Trajectory Control of Multiple Aircraft: An NMPC Approach

Juan J. Arrieta–Camacho[1], Lorenz T. Biegler[1], and Dharmashankar Subramanian[2]

[1] Chemical Engineering Department, Carnegie Mellon University, Pittsburgh, PA 15213
{jarrieta,lb01}@andrew.cmu.edu
[2] IBM Watson Research Center, Yorktown Heights, NY 10598
dharmash@us.ibm.com

Summary. A multi-stage nonlinear model predictive controller is derived for the real-time coordination of multiple aircraft. In order to couple the versatility of hybrid systems theory with the power of NMPC, a finite state machine is coupled to a real time optimal control formulation. This methodology aims to integrate real-time optimal control with higher level logic rules, in order to assist mission design for flight operations like collision avoidance, conflict resolution, and reacting to changes in the environment. Specifically, the controller is able to consider new information as it becomes available. Stability properties for nonlinear model predictive control are described briefly along the lines of a dual-mode controller. Finally, a small case study is presented that considers the coordination of two aircraft, where the aircraft are able to avoid obstacles and each other, reach their targets and minimize a cost function over time.

1 Introduction

Coordination of aircraft that share common air space is an important problem in both civil and military domains. Ensuring safe separation among aircraft, and avoidance of obstacles and no-fly zones are key concerns along with optimization of fuel consumption, mission duration and other criteria. In previous work [10] we developed an optimal control formulation for this problem with path constraints to define the avoidance requirements and flyability constraints. There we considered a direct transcription, nonlinear programming strategy solved with the IPOPT solver [11]. Results for conflict resolution, using detailed flight models and with up to eight aircraft, were obtained quickly, and motivated the implementation of such strategy in real time.

The level of information for these problems, including recognition of obstacles and the presence of other aircraft, evolves over time and can be incomplete at a given instant. This motivates the design of an on-line strategy able to consider new information as it becomes available. For this purpose we propose a nonlinear model predictive control (NMPC) approach. This approach integrates real-time optimal control with higher level logic rules, in order to assist mission design for flight operations like collision avoidance and conflict resolution. In this work,

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 629–639, 2007.
springerlink.com

such integration is achieved by coupling a Finite State Machine (FSM) with an NMPC regulator. The FSM receives the current state of the environment and outputs a collection of sets, which is used to alter a nominal optimal control problem (OCP) in the NMPC regulator. For instance, the detection of a new obstacle leads the FSM to add a new element to the relevant set. The update then alters a nominal OCP by adding the constraints pertinent to the obstacle just detected, thus leading to an optimal avoidance maneuver.

In the next section we derive the multi-stage NMPC problem formulation. Within this framework the NMPC regulator incorporates a 3 degree-of-freedom nonlinear dynamic model of each aircraft, and considers a path constrained OCP that minimizes a performance index over a moving time horizon. In addition, we describe characteristics of the NMPC formulation that allow the aircraft to meet their targets. Stability properties for NMPC are discussed and adapted to the particular characteristics of this application in Section 3. In Section 4, our overall approach is applied to a small case study which demonstrates collision avoidance as well as implementation of the NMPC controller within the FSM framework. Finally, Section 5 concludes the paper and presents directions for future work.

2 Optimization Background and Formulation

We begin with a discussion of the dynamic optimization strategy used to develop our NMPC controller.

Optimization of a system of Differential Algebraic Equations (DAEs) aims to find a control action $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ such that a cost functional is minimized. The minimization is subject to operational constraints and leads to the following Optimal Control Problem (OCP):

$$\begin{aligned} \min_{u} \quad & J[z_d(t_F), t_F] \\ \text{subject to: } \dot{z_d} &= f_d[z_d, z_a, u], \qquad t \in T_H \\ 0 &= z_d(t_I) - z_{d,I} \\ 0 &= f_a[z_d, z_a, u], \qquad t \in T_H \\ 0 &\leq g[z_d, z_a, u, t], u(t) \in \mathbb{U}, \, t \in T_H \end{aligned} \tag{1}$$

