Receding Horizon Control for Free-Flight Path Optimization

Xiao-Bing Hu[1] and Wen-Hua Chen[2]

[1] Department of Informatics, University of Sussex, UK
xiaobing.Hu@Sussex.ac.uk

[2] Department of Aero. and Auto. Engineering, Loughborough University, UK
W.Chen@lboro.ac.uk

Summary. This paper presents a Receding Horizon Control (RHC) algorithm to the problem of on-line flight path optimization for aircraft in a dynamic 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 in the real-time optimization of this complicated air traffic control problem. Compared with existing algorithms, the new algorithm proves more efficient and promising for practical applications.

1 Introduction

"Free-Flight"(FF) is one of the most promising strategies for future air traffic control (ATC) systems [1, 2]. Within the FF framework, each individual aircraft has the first responsibility to plan its flight in terms of safety, efficiency and flexibility. One of the key enabling techniques is real-time path planning using onboard flight management systems. Reference [3] proposes an effective Genetic Algorithm (GA) for searching optimal flight paths in an FF environment, where no pre-defined flight routes network exists. However, two questions arise for the GA in [3]: how to cope with unreliable information in a dynamic environment, and how to improve real-time properties.

This paper introduces the concept of Receding Horizon Control (RHC) to the GA in [3] and then develops a more efficient algorithm for online optimizing flight paths in a dynamical FF environment. As an N-step-ahead online optimization strategy, firstly, RHC provides a promising way to deal with unreliable information for far future, and therefore increase the robustness/adoptation of the algorithms against environmental uncertainties/changes; secondly, the introduction of RHC can significantly reduce the heavy computational burden of the GA in [3] to an acceptable level. These achievements mainly rely on carefully choosing horizon length and properly designing terminal penalty in the newly proposed algorithm.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 565–572, 2007.
springerlink.com

2 Online Flight Path Optimization in FF Environment

2.1 Optional Free Flight Paths

In contrast to conventional pre-defined flight routes networks, there are numerous optional free flight paths in an ideal FF environment, as illustrated in Fig.1. Following [4], this paper uses the concept of "time-slice" and a set of discrete optional headings to transform the non-conflict-airspace into a dynamic flight routes network, and the optimization problem can be reasonably simplified.

Time-slice and discrete optional headings set are two system parameters which determine the complexity of the flight routes network. As discussed in [3], longer time-slice and less optional headings lead to a less flexible network; in the opposite extreme, the network becomes unnecessarily complicated. Referring to some papers on air conflict detection and resolution [4], where 5-min-long time interval and 10° discrete angular change for optimizing only local manoeuvres is adopted, this paper, to optimize global flight paths, uses a 10-min time-slice and a discrete set

$$\Omega = [0°, 10°, 20°, \cdots, 350°, \theta_{dire}] \tag{1}$$

where θ_{dire} is the direct-heading, which is defined as the direction of the destination airport with reference to the waypoint where the aircraft arrives at the end of the current time-slice. The ground ATC systems are supposed to periodically broadcast environmental information, particularly data of unavailable-regions, to each individual aircraft. Each individual aircraft uses the latest information to optimize the remained 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.

