the absence of environmental uncertainties, it works better in a dynamic and uncertain environment. In either case, the online computational time of the proposed RHC algorithm is only a fraction of that of the existing algorithm.

Keywords: Free Flight, Air Traffic Control, Receding Horizon Control, Genetic Algorithm, Optimization.

1 Introduction

The last couple of decades have witnessed the continuously rapid increase in air traffic around the world, further fast growth of both air travel and cargo shipment is projected in the near future, and the existing air traffic control (ATC) infrastructure has been struggling to keep things going under large amounts of endless criticisms in terms of safety, capacity, flexibility and efficiency (Benoit, 1994, Pelegrin, 1994, Wickens, et al, 1998, Kahne, 2000, McLean, 2003). Consequently, a lot of attentions have been attracted and many efforts have been made to either improve the existing systems or develop new ones to attack these problems in the ATC area. One of the most ambitious and promising schemes in the development and innovation of future aviation concepts and systems is the so-called “Free Flight (FF)” (Pelegrin, 1994, Wickens, et al, 1998, Kahne, 2000, McLean, 2003). “Simply put, FF is the safe, efficient movement of air traffic resulting from the coordinated actions of pilots, air traffic controllers, dispatchers and planners, and traffic flow specialists” (Wickens, et al, 1998). The traditional approach to managing air traffic is characterized by central control of flight operations by ground based personnel supported by ground-based technology. Differently, FF will feature collaborative decision making among pilots, ground based controller personnel, and air line operations control centers – all supported by space based technology with significant airborne and ground components. The effective realization of FF requires advances in communications, navigation, surveillance (CNS), and human factors technology and procedural changes. Some investigations required to support these advances have been reported in literature in recent years (Braune, et al, 1996, Wickens, et al, 1998, Kahne, 2000, Hoekstra, et al, 2002, McLean, 2003).

One of the primary features of FF is allowing pilots to change routes, with respect to safety, efficiency and flexibility, in real time without consulting with ATC (Hu, et al, 2004). As is well known, the current air traffic system is characterized by structured airspace, where aircraft fly predefined routes by using ground-based CNS stations and rudimentary decision support, with limited collaboration between ATC agencies and aircraft. This traditional structured airspace has proved to be a bottleneck for further improving air traffic capacity and efficiency to cope with the rapid increase in air traffic volume. Hopefully, under the FF scheme, the structured airspace will be discarded, and the pilots can decide and fly their preferred routes in the entire non-conflict-airspace, as illustrated in Figure 1 in an intuitive way.

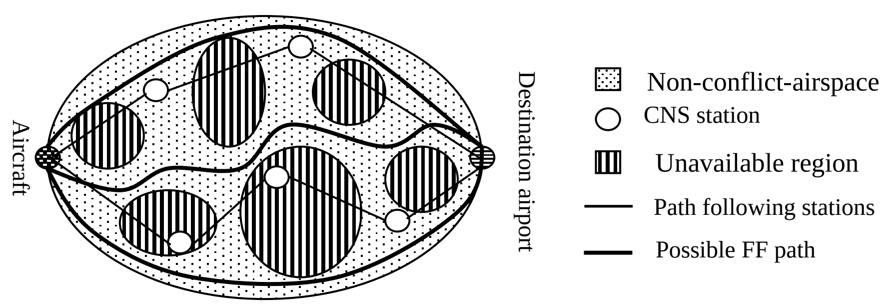

Figure 1.   Structured airspace & FF scheme

Many researches have now been under the way to improve the onboard capability of deciding user-preferred trajectories in an FF environment, particularly, optimizing the flight path in terms of safety and efficiency. Most of them put emphases on attacking the problem of conflict detection and resolution, presenting many interesting methods such like geometric approach (Geser and Munoz, 2002), mixed integer linear programming (Pallottino et al, 2002), token allocation strategy (Granger et al, 2001), Semi-Definite Program relaxation approach (Oh, 1999), linear matrix inequalities (Shewchum et al 1997), and genetic algorithm (GA) (Durand et al 1995). However, these results for conflict detection and resolution are medium-term or even short-term strategies to determine or optimize flight trajectories, and safety, compared with flight costs, is the overwhelming concern in the decision procedures. Whether or not they are suitable for long-term flight trajectory optimization still remains as an open question, because, for long-term flight trajectory optimization, say, inter-continental flight trajectory optimization, flight costs such as fuel cost and/or time cost are among the main concerns. From a practical viewpoint, since detecting and resolving conflicts globally and precisely in a dynamic environment is very time-consuming and then unrealistic for implementations, flight costs usually replace safety and become the major concern in long-term flight trajectory optimization. Safety separations are usually taken into account as constraints based on available information of air traffic. Once safety problem arises in a medium-term or short-term, the above mentioned methods can be adopted. Therefore, issues other than conflict detection and resolution become the main interest of most literature on long-term flight trajectory optimization. For example, Warren and Schwab (1997) focuses on validating the practicability of optimal flight path, and Plaettner and Zhao (2000) and McDonald and Zhao (2000) on analyzing its theoretic benefits.

Optimizing flight path under FF to minimize a certain flight cost with safety constraints is a very difficult problem. Challenges come from three aspects. Firstly, it is not a convex optimization problem and exhibits significant nonlinearalities. Secondly, the real-time optimization suffers from heavy computational burden and restrictive time limit, especially in the case of long-distance flight. Thirdly, the real flight environment is dynamic and uncertain. In general, the optimum solution at each time instant does not necessarily make the actual flight cost minimized.

In McDonald and Zhao (2000), a combined function and parameter optimization algorithm, an off-line algorithm, was given to find flight trajectories that take advantage of atmospheric conditions in a theoretical study, which ignored many other factors affecting actual flight like safety constraints. Valuev and Velichenko (2002) developed a branch-and-bound algorithm, which is able to find an approximate solution to the trajectory optimization problem with respect to the flight cost in a specified air environment having some static and dynamic domains prohibited for flights. The algorithm was claimed to be of use for planning the departure of all long-distance civil airplanes over vast regions. Based on the integration of a heuristic algorithm with an integer linear programming model, an exact algorithm was reported in Andreatta et al (2000) to calculate departure time, flight route and speed, such that the arrival at the destination airport matches a specified time decided by the central authority. Hu et al (2004) proposed an improved GA-based approach to conduct online flight path optimization under FF, where dynamic unavailable regions and several kinds of flight cost were considered. The approach was claimed to be very effective to find optimal or near-optimal solutions. However, all these algorithms and approaches can hardly match to the challenges regarding real-time properties and robust performance in a dynamic environment.

This paper presents a novel algorithm based on the concept of Receding Horizon Control (RHC), or Model Predictive Control (MPC), to solve the online flight path optimization problem in a dynamic FF environment. Simply speaking, RHC is an *N*-step-ahead online optimization strategy. At each time interval, based on current available information, RHC optimizes the concerned problem for the next *N* intervals into the near future, and only the part of solution corresponding to current interval is implemented. At the next interval, RHC repeats the similar optimizing procedure for another *N* intervals into the near further based on updated information. RHC has now been widely accepted in the area of control engineering, and proved to be very successful regarding its many advantages against other control strategies Clarke (1994). Recently, attentions have been paid to applications of RHC to those areas like management and operations research. For example, theoretical research work on how to apply MPC to a certain class of discrete-event systems was presented in De Schutter and Van Den Boom (2001), and many practical implementations of rolling horizon strategy in the area of management were reported in Chand et al (2002). However, as mentioned in Chen et al (2002), the research work on applying RHC to areas other than control engineering is just at the beginning.