where $z_d \in \mathbb{R}^{n_d}$ and $z_a \in \mathbb{R}^{n_a}$ are the vectors of differential and algebraic variables, respectively. Given $u(t)$, a time horizon of interest $T_H := [t_I, t_F]$ and appropriate initial conditions $z_d(t_I) = z_{d,I}$, the dynamic behavior of the aircraft can be simulated by solving the system of DAEs: $\dot{z_d} = f_d[z_d, z_a, u]$, $f_a[z_d, z_a, u] = 0$, with this DAE assumed to be index 1. Notice that some constraints are enforced over the *entire* time interval T_H. In this study, we solve this problem with a direct transcription method [4, 5], which applies a simultaneous solution and optimization strategy. Direct transcription methods reduce the original problem to a finite dimension by applying a certain level of *discretization*. The discretized version of the OCP, a sparse nonlinear programming (NLP) problem, can be solved with well known NLP algorithms [5] like sequential quadratic programming or interior point methods. The size of the NLP resulting from the discretization procedure can be very large, so the NLP algorithm used for the solution must be suitable for large scale problems.

In this work, the OCP is transcribed into an NLP via collocation on finite elements. As described in [5], the interval T_H is divided into n_E *finite elements*. Within element i, the location of collocation point j occurs at the scaled root of an orthogonal polynomial. In this work, roots of Radau polynomials are used, as they allow to stabilize the system [3] when high index constraints are present. State profiles are approximated in each element by polynomials; differential states are represented by monomial basis polynomials while algebraic states and controls are represented by Lagrange basis polynomials. These polynomials are substituted into the DAE model and the DAE is enforced, over time, at Radau collocation points over finite elements. Continuity across element boundaries is also enforced for the differential state profiles. With this approximation, the optimal control problem (1) can be written as:

$$\begin{aligned} \min \quad & \phi(w) \\ \text{subject to:} \quad & c(w) = 0 \\ & w_L \leq w \leq w_U . \end{aligned} \tag{2}$$

Here, the equality constraint vector $c(w)$ contains the discretized differential equations and constraints of (1). Notice that inequality constraints are enforced as equalities via slack variables. In a similar manner, the vector w consists of the polynomial coefficients for the state, control, algebraic and (possibly) slack variables.

The NLP (2) is solved using a primal-dual interior point method. Specifically, we use the Interior Point OPTimizer– IPOPT [11]. This solver follows a barrier approach, in which the bounds on the variables of the NLP problem (2) are replaced by a logarithmic barrier term added to the objective function, and a sequence of these barrier problems is solved for decreasing values of the penalty parameter. In essence, IPOPT approaches the solution of (2) from the interior of the feasible region defined by the bounds. A detailed description of IPOPT, both from the theoretical and algorithmic standpoints, can be found in [11]. In this study, IPOPT is used through its interface with AMPL [7], a modeling language that eases the problem declaration and provides the solver with exact first and second derivatives via automatic differentiation.

2.1 Nonlinear Model Predictive Control

If a perfect model is available for dynamic behavior of the aircraft, as well as full information regarding the surrounding environment, an *a priori* computation of the optimal control actions would be possible. However, neither of these occur in practice; the dynamic models merely approximate the behavior of the aircraft, and the system operates in a partially unknown airspace. An alternative to handle the modeling inaccuracies and relative lack of information, is to compute the optimal controls (maneuvers) in real time.

Nonlinear Model Predictive Control (NMPC) is a closed loop control strategy in which a nonlinear model of the system is used to compute an optimal control via the solution of an optimal control problem. This computation is performed in real time, at every sampling interval [1]. Among the main advantages of NMPC, is the ability to compute the control using higher fidelity nonlinear models (as opposed to linear-model approximation of the dynamics) and impose constraints explicitly. For a thorough overview of both the generalities and formal treatment of NMPC and related on-line control strategies, please refer to [1, 6, 8].

In the context of the methodology presented above, the NMPC controller requires us to formally represent the DAE model in (2) as the discrete time, nonlinear, autonomous system

$$z(k+1) = \bar{f}[z(k), u(k)], \tag{3}$$

where $z(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ are, respectively, the (differential) state and control variables, evaluated at time points t_k with integers $k > 0$. (Note that since the DAE system in (2) is index one, the algebraic variables can be represented as implicit functions of $z(k)$.) The nonlinear function $\bar{f} : \mathbb{R}^{n \times m} \mapsto \mathbb{R}^n$ is assumed to be twice continuously differentiable with respect to its arguments, and the evolution in (3) results from the solution of the DAE in (1). The goal is to find a control law such that a performance index is minimized, and both states and controls belong to a given set: $z(k) \in \mathbb{Z}$ and $u(k) \in \mathbb{U}, \forall k$.