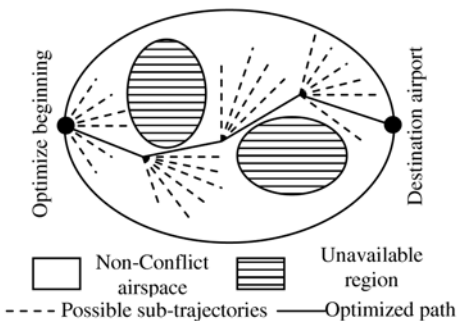

Fig. 1. Optimized path in an FF environment

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 be easily transformed into

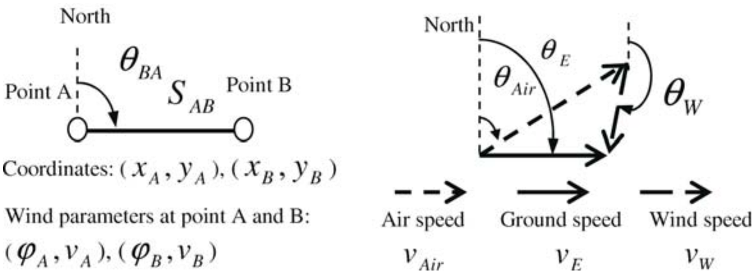

Fig. 2. Variables and parameters of a sub-trajectory and related speeds

other useful indexes for flight path optimization, such as fuel cost [5]. According to the discussion in Section 2.1, an optional flight path is determined by a number of waypoints. When the heading and the beginning waypoint of a sub-trajectory are given, since the flight time along a sub-trajectory is a time-slice (i.e., 10 minutes), the coordinates of the end waypoint of this sub-trajectory can be calculated by referring to Fig.2

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

where S_{AB} is the distance between two waypoints and

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

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

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

$$v_{Air} = f_{M2v}(M_{opti}, h_C) \qquad \theta_W = \varphi_A \qquad v_W = v_A \tag{6}$$

M_{opti} and h_c are cruise Mach and cruise altitude respectively, $f_{M2V}(\cdot)$ is a function calculating air speed with M_{opti} and h_c as inputs, and T_{ts} is 10 minutes. Since a sub-trajectory is very short as the result of the 10-min-long time-slice, it is reasonable to assume that the average wind parameters along the sub-trajectory are the same as those at the beginning waypoint, as described in Eq. 6. (x_B, y_B) are then used as the beginning waypoint of new sub-trajectory, and the wind parameter (φ_B, v_B) can then be calculated by an interpolation method proposed in [6] based on (x_B, y_B) and atmospheric conditions broadcasted by ATC agencies. The coordinates of the end waypoint 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 waypoint is the destination airport, and the actual flight time along the last sub-trajectory needs to be calculated using a similar method as Eq.(2-6). Suppose the flight time along the last sub-trajectory is t_{last}, and, excluding the last sub-trajectory, there are sub-trajectories in an optional flight path. Then the corresponding flight time cost is

$$J_1 = \bar{N}T_{ts} + t_{last} \tag{7}$$

3 RHC Algorithms

Similar to most other existing methods (e.g., see [7]), to online optimize FF paths, the GA in [3] optimizes, in each time-slice, the rest flight path from the end of current sub-trajectory to the destination airport. As a consequence, it suffers heavy computational burden, although it was proved to be effective in searching optimal paths in an FF environment. Also, the robustness of the algorithm in [3] against unreliable information in a dynamic FF environment has not been addressed.

3.1 The Idea of RHC

The proposed algorithm takes advantage of the concept of RHC to overcome the above problems in [3]. RHC is a widely accepted scheme in the area of control engineering, and has many advantages against other control strategies. Recently, attention has been paid to applications of RHC in those areas such as management and operations research [8]. Simply speaking, RHC is an N-step-ahead online optimization strategy. 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 horizon length. Also, a properly chosen receding horizon can work like a filter to remove unreliable information for the far future.

The online optimization problem in the proposed RHC algorithm is quite different from that in conventional dynamic optimization based methods, such as the GA in [3], where J_1 given in (7) is chosen as the performance index to be minimized in online optimization. The performance index adopted by the proposed RHC algorithm is given as

$$J_2(k) = N(k)T_{ts} + W_{term}(k) \tag{8}$$

where $W_{term}(k)$ is a terminal penalty to assess the flight time from the last way-point to the destination airport. The discussion about $W_{term}(k)$ will be given later and more detailed discussion can be found in [9]. The proposed RHC algorithm for optimizing flight paths in a dynamic FF environment can be described as following:

S1: When an aircraft takes off from the source airport, fly the departure program, let $k = 0$, and set $P(0)$ as the allocated departure fix of the departure program.