In the field of air traffic management, by intuitions, methodologies based on freeze horizon, influence horizon, optimization interval or similar ideas are used to resolve the problem of arrival sequencing and scheduling (ASS) in a dynamic fashion (Neuman and Erzberger 1991, Pelegrin, 1994, Schick, 1998). However, little insight is provided about how to design methodology-related parameters or what are the influences of these parameters on performance and robustness. Most recently, Hu and Chen (2005) reports an attempt to systematically study how to apply RHC strategy effectively to the dynamic ASS problem. However, to the best of the authors' knowledge, no systematical work has ever been reported to introduce the concept of RHC into the problem of online flight path optimization under FF, which is exactly the topic of this paper. The main motivations for using RHC are: first, to improve the real-time property such that the applications of proposed algorithm is practicable no matter how long the flight distance is, and second, to guarantee robust performance in a dynamic environment. The length of the receding horizon is the key issue not only to achieve computational efficiency, but also to make a proper trade-off between useful information for the near future and unreliable information for the far future in a dynamic environment. Terminal weighting terms in the performance index, which has never appeared in any existing literature on the problem of flight path optimization, are introduced and prove to be vital to guarantee stability and robust performance of the proposed RHC algorithm.

The remainder of this paper is organized as following. Section 2 describes the problem of online flight path optimization in an FF environment. The details of the proposed RHC algorithm are presented in Section 3. The efficiency of the algorithm is demonstrated by simulation results in Section 4. The paper ends up with some conclusions in Section 5.

2 Online flight path optimization problem in a Free Flight environment

In the real world, ATC agencies collect various data and information such as weather conditions and air traffic flows. Then, they broadcast information like weather conditions, calculate constraints/criteria for the sake of safety, efficiency and capability, and issue them to each individual aircraft. This paper assumes that the constraints/criteria issued by ATC agencies are unavailable-regions. The online optimization of flight path in this paper is defined as that, based on the information from ATC agencies, how a commercial aircraft finds out the optimal flight path from its non-conflict-airspace in real time to minimize a specified index.

2.1 Optional free flight paths

In an ideal FF environment, there are numerous optional free flight paths, which makes it very difficult, if not impossible, to finish computation in an acceptable period of time. Therefore, it is necessary to make reasonable and appropriate simplifications to the original problem. Using the traditional structured airspace is an easy way to simplify the problem, but it can hardly lead to the optimal solution in an FF sense because aircraft have to fly within the structured flight path network. Like in Hu et al (2004), in this paper, based on a set of discrete optional heading and the concept of "time-slice", the non-conflict-airspace is transformed into a dynamic flight path network such that a proper trade-off can be achieved between the simplification of problem and the optimality of the solution found by the proposed algorithm.

Using a set of discrete values to represent the optional headings is one of the key techniques to discretize the non-conflict-airspace. Instead of the original infinite heading set, a subset of finite discrete optional headings is assumed as

$$\Omega = [\,0^\circ, 10^\circ, 20^\circ, \cdots, 350^\circ, \theta_{dire}\,] \qquad (1)$$

where θ_{dire} is the direct-heading, which is defined as the direction of the destination airport with reference to the trajectory point the aircraft will reach at the end of the current time-slice. Every time when a heading needs to be determined, only these 37 values in Ω are available.

With the concept of "time-slice", the air traffic system is supposed to operate in the following manner. The ground ATC system transmits periodically both environment data and unavailable-region data to each individual aircraft. This period is called a "time-slice". Each individual aircraft uses the currently updated information to optimize the remaining flight path starting from the next time-slice. An optional flight path is composed of a series of sub-trajectories associated with time-slices. The sub-trajectory for the current time-slice is determined by the previous run of optimization. The optimization is based on sub-trajectories, as illustrated in Figure 2.

If a time-slice is too long and the optional headings in Ω are too less, the discrete non-conflict-airspace may become similar to the traditional structured airspace, which can hardly contain the globally optimal flight path. On the other hand, if the time-slice is too short and the optional headings are too many, there will be a huge number of optional free flight paths, and consequently, the online computational time for finding the optimal solution will be greatly increased, which might be unrealistic for real-time implementation. Like in Hu et al (2004), in this paper, a time-slice is assumed to be 10-minute long, and the set Ω given by (1) provides necessary and sufficient optional headings.

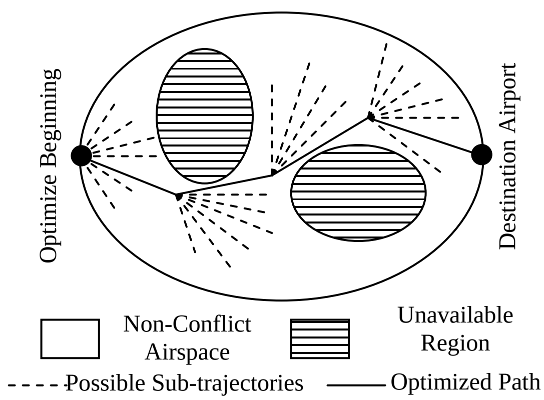

Figure 2 Optimized path in an FF environment

2.2 Performance index for flight path optimization

In this paper, for the sake of simplification, only flight time cost is chosen as the index for flight path optimization. Flight time cost can usually be easily transformed into another useful index for flight path optimization: fuel cost. As is well known, corresponding to a specified cruise altitude, each individual aircraft has an optimum cruise Mach number which leads to not only the minimum fuel cost rate but also the engine's optimum working conditions and maintenance. It is assumed that the cruise altitude is fixed for each individual aircraft, and the corresponding optimum cruise Mach and fuel cost rate can be checked out from a tabulated data. This optimum cruise Mach is used to calculate the flight time cost along an optional path. The total fuel cost along the optional path can then be determined by multiplying the flight time cost with the fuel cost rate parameter.

As defined in Subsection 2.1, an optional flight path is composed of a series of sub-trajectories. The flight time for each sub-trajectory (except the last sub-trajectory in the optional flight path) is supposed to be a time-slice, i.e., 10 minutes, and then, the total flight time for optional flight path is determined by the number of sub-trajectories included. Therefore, although the index is flight time cost, the basic variables for the online optimization are the coordinates of beginning point and end point of sub-trajectories. These basic variables and some important parameters are depicted in Figure 3, where (x,y) are the coordinates of a point, S_{AB} is the distance between point A and B, (φ, v) are the wind heading and speed at a point, θ denotes a certain heading depending on the subscript, and all headings are with respect to the direction of north. Strictly speaking, it is impossible to calculate the coordinates of the end point of a sub-trajectory, i.e., (x_B,y_B), because (x_B,y_B) and (φ_B, v_B) are prerequisites to each other. However, since a sub-trajectory is very short as the result of the 10-minute-long time-slice, it is reasonable to assume that the average wind parameters along the sub-trajectory are considered as the same as those at the beginning point, i.e., (φ_A, v_A). In other word, (φ_B, v_B) are not required for computing (x_B,y_B) under this assumption.

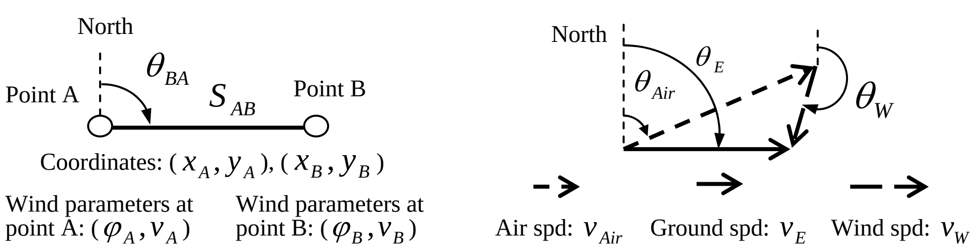

Figure 3. Variables and parameters of a sub-trajectory & related speeds

The coordinates of the end point of a sub-trajectory, (x_B,y_B), are calculated by

$$x_B = x_A + S_{AB}\cos\theta_{BA},\ \ y_B = y_A + S_{AB}\sin\theta_{BA}, \tag{2}$$

where

$$S_{AB} = v_E T_{ts},\ \ \theta_{BA} = \theta_E, \tag{3}$$

$$v_E = \sqrt{v_W^2 + v_{Air}^2 + 2v_W v_{Air}\cos(\theta_W - \theta_{Air})}\ , \tag{4}$$

$$\theta_E = \theta_{Air} + \sin^{-1}(v_W \sin(\theta_W - \theta_{Air})/v_E)\ , \tag{5}$$

$$v_{Air} = f_{M2V}(M_{opti}, h_c)\ , \tag{6}$$

$$\theta_W = \varphi_A\ ,\ v_W = v_A\ , \tag{7}$$

M_{optic} and h_c are cruise Mach and cruise altitude respectively, $f_{M2V}(\cdot)$ is a function calculating air speed with M_{optic} and h_c as inputs, and T_{ts} equals to 10 minutes, i.e., a time-slice.