It is important to distinguish between the *actual* states and controls, and the *predicted* or *computed* states and controls. For this reason, we introduce the following notation: $z(k)$ is the *actual* state of the physical system at time step k, which is reached by the *actual* implementation of the control action $u(k-1)$. On the other hand, $\bar{z}(l)$ is the *predicted* state from time step k, l steps into the future, by the simulation of the system with the *computed* control action $\bar{u}(l-1)$.

At time step k, we define the performance index

$$J[z(k), \bar{u}, N] = \sum_{l=0}^{N-1} \psi[\bar{z}(l), \bar{u}(l)] + F[\bar{z}(N)], \tag{4}$$

which is a function of the initial condition $z(k)$, the vector of control actions $\bar{u}$ used to simulate the system, and the length of the prediction horizon N. In the interest of finding the best performance index, an optimization problem is formulated:

$$\begin{aligned} \min_{\bar{u}} \; J[z(k), \bar{u}, N] &= \sum_{l=0}^{N-1} \psi[\bar{z}(l), \bar{u}(l)] + F[\bar{z}(N)] \\ \text{subject to:} & \\ \bar{z}(l+1) &= \bar{f}[\bar{z}(l), \bar{u}(l)] \\ \bar{z}(0) &= z(k) \\ \bar{g}[\bar{z}(l), \bar{u}(l)] &\leq 0 \\ \bar{u} &\in \mathbb{U}. \end{aligned} \tag{5}$$

where the inequality constraints $\bar{g}[\cdot] \leq 0$ correspond to inequality constraints from (2).

This problem can be considered within the framework of (1) and is solved using the direct transcription strategy outlined in the previous section. The solution to (5) is given by $\bar{u}_k^* = [\bar{u}^*(0), \bar{u}^*(1), \ldots, \bar{u}^*(N-1)]$.

In NMPC, the first element of $\bar{u}^*$ is implemented on the actual system, defining the control law $u(k) = \bar{\kappa}[z(k)] := \bar{u}^*(0)$ that leads to the closed loop system $z(k+1) = \bar{f}[z(k), \bar{\kappa}[z(k)]] = \bar{f}[z(k), u(k)]$. At the next sampling interval $k+1$, a new control action, $u(k+1)$, is found in a similar manner.

2.2 Multistage Controller

In the application at hand, some information about the environment is not known a priori. For instance, the presence of an obstacle could be unknown until such obstacle is within radar distance of the aircraft. For this reason, it is not possible to include all the pertinent constraints in the optimization problem a priori. Also, a new way-point might be assigned to an aircraft at any given time. These difficulties can be overcome by using a multi-stage controller. Specifically, we couple a finite state machine (FSM) with the NMPC controller.

An FSM is an event-driven system, that makes a transition from one state to another when the condition defining the transition is true. In our application, to each state of the FSM corresponds a set $\mathcal{S}$ relevant to a nominal OCP. The OCP is formed by constraints and variables that are indexed by $\mathcal{S}$. The FSM is also able to alter parameters relevant to the OCP, for instance, the position and radius of a recently detected obstacle. The new information is passed to the nominal OCP by altering the set $\mathcal{S}$, and irrelevant information is removed in a similar manner.

The states in the FSM correspond to the modes of operation: **provide_mission**: which assigns missions to to the corresponding aircraft and issues an appropriate trigger, **wait**: which forces aircraft to wait until a mission is assigned, **cruise**: where control actions are computed and implemented for each aircraft to reach the setpoint defined by the current mission and detect obstacles, **avoid**: which obtains geography (e.g. position and radius) of detected obstacles and formulates appropriate constraints for the **cruise** mode, **assess_outcome**: which verifies whether the targets have been reached and triggers new missions, and **lock** mode, described below. Additional information related to the FSM can be found in [2]. The NMPC controller, formed by a nominal OCP whose constraints and variables are indexed by the set $\mathcal{S}$, is embedded into the **cruise** and **lock** modes.