S2: 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

$$\min_{P(k+1|k),P(k+2|k),\ldots,P(k+N|k)} J_2(k) \tag{9}$$

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

S3: When the aircraft arrives at $P(k)$, set $P(k+1) = \hat{P}_f(k+1|k)$ and then fly along the sub-trajectory determined by $[P(k), P(k+1)]$.

S4: If $P(k+1)$ is not the destination airport, let $k = k+1$, and go to Step 2; otherwise, the algorithm stops.

3.2 The Length of Receding Horizon and Terminal Penalty

The choice of N, the horizon length, is important to design the proposed algorithm. The online computational time for each optimization is covered by an upper bound, which mainly depends on N and can be estimated through simulations. As long as the time-slice is larger than the upper bound, no matter how long the entire flight distance is, the real-time properties of the proposed algorithm are always guaranteed. Also, a properly chosen receding horizon can work like a filter to remove unreliable information for the far future. A larger N results in heavier online computational burden, but if N is too small, the RHC algorithm becomes "shortsighted", and the performance significantly degrades. A properly chosen N should be a good trade-off on these factors which depend on the dynamics of the systems and the quality of the information.

However, the nature of the receding horizon concept makes the proposed algorithm only taking into account the cost within the receding horizon, which implies shortsightedness in some sense. The introduction of terminal penalty $W_{term}(k)$ in $J_2(k)$ can compensate for this shortsightedness. When applying RHC in online FF path optimization, if no terminal penalty is used, very poor performance even instability (in the sense that the aircraft fails to arrive at the destination airport) is observed in [9]. Several choices of the terminal penalty have been proposed and investigated in [9]. Due to space limit, only one terminal penalty is presented in this paper, which is defined as

$$W_{term}(k) = (\beta|\theta_3|/\theta_4 + 1)dis(P_{last}(k), P_{DA})/v_E \tag{10}$$

where θ_3, θ_4 and β are illustrated in Fig.3. $P_{SA}, P_{DA}, P_{prev}(k)$ and $P_{last}(k)$ are the source airport, the destination airport, the second last waypoint in an optional FF path, and the last waypoint in an optional FF path, respectively,

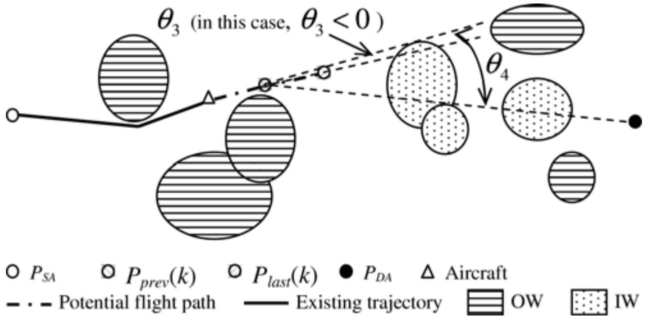

Fig. 3. Definition of a terminal penalty

and IW/OW stands for unavailable airspace regions located on/outside the way directly from $P_{last}(k)$ to P_{DA}. From Fig.3, one can see that: $\theta_3 > 0$ means that the heading of the last sub-trajectory in a potential flight path is over-turning, i.e., aircraft will turn unnecessarily far away from P_{DA}; $\theta_3 < 0$ means under-turning, i.e., aircraft will fly into IW unavailable airspace. $|\theta_3|/\theta_4$ is used to assess how much the over-turning or under-turning is when compared with θ_4. A larger value of $|\theta_3|/\theta_4$ means more excessive turning of the aircraft (either over-turning or under-turning), and will therefore lead to a heavier terminal penalty. $\beta \geq 0$ is a tuning coefficient, and $\beta = 0$ when there is no IW unavailable airspace.