The coordinates (x_B,y_B) are then used as the beginning point of next sub-trajectory. Then, by an interpolation method presented in McDonald and Zhao (2000), the wind parameter (φ_B, v_B) can be calculated based on the coordinates (x_B,y_B) and the atmospheric conditions broadcasted by ATC agencies. Therefore, the coordinates of the end point of the new sub-trajectory can be calculated in the same way. The computation of sub-trajectories keeps going on until the destination airport is reached.

For the last sub-trajectory in an optional flight path, the end point is the destination airport, therefore, (x_B,y_B) are already available. However, the flight time for the last sub-trajectory is not necessarily a time-slice and needs to be calculated. Suppose the point B in Figure 3 is the destination airport, then the flight time can be computed by

$$t_{last} = S_{AB} / v_E\ , \tag{8}$$

where

$$S_{AB} = dis(P_A, P_B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} \tag{9}$$

$$\theta_E = \theta_{BA} = 90^\circ - a\tan 2((y_B - y_A),(x_B - x_A)) \tag{10}$$

$$\theta_{Air} = \theta_E - \sin^{-1}(-v_W \sin(\theta_E - \theta_W)/v_{Air}) \tag{11}$$

$$v_E = v_{Air}\cos(\theta_E - \theta_{Air}) + v_W \cos(\theta_E - \theta_W)\ , \tag{12}$$

and "$a\tan 2(\cdot)$" is a function calculating the four quadrant arctangent.

Suppose that, excluding the last sub-trajectory, there are N sub-trajectories in an optional flight path. Then the corresponding flight time cost is

$$J_1 = NT_{ts} + t_{last}\ . \tag{13}$$

ATC agencies sometimes require aircraft to arrive as soon as possible. For instance, when the ATC agency plans an arrival sequence at a busy airport, the first aircraft in the sequence is normally commanded to arrive as soon as possible, in order to leave more time for the following aircraft. In this case, the maximum cruise Mach number corresponding to the specified cruise height should be used. The computing process is the same as the above, except that M_{optic} is replaced by $M_{\max c}$.

3 RHC algorithm

Existing methods in the literature for online optimizing flight path under FF have one thing in common; that is, in each time-slice, they optimize the rest flight path from the end of current sub-trajectory to the destination airport. Consequently, they all suffer from two common problems. One problem is that, since the path optimization is a NP (Nondeterministic Polynomial Time) complete problem, for long-distance flight, it is unlikely that computing can be completed within a time-slice. The other problem is that, in a dynamic environment, it is very likely that the performance of conventional dynamic optimization could be degraded due to the involvement of uncertain information for the far future. Particularly, for those methods where optimization starts from the destination airport backward to the end of current sub-trajectory, for instance, see Andreatta et al (2000), their optimized paths for the near future depend on the optimized paths for the far future, which are calculated based on more unreliable information.

3.1 The idea of RHC

The proposed algorithm takes advantage of the concept of RHC to overcome the above problems in existing methods. At each step, i.e., time-slice, the proposed RHC algorithm optimizes the flight path for the next *N* time-slices into the near future. Therefore, no matter how long the flight distance is, the online computational time for each optimization is covered by an upper bound, which mainly depends on *N*, the length of the receding horizon. Also, a properly chosen receding horizon can work like a filter to remove the unreliable information for the far future. Figure 4 gives an intuitive demonstration of the idea of RHC and the potential advantages against those conventional dynamic optimization based methods.

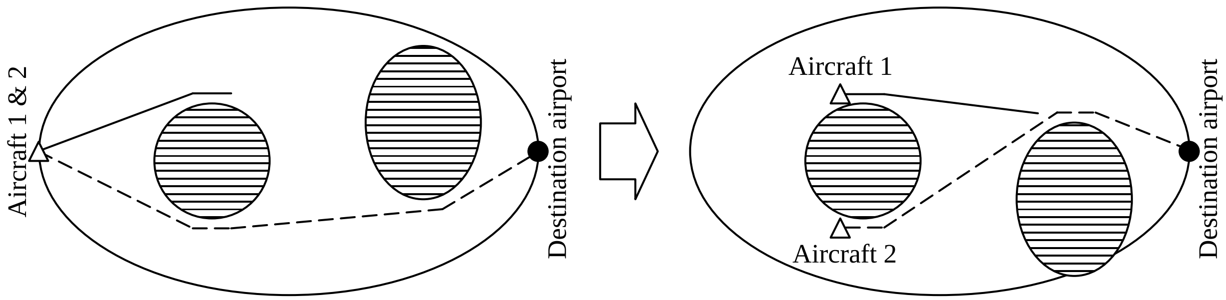

Figure 4. RHC (aircraft 1) vs conventional dynamic optimization algorithms (aircraft 2) in a dynamic FF environment

The online optimization problem in the proposed RHC algorithm is quite different from that in conventional dynamic optimization based methods, where J_1 given in (13) or similar ones are chosen as the performance index to be minimized in online optimization. For the RHC algorithm, in each time-slice, it is supposed to optimize flight path only for the receding horizon, which is N-time-slice-long or even shorter, depending on how far away the destination airport is. Therefore, one might think that minimizing flight time seems no sense to the RHC algorithm. The fact is that, in a FF airspace with unavailable regions, most potential paths of N-time-slice-long are of zigzag shape, and shortcut often exists between some of their sub-trajectories, as illustrated in Figure 5. It is evident that, after taking a shortcut, even though the original potential zigzag path is planed based on the receding horizon of N-time-slice-long, the final potential path is of an uncertain but shorter length. As the result, minimizing flight time based on a receding horizon of fixed length still makes sense. In fact, in order to find optimal flight paths, the conventional dynamic optimization based method also needs to take shortcut.

Suppose that, at the kth time-slice, after taking shortcut, an original potential path becomes $M(k)$-time-slice-long, where $0 \le M(k) \le N$ is a real number, and the fraction of $M(k)$ equals to the flight time through the last sub-trajectory divided by T_{ts}, i.e., one has

$$M(k) - floor(M(k)) = t_{last} / T_{ts} \tag{14}$$

where "*floor*" rounds $M(k)$ to the nearest integer towards negative infinity. For an original potential zigzag flight path, one has $M(k)=N$ except the case where the destination airport is reached.

