

Practical approach to real-time trajectory tracking of UAV formations

Bálint Vanek*, Tamás Péni†, József Bokor‡ and Gary Balas§

* † ‡Systems and Control Laboratory Computer and Automation Research Institute of the Hungarian Academy of Sciences 1111 Budapest XI. Kende u. 13-17. Hungary

* §Department of Aerospace Engineering and Mechanics, University of Minnesota 15 Akerman Hall 110 Union St. SE Minneapolis, MN 55455

*corresponding author: vanek@aem.umn.edu

Abstract—An Unmanned Aerial Vehicle (UAV) formation in a leader-follower structure, where the UAVs are flying a common trajectory determined by a route planner hosted on the leader is considered. The path description is compressed by polynomial functions with respect to the flight envelope constraints and transmitted to the followers, where a Model Predictive Control (MPC) outer loop controller specifies the command signals for the \mathcal{H}_∞ locally controlled dynamics with respect to the nonlinear constraints of the aircraft dynamics. Real time feasibility issues associated with the design are discussed.

I. INTRODUCTION

The problem of controlling multiple vehicle formations has received considerable attention recently [1], [2], [3], [4], [5]. The development of increasingly intelligent Unmanned Aerial Vehicles render more precise tasks possible, e.g. close formation keeping or flying cooperatively in unfamiliar environments. The need to estimate neighbors behavior leads to the use of prediction based methods, in which the future states, positions of the vehicles can be calculated based on the actual status of them. The Model Predictive Control [6] approach is also useful in coping with constrained and nonlinear dynamics. But it has some drawbacks like non-negligible computational time and the property that the system can only be handled as a low frequency sampling time discrete system in which the information flow is limited. These properties were not critical in applications on chemical processes but in aerospace they must be handled well. The present work focuses on the feasibility issues of a trajectory tracking problem solved by an MPC outer loop controller, with respect to computational time and various types of simplified dynamics. In contrast with others [1], [7] the algorithm addresses the real-time feasibility of the computations, copes with boundary conditions and provides continuous, smooth transitions between calculations of different horizons. In cooperative tasks one of the main limitations is the communication delay [8], in leader-follower structures the leader needs to transfer its determined route to the followers, but a simple time stamped waypoint data flow at every sampling time may use irresponsible high amount of bandwidth. To shorten the amount of transmitted data the position coordinates can be approximated by polynomial functions, so with preprocessing before transmission the amount of data can be effectively reduced.

II. PROBLEM FORMULATION

The focus of this work is on a leader-follower structure aircraft formation where every aircraft's role is defined in advance (Fig. 1). Given a low bandwidth communication

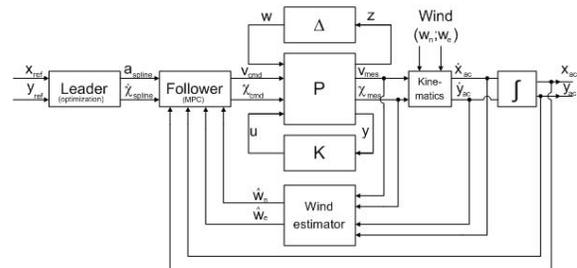


Figure 1: The structure of the complete control loop

channel between every leader follower pair of UAVs, we want each of them to follow the other one's path simultaneously with a constant position offset between them, while respecting the aircraft dynamic response. The communication algorithm must compress the position data being transmitted, to ensure stealth requirements with low radio emission, independently from the types of aircraft. The approach discussed in this paper uses splines to parameterize the designated flight trajectory determined by the leader's route planner. All followers can add their offsets to the spline parameterized trajectory received from their leader. With this information they calculate their own control inputs corrected by the estimate of the actual wind. A nonlinear Model Predictive Control approach is employed to handle the flight envelope constraints of the UAVs. Both the communication and control algorithms make use of optimization based on Sequential Quadratic Programming, which can handle constrained dynamics and the computational time depends on the iteration steps defined in advance. The aircrafts have \mathcal{H}_∞ local multivariable controllers to stabilize their dynamics [9], [10] and achieve desired tracking properties.

This study focuses on horizontal movements, but the approach can be easily generalized to three dimensional flight. After designing a suitable inner loop controller (Section III) the communication algorithm is specified (Section IV) and last the followers outer loop controller which calculates the required control reference signals is specified (Section V). To demonstrate the properties of the design method, examples are presented in Section VI.