In order to evaluate the proposed RHC algorithm, the simulation system reported in [3] is adopted to set up different FF environments, and the conventional dynamic optimization based GA in [3], denoted as CDO, is also used for the comparative purpose. The proposed RHC algorithm, denoted as RHC, modifies the online optimizer in [3] by taking into account the concept of RHC, as discussed before. More details of the GA optimizer can be found in [3]. In the simulation, unless it is specifically pointed out, the horizon length is $N = 6$, and the terminal penalty $W_{term}(k)$ defined in (10) is adopted for RHC. Six simulation cases are defined in Tab. 1 with different degrees of complexity of the FF environment, where DD stands for the Direct Distance from the source airport to the destination one, and UR for Unavailable Region. In Cases 1 to 3, the UR's are static, while the UR's vary in Cases 4 to 6; in other words, they may move, change in size, and/or disappear 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 one. Numerical results are given in Tables 2 to 4, where 10 simulation runs are conducted under either RHC or CDO for each static case, while 200 simulation runs are carried out for each dynamic case. Firstly, RHC is compared with CDO in static cases, and simulation results are given in Table 2. One can see that CDO achieves the best performance, i.e., the least AFT's, in all 3 cases. This is understandable because conventional dynamic optimization strategy, by its nature, should be the best in terms of

Table 1. Six simulation 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

Table 2. Simulation 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	77.536	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	364.92	5.7970	7.408	15.551
Max. AFT(s)	3966.9	7435.7	14913	3968.7	7480.4	15052

Table 3. Simulation results in dynamic cases

	CDO			RHC		
	Case 4	Case 5	Case 6	Case 4	Case 5	Case 6
Ave. OCT(s)	0.9623	9.448	68.9219	2.4930	3.8419	7.8754
Ave. AFT (s)	4222.0	7475	16192	4221.6	7454.3	15932
Max.OCT(s)	5.317	38.96	347.915	5.8990	6.3190	17.694
Max. AFT(s)	4223.9	8492	16638	4223.1	7995.8	16118

a given performance index when no uncertainties are present. Table 2 shows that the performance of RHC is very close to that of CDO, which implies that RHC works very well in static cases. As for OCT's, RHC is clearly much more efficient than CDO. Since one time-slice is 10-minutes-long, one can see that there is no problem for RHC to run in real-time, while CDO does struggle to complete online computation in some cases. Dynamic cases are our main concern, and some corresponding simulation results are given in Table 3. As for performance, in relatively simple cases like Case 4 and Case 5, CDO and RHC have similar AFT's, while in complicated cases like Case 6, the performance of RHC is better than that of CDO. Again, RHC provides reliable and promising real-time properties against CDO.Tab. 4 highlights that the horizon length N should be properly chosen. If N is too small, the performance is very poor, as is the case of N=1 and N=3 in Tab. 4. However, 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 for $N = 9$.

Table 4. Influence of N on 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	1.336	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	2.539	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	7.305	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	18.25	4.0966	8.5754	17.737
	AFT(s)	3965.9	7407.6	14894	4221.9	7462.4	16074

4 Conclusions

This paper introduces the concept of RHC to the online optimization of flight paths in a dynamical FF environment. Attention is particularly paid to the horizon length and terminal penalty to guarantee the success of the proposed algorithm. 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.

References

[1] C.D. Wickens, et al (1998). Airspace System Integration-The Concept of Free Flight. National Academy Press.
[2] S. Kahne (2000). Annual reviews in Control, 24:21–29.
[3] XB Hu, et al (2004). Eng. Applications of Artificial Intelligence,17:897–907.
[4] N. Durand N, et al (1995). Air Traffic Control Quarterly, 3(3).
[5] S.F. Wu (1990). Studies on the flight performance management, Ph.D. Thesis, Nanjing University of Aero. and Astro., China.
[6] J.A. McDonald and Y. Zhao (2000). Time benefits of free-flight for a commercial aircraft. In *The Proc. of the AIAA GNC Conference*, Denver,CO,USA, 2000.
[7] G. Guastalla, et al (2000). Transportation Sci., 34:394–401.
[8] S. Chand et al (2002). Manufacturing and Service Operations Manag., 4:25-43.
[9] X.B. Hu (2005). *New Results and New Developments in Model Predictive Control.* Ph.D. Thesis, Dept. of Aero. and Auto. Eng., Loughborough University.