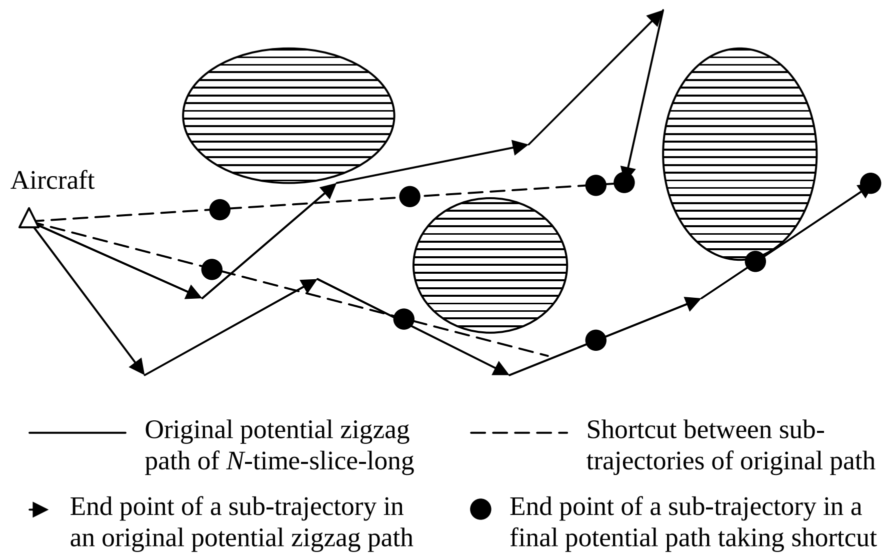

Figure 5. Zigzag flight paths and shortcut

The performance index adopted by the proposed RHC algorithm is given as

$$J_2(k) = M(k)T_{ts} + \mathbf{W}_{term}(k) \tag{15}$$

where $\mathbf{W}_{\mathrm{term}}(k)$ is a terminal weighting term function in terms of the last sub-trajectory and the destination airport. More detailed discussions about $\mathbf{W}_{\mathrm{term}}(k)$ will be given later. Then, the proposed RHC algorithm for optimizing flight path in a dynamic FF environment can be described as following.

Step 1: When aircraft takes off from the source airport, flying the departure program, let $k = 0$, and set $P(0)$ as the end point of the departure program.

Step 2: Receive updated environment data from ATC agencies, set $P(k)$ as the initial point to start flight path optimization, and then solve the following minimization problem

$$\min_{P(k+1|k),P(k+2|k),\cdots,P(k+N|k)} J_2(k) \tag{16}$$

subject to available headings in Ω and unavailable regions, where $P(k+i|k)$, $i = 1,\ldots,N$, is the end point of the ith sub-trajectory in a original potential zigzag flight path at the kth step. Denote the optimal solution as $[\hat{P}(k+1|k), \hat{P}(k+2|k),\cdots,\hat{P}(k+N|k)]$, and the associated shortcut-taken flight path as $[\hat{P}_f(k+1|k),\cdots,\hat{P}_f(k+ceil(M(k))|k)]$, where "*ceil*" rounds $M(k)$ to the nearest integer towards infinity.

Step 3: When aircraft arrives at $P(k)$, set

$$P(k+1)=\hat{P}_f(k+1|k), \tag{17}$$

and then fly along the sub-trajectory determined by $[P(k), P(k+1)]$.

Step 4: If $P(k+1)$ is not the destination airport, let $k=k+1$, and go to Step 2. Otherwise, the algorithm finishes.

3.2 GA-based optimizer

Many existing methods can be used as the online optimizer to solve the minimization problem (16). Since the model in Section 2 provides no predefined flight path network, a potential online optimizer should firstly be effective in searching feasible flight paths in the non-conflict-airspace. As is well known, GA is a large-scale parallel stochastic searching and optimizing algorithm, and it suits well the nature of the problem (16). In this paper, the improved GA presented in Hu et al (2004) is adopted as the online optimizer. A chromosome in the GA optimizer is structured based on the end points of sub-trajectories in an original potential zigzag flight path or a shortcut-taken flight path. Since $M(k)$ is an uncertain bounded real number, different chromosomes could have different length. Therefore, a chromosome is structured like this: the first gene records the value of $M(k)$, "$ceil(M(k))$" is number of end points of sub-trajectories in the corresponding flight path, and the following genes record in order the coordinates of these points, as illustrated in Figure 6.

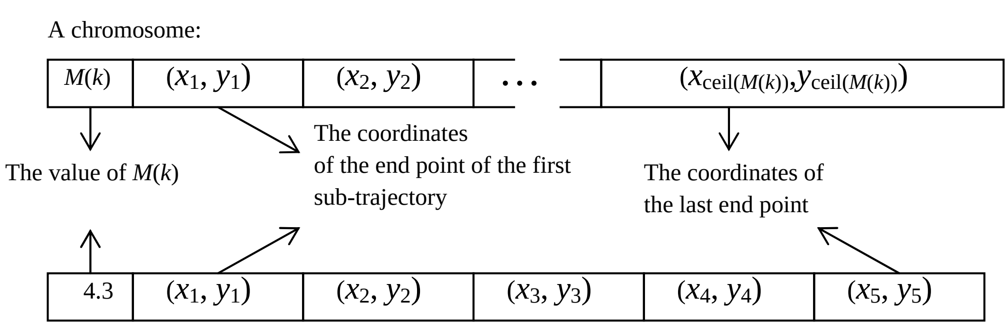

Figure 6. Structure of Chromosome

With the information recorded in a chromosome, the value of $J_2(k)$ for the corresponding potential flight path can be calculated according to (2) to (15). Suppose at the kth time-slice, there are n chromosomes in a generation, the value of $J_2(k)$ for the ith chromosome is $q_i(k)$, and $q_{max}(k)$ and $q_{min}(k)$ stand for the maximum and minimum values of $J_2(k)$ in the generation. Then, the fitness of the ith chromosome is defined as

$$F_i(k) = \begin{cases} q_{\max}(k) - q_i(k) + (q_{\max}(k) - q_{\min}(k))/n, & q_{\max}(k) \neq q_{\min}(k) \\ q_{\max}(k) - q_i(k) + (q_{\max}(k) - q_{\min}(k))/n + q_{\max}(k), & q_{\max}(k) = q_{\min}(k) \end{cases} \quad (18)$$

The GA presented in Hu et al (2004) used many effective techniques, such as young generation and its growing process, self-adapted crossover and mutation probabilities, and heuristic rules, to improve the performance of the algorithm. It proved to be effective to find optimal or sub-optimal solutions in Hu et al (2004). As will be illustrated in the simulation section, the proposed RHC algorithm integrated with this GA optimizer works very well.

3.3 The length of receding horizon and terminal weighting

The choice of N, the length of the receding horizon, is important. The online computational time for each optimization is covered by an upper bound, which mainly depends on N and can be estimated through simulations. Therefore, as long as the time-slice is larger than the upper bound, no matter how long the global flight distance is, the real time property of the proposed algorithm is always guaranteed. Also, a properly chosen receding horizon can work like a filter to remove unreliable information for the far future. If N is too large, the RHC algorithm will face the same problems regarding requirements for real-time computation and dynamic environment, as existing methods do. Otherwise, if N is too small, the RHC algorithm will become shortsighted, and the performance will significantly degrade. A properly chosen N should be such that a good trade-off could be achieved between online computational burden and robust performance of the algorithm.

However, the nature of receding horizon makes the proposed algorithm inevitably shortsighted in some sense, especially when compared with conventional dynamic optimization based methods in a static FF environment. The introduction of terminal weighting term $\mathbf{W}_{term}(k)$ in $J_2(k)$ can further reduce the shortsightedness of the algorithm, although $\mathbf{W}_{term}(k)$ has other much more important work to do.

In the earlier implementation of RHC in control engineering, performance indices without terminal weighting terms were widely used, but it was observed that the plants under RHC might become unstable. To address this issue, the technique of terminal weighting was introduced (Clarke, 1994). Now, terminal weighting has been widely accepted in the area of control engineering as a key technique to guarantee the stability of RHC. In the case of applying RHC to online flight path optimization under FF, if no terminal weighting term or an improper terminal weighting term is used, very bad performance may be obtained.