The NMPC block solves an OCP that includes the following constraints: **DAE system** describing the dynamic response of the aircraft and flyability constraints (like stall speed, maximum dynamic pressure, and others); **conflict resolution** enforcing a minimum radial separation among aircraft; and **obstacle avoidance** enforcing a minimum separation between aircraft and obstacles. A schematic view of the coupling between the FSM and the NMPC block is presented in Figure 1.

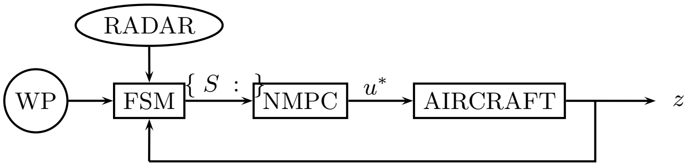

Fig. 1. Schematic view of the coupling between the FSM and the NMPC block. The current state together with the set-point and radar readings cause the FSM to update the set S which, in return, alters the structure of the nominal OCP within the NMPC block. The optimal control action u^* obtained in the NMPC block is implemented in the system.

In order for each aircraft to reach a given target in an efficient manner, we define the following objective functional for each prediction horizon k:

$$J[z(k), \bar{u}, N] = \sum_i P_i \left[\frac{1}{2} \int_{t_I^k}^{t_F^k} \left(\bar{u}_{1,i}^2 + \bar{u}_{2,i}^2 \right) dt + \eta_i \Phi(\bar{\mathbf{z}}_i, \mathbf{z}_i^{sp})|_{t_F^k} \right] \tag{6}$$

where $u_{1,i}$ and $u_{2,i}$ are the forward and vertical load factors for aircraft i, respectively. In this application it suffices to consider the load factor as the acceleration experienced by the aircraft. We choose to minimize the load factor terms because there is a direct relation between the acceleration of an aircraft and fuel consumption (higher forward or upward accelerations require more fuel) and pilot safety and comfort.

In (6), the contributions of each aircraft are added up, weighted by a factor $P_i \geq 0$ representing the *priority* of each aircraft. Each contribution includes an integral term, that measures the control effort, and an exact penalty term $\Phi(\bar{\mathbf{z}}_i, \mathbf{z}_i^{sp})|_{t_F^k} = \|\bar{\mathbf{z}}_i(t_F^k) - \mathbf{z}_i^{sp}\|_1$, weighted by a factor $\eta_i \gg 0$, that enforces the target.

The target is imposed with an exact penalty term and not with a hard constraint, because it is not possible to know *a priori* when the aircraft will reach the target. If the aircraft are far from their targets, the exact penalty formulation encourages a closer distance to the target, without necessarily reaching it. On the other hand, *if the target can be reached within the time horizon of the NMPC controller, the exact penalty is equivalent to a hard constraint*, provided that the weighting factor η_i is sufficiently large (see [9]; in this work we use $\eta_i = 10^5$). If the target can be reached within the time horizon k, the FSM transitions to the **lock** mode, which reduces the interval t_F by one unit at $k+1$, until the target is reached. The penalty term has important implications on the stability properties, as discussed in the next section. The objective functional (6) together with the above constraints and appropriate initial conditions specify the OCP given to the NMPC block.

The FSM was implemented in MATLAB as a collection of `switch` statements. The optimization step of the NMPC block is implemented in AMPL using

IPOPT as the NLP solver. Communication between the NMPC and the FSM was carried out through files updated in every time horizon. In this work, we use a prediction horizon of $N = 10$, $t_F - t_I \leq 60$ seconds, and a sampling time of 6 seconds. We acknowledge that the length of the prediction and implementation horizons are critical tuning parameters in NMPC. The main trade-off is that longer horizons provide solutions closer to the off-line, full-length optimization, but require longer CPU times.