Suppose that $\mathbf{W}_{term}(k)$ is removed from $J_2(k)$, i.e.,

$$\mathbf{W}_{term}(k) = 0\,. \tag{19}$$

Then, if the destination airport is beyond reach at the kth time-slice, $J_2(k)$ will have no information of the destination airport. In this case, the result of online optimization will result in a random flight path, which could probably never lead to the destination airport, as illustrated by the dashed line in Figure 7. This is an unstable situation, due to no terminal weighting in $J_2(k)$.

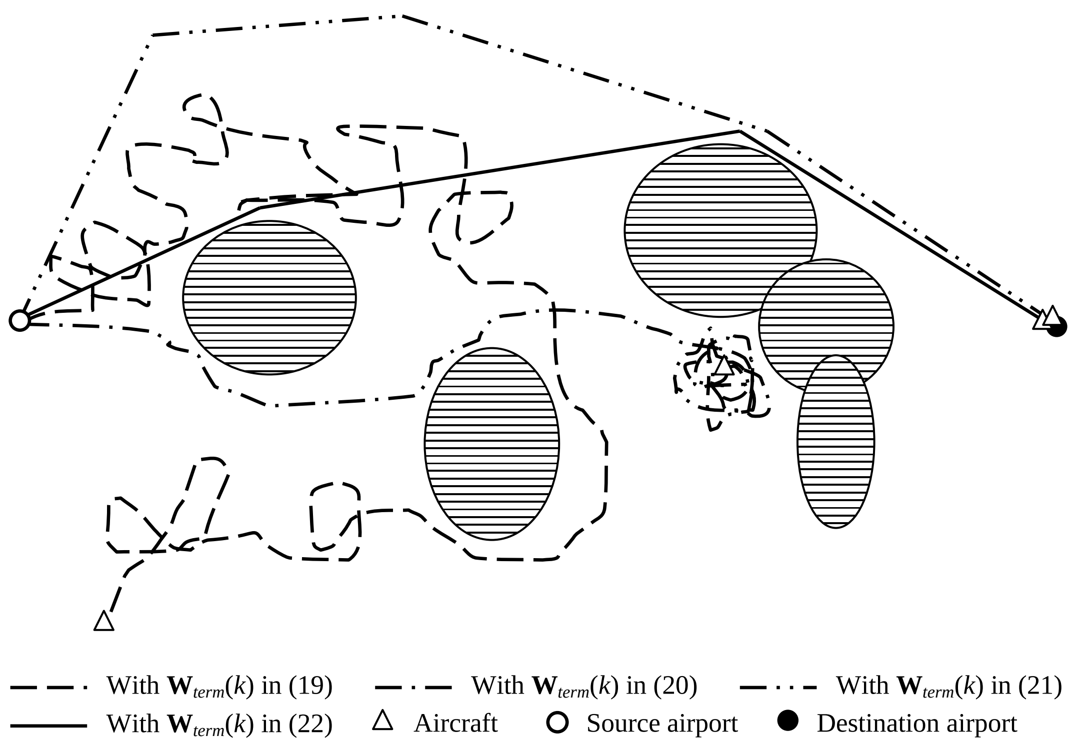

Figure 7. Flight trajectories under different terminal weighting terms

A simple way to add the information of the destination airport into $J_2(k)$ is using the following terminal weighting term

$$\mathbf{W}_{term}(k) = dis(P_{last}(k), P_{D.A.}) / v_E \,, \tag{20}$$

where $P_{last}(k)$ is the end point of the last sub-trajectory in a potential flight path, $P_{D.A.}$ is the destination airport, and the ground speed v_E and the function "*dis*" are given in Eq. (9) and (12), respectively. The $\mathbf{W}_{term}(k)$ in (20) can effectively avoid such random flight trajectories resulting from (19), and could lead aircraft to the destination airport in many cases. However, without using the information of unavailable regions, a new problem arises sometimes that aircraft is trapped in a small region and the algorithm can hardly get it out, as shown by the dot–and-dash line in Figure 7.

To avoid such trapping regions and the corresponding undesired phenomenon, some necessary information of unavailable regions should be included in the terminal weighting term. Basically, those unavailable regions standing between $P_{D.A.}$, the destination airport, and $P_{last}(k)$, the end point of the last sub-trajectory in a potential flight path, are the main concern. For the sake of convenience, hereafter, we call these unavailable regions as IW (in-the-way) regions, and other unavailable regions as OW (out-of-way) regions. If there are no IW regions, then $\mathbf{W}_{term}(k)$ is defined by (20). Otherwise, the closest IW region (maybe including several ellipsoidal regions which overlap each other) to $P_{last}(k)$ can be easily used to improve the terminal weighting term as following

$$\mathbf{W}_{term}(k) = (\alpha \,\mathbf{min}(\theta_1, \theta_2) / \mathbf{max}(\theta_1, \theta_2) + 1) dis(P_{last}(k), P_{D.A.}) / v_E \,, \tag{21}$$

where θ_1 and θ_2 are angles illustrated in Figure 8, and $\alpha > 0$ is a coefficient for tuning. It is evident that using $\mathbf{W}_{term}(k)$ in (21) can prevent aircraft from getting trapped in a region, because in a potential trapping region, $\mathbf{min}(\theta_1, \theta_2) / \mathbf{max}(\theta_1, \theta_2)$ gets close to 1, the maximum, which will lead to heavy penalty.

However, $\mathbf{W}_{term}(k)$ in (21) is still not very efficient regarding flight time. As shown by the double-dot-and-dash line in Figure 7, one can see, to avoid trapping regions, the aircraft could turn away too much from the direct heading θ_{dire}. To make the proposed RHC algorithm more efficient to find optimal flight paths rather than feasible paths, more modifications are needed to the terminal weighting. Denote $P_{prev}(k)$ is the point $P_{last}(k)$

just follows in a potential flight path. If the number of IW regions is not zero, then a more efficient terminal weighting is

$$\mathbf{W}_{term}(k) = (\beta\,|\,\theta_3\,|\,/\,\theta_4 + 1)dis(P_{last}(k), P_{D.A.})\,/\,v_E\,, \tag{22}$$

where θ_3 and θ_4 are angles illustrated in Figure 9, and $\beta > 0$ is a coefficient for tuning. $\theta_3 > 0$ means the heading of the last sub-trajectory in a potential flight path is over-turning. Oppositely, $\theta_3 < 0$ means under-turning. In either case, it will be penalized by $\mathbf{W}_{term}(k)$ defined by (22). Regardless of the influence of atmospheric conditions, which is in fact already covered by the first part of $J_2(k)$, i.e., $M(k)T_{ts}$, $\mathbf{W}_{term}(k)$ defined by (22) should be a very efficient choice, as illustrated by the solid line in Figure 7.

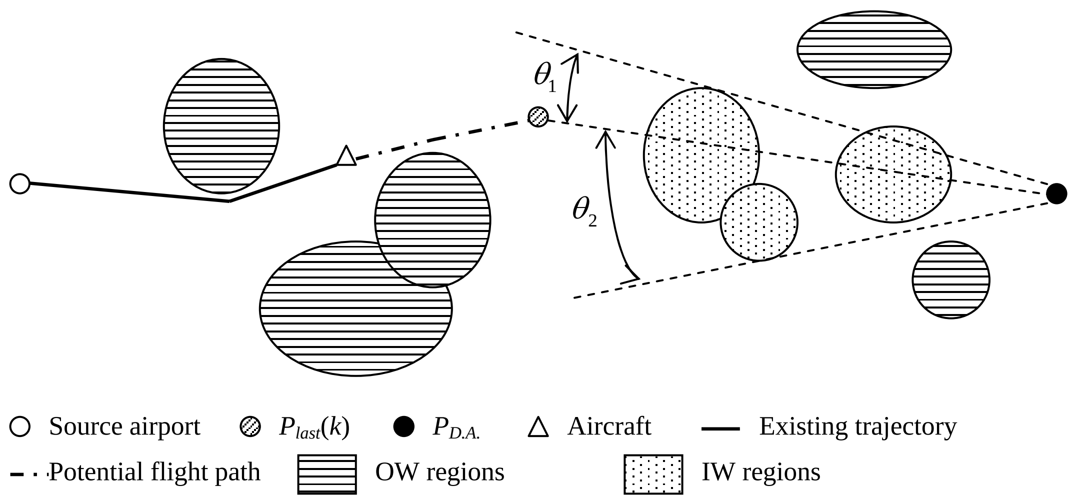

Figure 8. How to define terminal weighting in (21)

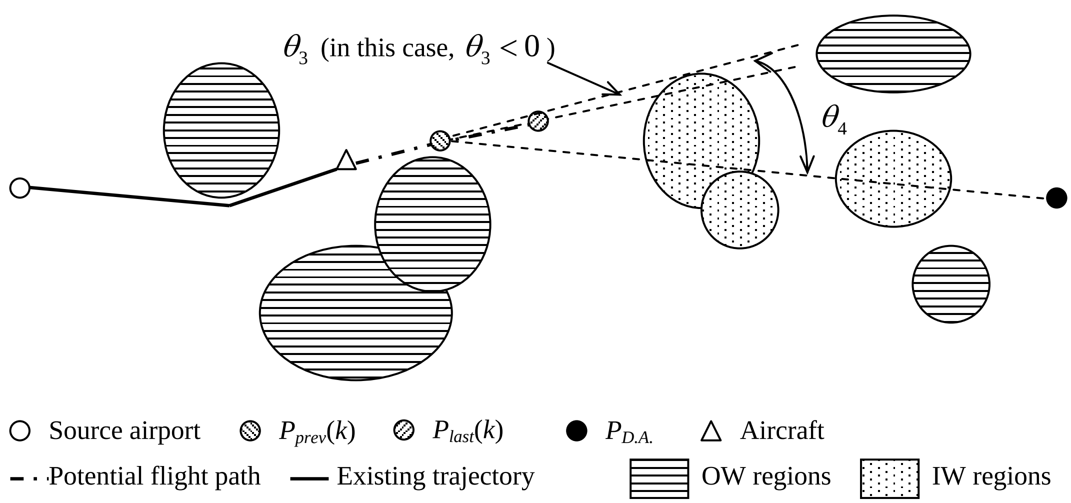

Figure 9. How to define terminal weighting in (22)

In a dynamic environment, unavailable regions could move, change in size, or even disappear. The dynamics of unavailable regions can also be simply included in $\mathbf{W}_{term}(k)$. A simple way of taking advantage of the dynamics of unavailable regions to some extent is considering the direction in which the closest IW region to $P_{last}(k)$ is moving:

$$\mathbf{W}_{term}(k) = ((1+\rho)\beta\,|\theta_3|/\theta_4 + 1)dis(P_{last}(k), P_{D.A.})/v_E\,, \tag{23}$$

$$\rho = \gamma sign(\theta_{IW} - \theta_6)sign(\theta_5 - \theta_6)\,, \tag{24}$$

where θ_3 and θ_4 are defined as in Figure 9, θ_5, θ_6 and θ_{IW} are clockwise-turning angles with respect to the north, as illustrated in Figure 10, θ_{IW} is the direction in which the closest IW region to $P_{last}(k)$ is moving, $\gamma > 0$ is a tuning parameter, and "*sign*" is a function which takes the sign of input.

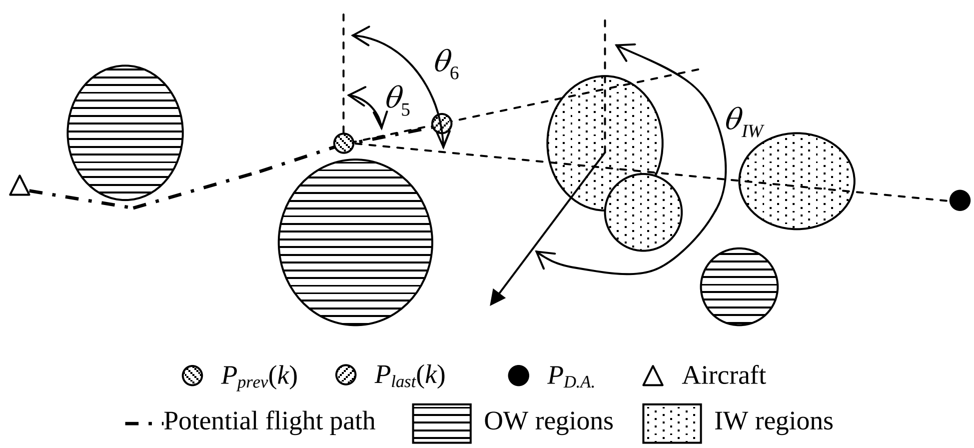

Figure 10. How to define terminal weighting in (23)

So far, only the closest IW region to $P_{last}(k)$ is used by the terminal weighting term. Further study can be focus on how to use other unavailable regions and how to make most of them. Before this can be possible, investigations on the stochastic distribution and dynamics of unavailable regions should be carried out, which are beyond the scope of this paper.

4 Simulation results

In order to evaluate the proposed RHC algorithm, the simulation system reported in Hu et al (2004) is adopted to set up different FF environments, and the conventional dynamic optimization based algorithm in Hu et al (2004) is also used for comparative purposes. For the sake of identification, hereafter, the proposed RHC algorithm is denoted as RHC, and the algorithm in Hu et al (2004) as CDO. It is fair to compare RHC with CDO, because they use the same GA as online optimizer. More details of the GA optimizer can be found in Hu et al (2004). In the simulation, unless it is specifically pointed out, the length of receding horizon is N=6, or 1-hour-long, and the terminal weighting term $\mathbf{W}_{term}(k)$ defined in (23) is adopted for RHC.

Six simulation cases are defined in Table 1 with different degree of complexity of the FF environment, where DD stands for the Direct Distance from the source airport to the destination airport, and UR for Unavailable Region. In Case 1 to 3, the UR's are static, while in Case 4 to 6, UR's may vary with time, in other words, they can move, change in size, disappear, or some new UR's could turn up randomly. The comparative simulation focuses on online computational times (OCT's) and performances, i.e., actual flight times (AFT's) from the source airport to the destination airport, of the RHC and CDO. Figure 11 gives an example of Case 5 to demonstrate the dynamic process of optimizing the free-flight path under RHC. In Figure 11, solid circles indicate unavailable airspace, dashed circles stand for source/destination airports, the triangle represents aircraft for which the free-flight path is optimizing, dashed line is the current optimal path, dot-and-dash line is the optimal path calculated in the previous time-slice, and solid line is the flight trajectory of the aircraft in the past. To save space, we only pick up and show the results associated with certain eight time-slices of the whole flight process. Numerical results are given in Table 2 to 6, where for each static case, 10 simulation runs are conducted under either RHC or CDO, while 200 simulation runs for each dynamic case.

Table 1 Six simluation cases

	Static environment			Dynamic environment		
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
DD (nm)	500	1000	2000	500	1000	2000
No. of UR's	1	6	14	1	6	14

Although the RHC is mainly proposed for dynamic cases, it still needs to work well in static cases. Table 2 gives the simulation results in Case 1 to 3 under different algorithms. From Table 2, one can see, the CDO achieves the best performances, i.e., the least AFT's, in all 3 cases. This is understandable, because, theoretically, in static cases, conventional dynamic optimizing strategies like CDO should be the best by nature in terms of performance. Table 2 also shows that the performances of the RHC are very close to those of the CDO, which means the RHC works very well in static cases. With respect to OCT's, the RHC is clearly much more efficient than the CDO. Since one time-slice is 10-minute-long, one can see that there is no problem for the RHC to run in real-time, while the CDO does struggle to finish online computation in some cases.