3 Stability Properties

In this section we consider the nominal stability for a particular stage of our controller which solves (5). Ensuring stability of $z(k+1) = \bar{f}[z(k), h(z(k)]$ is a central problem in NMPC, and can be achieved by several methods [1]. The aim is to find a control law such that the origin for the closed-loop system (without loss of generality assumed to be the setpoint) is asymptotically stable, with a large region of attraction. All techniques require some modification to the OCP (5) solved on-line, but have the following in common:

- a positive definite, scalar cost function or performance index $J(\cdot)$, with a final penalty term $F(\cdot)$,
- a nonlinear model $\bar{f}(\cdot)$ describing the dynamic response of the system, from an initial condition $z(0)$, N steps into the future.
- control constraints $\mathbb{U}$ and state constraints $\mathbb{Z}$, and
- a terminal constraint $z \in \mathbb{Z}_s$.

For instance, setting $N = \infty$ in (5) leads to an Infinite Horizon nonlinear control (IH), which can be proved to provide a stabilizing control law. However, its implementation requires the approximation of an infinite summation, leading to a difficult optimization problem that normally cannot be solved in a reasonable time frame. The difficulties associated with the implementation of the IH, motivated the development of control strategies based on finite-horizon (FH) optimization. In particular Nonlinear Receding Horizon (NRH) control, is a group of methodologies (of which NMPC is a member) that *specifically* aims to solve problem (5).

Important cases of NRH include the *zero-state* (ZS) terminal constraint for which the terminal cost $F(\cdot) \equiv 0$ and $\mathbb{Z}_s = \{0\}$, meaning that the end point constraint is enforced as a hard constraint. ZS can guarantee stability if there exists a nonempty neighborhood of the origin $Z^C(N)$ for which it is possible to find a control sequence $u(k),\ k = \{0, \ldots, N-1\}$ capable of driving $z(k+1) = \bar{f}[z(k), u(k)]$ to the origin in N steps (i.e. $z(N) = 0$), and the initial condition $z(0)$ is within that neighborhood. An important drawback of the ZS methodology is that it can require prohibitively long time horizons for $Z^C(N)$ to exist and, even if $Z^C(N)$ exists for a short horizon, this might result in excessive control effort. In addition, satisfying the equality constraint can be computationally demanding.

The idea of replacing the equality constraint by an inequality, which is much easier to satisfy, motivates the *Dual Mode* (DM) controller, for which the $F(\cdot)$ is chosen as an upper bound on the cost of some stabilizing controller that regulates the system, whenever it is within the neighborhood of the origin defined by $\mathbb{Z}_s$. In the implementation of DM, FH control is applied until $z(t) \in \mathbb{Z}_s$, at which point the controller switches to a stabilizing state feedback controller $u(t) = \kappa(z(t))$. Stability of the DM controller can be paraphrased by the following theorem [1, 6, 8]:

Theorem 1 (Nominal Stability of NMPC). *Consider the system described by (3), then with advancing $k > 0$, the NMPC controller leads to a monotonic decrease of $J[z(k)]$ and it is asymptotically stable within a region at least twice the size of $\mathbb{Z}_s$, if we assume:*

- $F(z) > 0, \forall z \in \mathbb{Z}_s \backslash \{0\}$,
- there exists a local control law $u = \kappa(z)$ defined on $\mathbb{Z}_s$, such that $\bar{f}(z, \kappa(z)) \in \mathbb{Z}_s, \forall z \in \mathbb{Z}_s$, and
- $F(\bar{f}[z, \kappa(z)]) - F(z) \leq -\psi[z, \kappa(z)], \forall z \in \mathbb{Z}_s$.

We can apply this result directly for a particular assigned set of way-points if we assume that the cost of some stabilizing controller (including manual control of the aircraft) can be overestimated by the exact penalty term in (6) over the entire test field, i.e., $F(z(t_F^k)) = \eta \Phi(\mathbf{z}_i, \mathbf{z}_i^{sp})|_{t_F^k}$ and $\mathbb{Z}_s = \mathbb{Z}$. A practical realization of this assumption occurs for η suitably large. Because of this assumption and the implementation of the exact penalty term, the stability result applies to (5) for aircraft only in the **cruise** and **lock** modes, and the performance index decreases monotonically within these modes. However, we caution that this result does not imply monotonic decrease over the *entire* set of missions. As new missions are assigned or as different constraints are added in the **avoid** mode, the performance index may indeed increase. The analysis of overall stability (the global case) is left for future work.