Table 2 Simuluation results in static cases

	CDO			RHC		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
Ave. OCT (s)	1.2687	8.3675	77.5364	2.5675	4.8498	7.3047
Ave. AFT (s)	3965.6	7407.3	14868	3966.2	7421.5	14905
Max. OCT (s)	5.3970	37.479	364.924	5.7970	7.408	15.5510
Max. AFT (s)	3966.9	7435.7	14913	3968.7	7480.4	15052

Dynamic cases are the main concern, and some corresponding simulation results are given in Table 3. As for performances, in relatively simple cases like Case 4 and 5, the CDO and RHC have similar OCT's, while in complicated cases like Case 6, the performance of RHC is better than that of the CDO. The reason for this has already been fully discussed in Section 2 and 3. Again, the RHC provides reliable and promising real-time property against the CDO.

Table 3 Simuluation results in dynamic cases

	CDO			RHC		
	Case 4	Case 5	Case 6	Case 4	Case 5	Case 6
Ave. OCT (s)	0.9623	9.4485	68.9219	2.4930	3.8419	7.8754
Ave. AFT (s)	4222.0	7475.6	16192	4221.6	7454.3	15932
Max. OCT (s)	5.317	38.968	347.915	5.8990	6.3190	17.6940
Max. AFT (s)	4223.9	8492.5	16638	4223.1	7995.8	16118

Case 6 is the most complicated case in all 6 cases, but the DD is just 2000 nm's, which is still very short when compared with inter-continental flights. This implies that the CDO is unlikely to handle inter-continental flights regarding real-time properties. Then, how about the RHC, whose OCT's also increase in Case 6? As defined in Table 1, from Case 4 (or 1) to 6 (or 3), both DD and the number of UR's increase. Then, which one, DD or the number of UR's, influences the OCT of the RHC more significantly? Table 4 answers this question, where DD changes between [500,1000,2000], the number of UR's changes between [1,6,14], and all cases are dynamic. One can see from Table 4 that, the OCT of the RHC mainly depends on UR's (because the number of dynamics of UR's significantly influence the computational burden of the GA optimizer to find potential flight paths and to calculate terminal weighting), and has little to do with DD (because, for the RHC, it is not DD but *N* which determines the possible maximum flight time of a potential flight path). In the real world, most UR's are other aircraft. Those aircraft which are too far away, because of their fast dynamics, are of little use for the current online optimization. Therefore, the number of useful UR's will not increase significantly with DD, which makes the RHC ready for inter-continental flights in a real-time sense.

Table 4 Influence of DD and UR's on the OCT of the RHC

OCT (s)	DD=500 (nm's)		DD=1000 (nm's)		DD=2000 (nm's)	
	Ave.	Max.	Ave.	Max.	Ave.	Max.
1 UR	2.4930	5.8990	3.0098	6.4690	3.1031	8.8820
6 UR's	3.9371	6.8190	3.8419	6.3190	3.7580	7.0300
14 UR's	5.5200	10.765	5.1256	10.554	7.8754	17.694

Table 5 makes it more clear that, *N*, the length of the receding horizon, should be properly chosen. If *N* is too small, the performance is very poor, as the case of *N*=1 and 3 in Table 5. While, if *N* is too large, OCT's increase, but the performance is not necessarily improved further. Instead, the performance could degrade in dynamic cases, as shown by the case of *N*=9 in Table 5.

Table 5 Influence of *N* on the RHC

		Static environment			Dynamic environment		
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
N=1	OCT(s)	0.8340	0.9365	1.3362	0.7337	0.8465	1.2590
	AFT(s)	4006.5	8054.9	17891	4225.1	7976.8	16922
N=3	OCT(s)	1.3003	1.9507	2.5392	1.2907	1.4612	2.2652
	AFT(s)	3965.0	7811.0	15674	4226.5	7482.6	16207
N=6	OCT(s)	2.5675	4.8498	7.3047	2.4930	3.8419	7.8754
	AFT(s)	3966.2	7421.5	14905	4221.6	7454.3	15932
N=9	OCT(s)	4.6264	10.6017	18.2554	4.0966	8.5754	17.7370
	AFT(s)	3965.9	7407.6	14894	4221.9	7462.4	16074

Table 6 shows the influence of terminal weighting term on the RHC. Since the $\mathbf{W}_{term}(k)$ defined in (19) makes the algorithm unstable, no associated results are given in Table 6. Basically, one can see that the performance of the RHC is improved step by step after using $\mathbf{W}_{term}(k)$ defined in (20), (21), (22) and (23), with OCT maintain at the same level. The reason has already been fully discussed in Section 3.3

Table 6 Influence of terminal weighting on the RHC

$\mathbf{W}_{term}(k)$ in		Static environment			Dynamic environment		
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
(20)	OCT(s)	2.8102	4.9121	7.1125	2.8994	3.8453	9.4565
	AFT(s)	4114.9	7435.3	15183	4450.9	7496.1	16114
(21)	OCT(s)	2.7294	5.0376	7.3552	2.5127	3.7952	9.1149
	AFT(s)	3969.0	7421.8	15042	4219.0	7465.3	16089
(22)	OCT(s)	2.7897	5.1164	7.1016	2.5353	3.7683	8.5240
	AFT(s)	3966.3	7414.7	14896	4227.3	7405.4	16028
(23)	OCT(s)	2.5675	4.8498	7.3047	2.4930	3.8419	7.8754
	AFT(s)	3966.2	7421.5	14905	4221.6	7454.3	15932

5 Conclusions

As is well known. “Free Flight” is one of the most promising strategies for future air traffic control systems. With the framework of “Free Flight”, each individual aircraft has the first responsibility to plan its flight in terms of safety, efficiency and flexibility. One of the key techniques in this strategy is the ability of onboard flight management computer systems to optimize flight paths in real time. Two questions arise in the online flight path optimization problem: how to achieve robust performance in a dynamic environment, and how to reduce online computational burden to satisfy the time limit in practical applications.

This paper introduces the concept of Receding Horizon Control to attack the problem. After the mathematical model for the online flight path optimization problem in a “Free Flight” environment is formulated, the RHC algorithm for free flight is presented in details. The major techniques of the algorithm, suck as how to choose the length of the receding horizon and how to use the terminal weighting, are fully investigated and discussed. Simulation results show that, regarding performance, the proposed RHC algorithm is as good as the existing algorithm in the absence of uncertainties, and achieves better solutions in a dynamic environment. The main advantage of the RHC algorithm is its high efficiency regarding the online computational time, which makes the proposed algorithm ready for practical applications.

6. Acknowledgements

This work is supported by an ORS Award from the Overseas Research Students Awards Scheme, Universities UK.

Reference

Benoit, A, 1994. Handing of Air Traffic. AGARD-AG-321.

Pelegrin, M, 1994. Towards Global Optimization for Air Traffic Management. AGARD-AG-321,

Wickens, C.D., Mavor, A.S., Parasuraman, R., McGee, J.P., 1998. Airspace System Integration-The Concept of Free Flight. National Academy Press.

Kahne, S., 2000. Research issues in the transition to free flight. Annual reviews in Control, vol.24, pp. 21-29.

McLean, D., 2003. Operational requirements for fully automatic flight. Measurement and Control, vol.36, no.6, pp. 181-183+187.

Hoekstra, J.M., Van Gent, R.N.H.W., Ruigrok, R.C.J., 2002. Designing for safety: free flight air traffic management concept. Reliability Engineering and System Safety, vol.75, no.2, pp. 215-232.

Braune, R.J., Jahnsad, D.W., Bittner, A.C.J., 1996. Human factors systems engineering as a requirement for the safe and efficient transition to free flight. Proceedings of the 40th Annual Meeting of Human Factors and Ergonomics Society, Part 1 (of 2), Philadelphia, PA, USA.