4 Two Aircraft Case Study

We now consider the case of two aircraft that accomplish separate missions (defined by way-points (wp)) in a constrained airspace. The trajectory through which a given aircraft reaches the target must be obstacle free and, at every point in time, the different aircraft must maintain a safe distance from each other. The airspace is known to have obstacles, for some of which the position and size are known *a priori*. The aircraft are also equipped with radar, which can detect a previously unknown obstacle. It is assumed that the radar is able to determine both shape and location of a given obstacle within its scope.

Aircraft dynamics can be described by the state variables, $z_d =^{\mathrm{T}} [x\ y\ h\ v\ \chi\ \gamma]$, corresponding to east-range, north-range, altitude, air speed, heading angle and flight path angle, respectively. The control variables are given by $u =^{\mathrm{T}} [u_1\ u_2\ u_3]$ and correspond to forward load factor, vertical load factor, and bank angle,

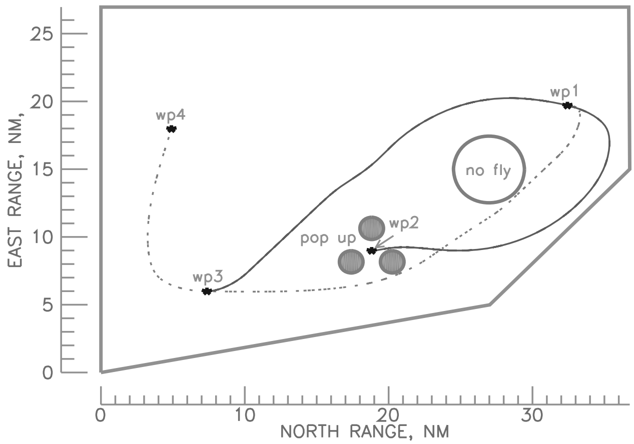

Fig. 2. Test field for the case studies. Aircraft 1 (dotted) and 2 (solid).

respectively. After some simplifying assumptions, the equations of motion are stated for each aircraft as:

$$\begin{aligned} \dot{x} &= v\cos\gamma\cos\chi, \ \dot{v} = g(u_1 - \sin\gamma), \\ \dot{y} &= v\cos\gamma\sin\chi, \ \dot{\chi} = -\tfrac{g}{v}\left(\tfrac{u_2\sin u_3}{\cos\gamma}\right), \\ \dot{h} &= v\sin\gamma, \qquad\quad \dot{\gamma} = -\tfrac{g}{v}\left(u_2\cos u_3 + \cos\gamma\right), \end{aligned} \tag{7}$$

where g is the standard acceleration. In order to produce flyable maneuvers, constraints defining the flight envelope and other restrictions modeling the performance capabilities of the aircraft are added to the formulation. Using SI units, we have the air density, $\rho = 1.222\exp(-h/9144.0)$ and bounds on velocity, $v \geq v_S\sqrt{9144.0/\rho}$, $v^2 \leq 2q_{\max}/\rho$ and control variables $u_j \in [u_{j\ \min}, u_{j\ \max}]$, $j = 1,\ldots 3$. Here v_S is the stall speed and $q_{\max}$ is the maximum dynamic pressure.

We now consider two aircraft flying in the test field presented in Figure 2, where the three small cylinders are pop-up obstacles; their presence is not known *a priori*. Two missions are assigned to each aircraft: **wp1→wp3→wp4** for aircraft 1, and **wp3→wp1→wp2** for aircraft 2. Using the proposed multi-stage NMPC approach, both aircraft are able to reach the assigned way-points, while avoiding obstacles and (locally) minimizing the load factor terms. In Figure 3, notice that aircraft 1 reached the second way-point in 560 seconds, while aircraft 2 reached the second way-point in 660 seconds. The optimization problem solved at each NMPC horizon varies in size, since different information is added and subtracted as the flight evolves. The largest NLP solved consists of 1406 variables and 1342 constraints. The average CPU time required to solve the NMPC problem was 0.2236 seconds, and the maximum CPU time required was of 0.8729 seconds.[1]

[1] SUN Java Workstation: dual AMD64-250 processors @ 2.4GHz with 16GB RAM, running Linux operating system.