Hu, X.B., Wu, S.F., Jiang, J., 2004. On-Line Free-Flight Path Optimization Based on Improved Genetic Algorithms. *Engineering Applications of Artificial Intelligence*, Vol.17, No.8, pp. 897-907.

Geser, A., Munoz, C., 2002. A geometric approach to strategic conflict detection and resolution. AIAA/IEEE Digital Avionics for Commercial and Military Systems-Proceedings, Irvine, CA, USA, vol.1 pp. 6B11-6B111.

Pallottino, L., Bicchi, A., Pancanti, S., 2002. Safety of a decentralized scheme for free-flight ATMS using mixed integer linear programming. Proceedings of the American Control Conference, Anchorage, AK, USA, vol.1, pp. 742-747.

Granger, G., Durand, N., Alliot, J.M., 2001. Token allocation strategy for free-flight conflict solving. Innovative Applications of Artificial Intelligence-Conference Proceedings, Seattle, WA, USA, pp. 59-64.

Oh, J.H., 1999. SDP relaxation approach to air traffic control under free flight. IEEE Conference on Control Applications-Proceedings, Kohala Coast, HI, USA, vol.1, pp. 158-163.

Shewchum, J.M., Oh, J.H., Feron, E., 1997. Linear matrix inequalities for free flight conflict problems. Proceedings of the IEEE Conference on Decision and Control, San Diego, CA, USA, vol.3, pp. 2417-2422.

Durand, N, Alliot, J.-M., Chansou, O., 1995. Optimal Resolution of En Route Conflicts. Air Traffic Control Quarterly, 3(3).

Warren, A.W., Schwab, R.W., 1997. A Methodology and Initial Results Specifying Requirements for Free Flight Transitions. Air Traffic Control Quarterly, 5(3).

Plaettner, J.K., Zhao, Y., 2000. A Modularized Approach For Comprehensive Air Traffic System. AIAA-2000-4478.

McDonald, J.A., Zhao, Y., 2000. Time Benefits of Free-Flight for a Commercial Aircraft. The Proceedings of the AIAA Guidance Navigation & Control Conference 2000-Denver CO.

Valuev, A.M., Velichenko, V.V., 2002. On th eproblem of planning a civil aircraft flight along a free route. Izvestiya Akademii Nauk. Teoriyai Sistemy Upravleniya, no.6, pp. 149-157.

Andreatta, G., Brunetta, L., Guastalla, G., 2000. From ground holding to free flight: an exact approach. Transportation Science, vol.34, no.4, pp. 394-401.

Clarke, D.W., 1994. Advances in Model-based Predictive Control. Oxford University Press.

De Schutter, B., Van Den Boom, T., 2001. Model predictive control for max-plus-linear discrete event systems. Automatica, vol.37, no.7, pp. 1049-1056.

Chand, S., Hsu, V.N., Sethi, S., 2002. Forecast, solution, and rolling horizons in operations management problems: a classified bibliography. Manufacturing & Service Operations Management, vol.4, no.1, pp.25-43.

Chen, W.H., Van Den Boom, T., De Schutter, B., 2002. Discussion on: 'Properties of MPC for max-plus-linear systems' by T. van den Boom and B. De Schutter. European Journal of Control, vol.8, pp. 463-464.

Erzberger, H., 1995. Design principles and algorithms for automated air traffic management. AGARD Lecture Series No. 200*: Knowledge-Based Functions in Aerospace Systems*, Madrid, Paris, and San Francisco.

Neuman, F., Erzberger, H., 1991. Analysis of delay reducing and fuel saving sequencing and spacing algorithms for arrival traffic. NASA Technical Memorandum 103880, Ames Research Center.

Schick, F., 1998. Methods and measurements for the evaluation of ATM tools in real-time simulations and field tests. 2nd USA/EUROPE Air Traffic Management R&D Seminar, Orlando.

Hu, X.B., Chen, W.H., 2005. Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling. IEEE Transaction on Intelligent Transportation System, in press.

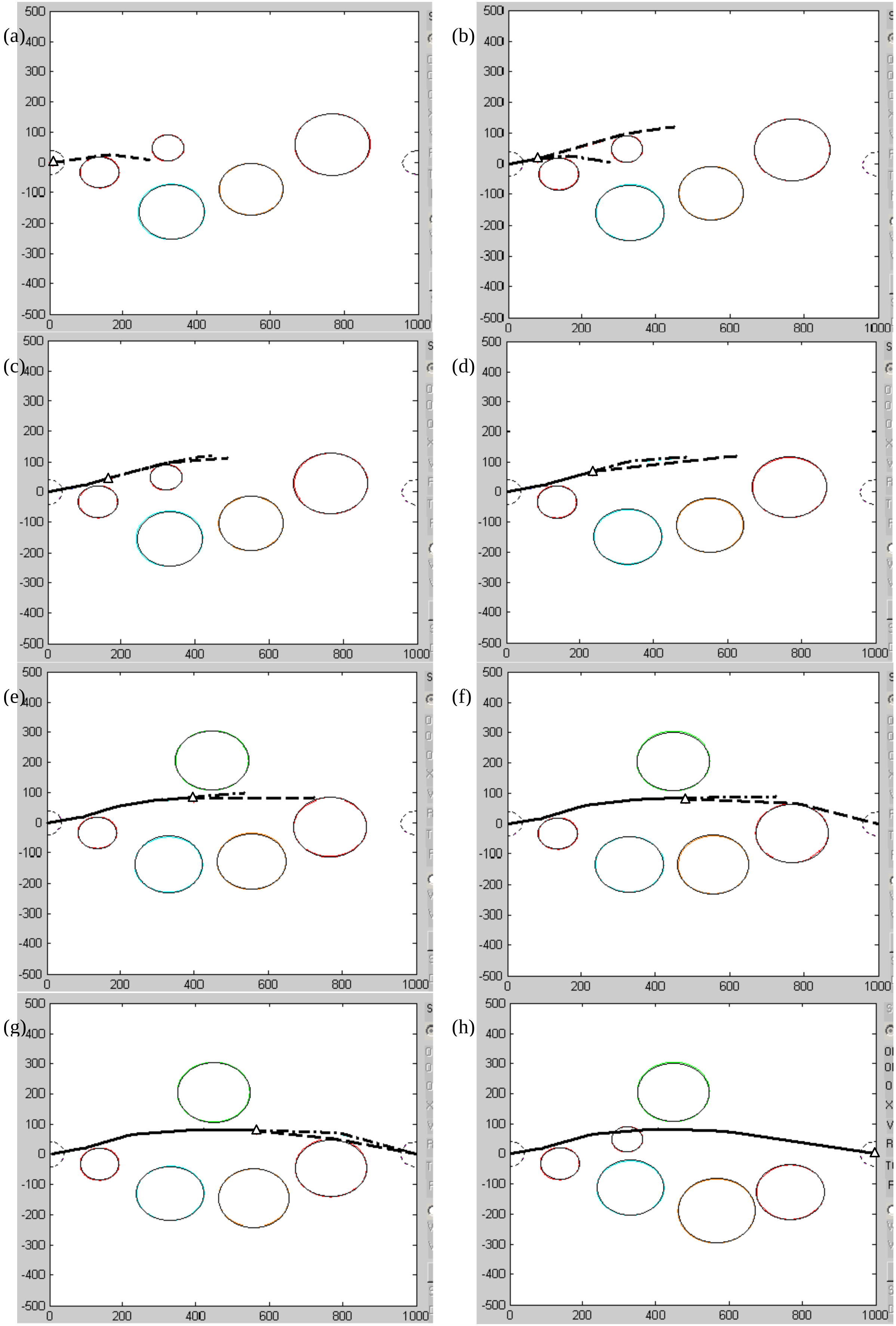

Figure 11. An example of optimising flight path under the RHC