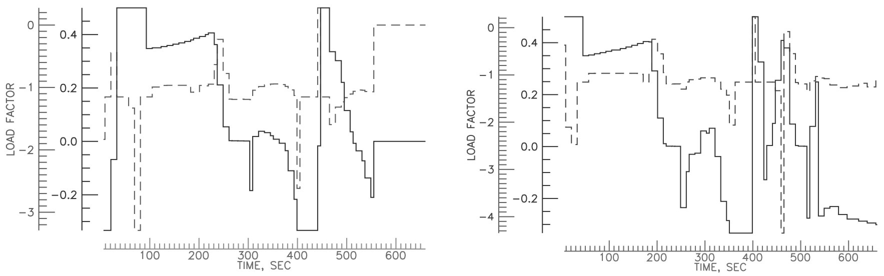

Fig. 3. Optimal control actions for aircraft 1 (left) and 2 (right). Forward load factor (solid, inner axis) and vertical load factor (dashed, outer axis).

5 Conclusions and Future Work

We present a multi-stage, NMPC-based control strategy for the real time coordination of multiple aircraft. The controller couples a finite state machine with a nonlinear model predictive controller. With the proposed methodology, it is possible to coordinate several aircraft, such that they can perform several missions in partially unknown environments. The main advantage of this controller is its ability to consider new information as it becomes available and its ability to define several modes of operation.

A case study with two aircraft was presented. It is noticed that the CPU times required to compute the control action are small compared to the physical time of the implementation (3.7%, on average). Stability of the controller is achieved based on properties of the dual mode NMPC controller and robustness can be promoted by tuning certain parameters within the NMPC regulator. Although good results can be obtained with the methodology presented, it is desirable to investigate more general conditions under which the controller is stable for the entire FSM, in the presence of disturbances, and also with known robustness margins.

The NMPC controller could also be used to assist in the decision-making process involved in the unmanned control of aerospace vehicles. We believe that the concept of combining the versatility of hybrid systems theory with the power of large-scale optimal control can prove very useful in the design of advanced control strategies for the efficient coordination of multiple aircraft.

References

[1] Allgöwer, F., T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, *Nonlinear predictive control and moving horizon estimation: An introductory overview*, in Advances in Control, Highlights of ECC99, P. M. Frank, ed., Springer, 1999, pp. 391.

[2] Arrieta-Camacho, J. J., L. T. Biegler and D. Subramanian, *NMPC-Based Real-Time Coordination of Multiple Aircraft*, submitted for publication (2005).

[3] Asher, U. M. and Petzold, L., *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM, 1998.

[4] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM, Philadelphia, PA, 2001.

[5] Biegler, L. T., *Efficient Solution of Dynamic Optimization and NMPC problems," Nonlinear Model Predictive Control*, edited by F. Allgöwer and E. Zheng, Birkhäuser, 2000, pp. 219.

[6] de Nicolao, G., L. Magni, and R. Scattolini, *Stability and robustness of nonlinear receding horizon control*, in Nonlinear Predictive Control, F. Allgöwer and A. Zheng, eds., vol. 26 of Progress in Systems Theory, Basel, 2000, Birkhauser, pp. 3, 23.

[7] Fourer, R., Gay, D., and Kernighan, B. W., *A Modeling Language for Mathematical Programming*, Management Science, Vol. 36, 1990, pp. 519-554.

[8] Mayne, D., *Nonlinear model predictive control: Challenges and opportunities*, in Nonlinear Predictive Control, F. Allgöwer and A. Zheng, eds., vol. 26 of Progress in Systems Theory, Basel, 2000, Birkhäuser, pp. 23,44.

[9] Nocedal, J. N., and S. J. Wright, *Numerical Optimization*, Springer, Berlin (1998).

[10] Raghunathan, A., V. Gopal, D. Subramanian, L. T. Biegler and T. Samad, *Dynamic Optimization Strategies for 3D Conflict Resolution of Multiple Aircrafts*, AIAA J. of Guidance, Control and Dynamics, 27 (4), pp. 586-594 (2004).

[11] Wächter, A., and L. T. Biegler, *On the Implementation of an Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming*, Mathematical Programming, 106, 1, pp. 25-57 (2006)