III. INNER LOOP CONTROL

The purpose of this paper is to develop leader-follower tracking algorithm which are applicable to current aircrafts. This requires realistic aircraft models and inner-loop con-

trollers be used. Simulation of the leader-follower formation requires precise tracking of trajectory commands, hence an \mathcal{H}_∞ [11], [12] feedback controller is designed to the F-16 aircraft, what can be treated as an UAV [13]. For simplicity the tracking problem is formulated in horizontal plane, hence the aircraft needs to follow velocity (v) and flight course angle (ψ) commands. The position of the aircraft is integrated from the initial coordinates using the kinematics and wind, based on the knowledge of speed and heading angle at the starting point.

$$\dot{x}(\tau) = v(\tau) \cos(\psi(\tau)) + E(\tau) \quad (1)$$

$$\dot{y}(\tau) = v(\tau) \sin(\psi(\tau)) + N(\tau) \quad (2)$$

Assuming linearized dynamics, velocity tracking can be easily achieved by slight modifications in the longitudinal dynamics state space equations. Unfortunately the usual airplane mathematical models are not suitable for the (ψ) tracking purpose. To achieve desired (ψ) command tracking properties, the lateral dynamics state space is extended with $\dot{\psi}$ which can be expressed using Ψ_{NED} and [14], based on the property that in horizontal flight the flight course angle (ψ) and the projection of the yaw angle into the horizontal plane (Ψ_{NED}) differs only by the angle of sideslip (β) [15]. That separation enables the design of an output feedback controller for velocity tracking based on the linearized longitudinal dynamics and design another controller for flight course angle tracking based on the modified linearized lateral dynamics of the airplane.

For the purpose of Model Predictive Control, simulation of the future behavior of the system is necessary. Simplified transfer functions of the $\dot{\psi}_{cmd} \rightarrow \dot{\psi}_{ac}$ and $\dot{\psi}_{cmd} \rightarrow \dot{\psi}_{ac}$ dynamics are determined, based on the original high order \mathcal{H}_∞ controlled mathematical description of the aircraft. First and third order transfer functions are identified experimentally from the frequency domain analysis of the original high order v and $\dot{\psi}$ dynamics for comparison.

IV. TRAJECTORY GENERATION

The leader airplane has a route planner which specifies the points of the intended flight path ($x_{ref}(\tau); y_{ref}(\tau)$) based on geography and the tasks of the formation [7]. The amount of data required depends on the sampling time of the system and the bandwidth required to transfer from leader to followers. So an alternate method is used. One obvious approach is to compress the data flow using splines [1], this allows the route description to be the continuous function of velocity and direction angle and splines are very effectively characterize these kind of functions [16] at the price of increased computational time. This drawback is unavoidable when the coefficients of the spline functions are computed.

In our case the acceleration ($\ddot{x}(\tau)$) and flight course angle rate ($\dot{\psi}(\tau)$) data are parameterized by spline functions which differs slightly from the method used by Papageorgiou [1] where the parameterized functions are $\dot{x}(\tau)$ and $\dot{\psi}(\tau)$. No integration is needed to obtain velocity in that approach though it is less effective in respect of the constraint handling. The approach is also more computational conformable to handle constraints within the optimization than parameterizing Eastern (x) and Northern (y) directional data. This is discussed more detailed later. The form of our spline functions are:

$$\dot{x}(\tau) \in \sum_i \alpha_i i(\tau); \dot{\psi}(\tau) \in \sum_i \beta_i i(\tau) \quad (3)$$

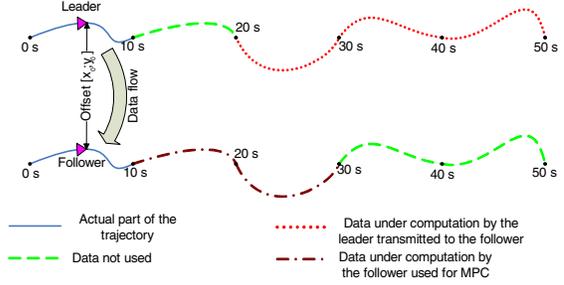


Figure 2: Required data flow between leader and follower ($u=10$ sec; $s=0.1$ sec)

α_i and β_i are the coefficients of the splines and $i(\tau)$ is the i^{th} order B-spline basis function. One key point of the design is to be familiar with the timing of different data flow transfer and the usage of the received items, to take care about the real-time feasibility. If we, for instance, assume h sec sampling time data from the route planner and u sec update frequency ($u \gg h$) in data transfer than for simultaneous flight the non negligible computational time demands the data structure seen on Fig. 2.

The n second long data intervals are parameterized by 6th order B-splines with $\alpha + 5$ coefficients and $\alpha + 11$ knot points with 1 sec equivalent spacing in which the first 5 knot points represent the past so the first 5 coefficients are fixed, obtained from the previous horizon to provide 5th degree fitting between updates and provide the conformance with constraints between horizons, the other coefficients are variables. The n second long data interval is a multiple of the u sec update frequency. The MPC trajectory length needs to be at least 2 times longer than the update rate. The preferred length of the flight trajectory though is 3 times the update rate. This ensures the first 20 sec trajectory is accurate and doesn't penalize the decreasing accuracy associated with the 20-30 sec time interval. Another consideration in connection with the data timing is the effect of computational time. While the follower develops its command signals for the next horizon the leader must compress the trajectory one data interval ahead to be able to transmit that before it will be needed. The optimization called at each refresh has the following form:

$$\min_{\alpha_i; \beta_i} \int_{t_0}^{t_0+t_H} \left\| \begin{matrix} sp(\tau) - damp(\tau) \\ sp(\tau) - damp(\tau) \end{matrix} \right\|_2^2 + \alpha_5 \dot{x}_{sp}^2(\tau) + \beta_6 \dot{\psi}_{sp}^2(\tau) \quad (4)$$

The coefficients of the splines are obtained by optimizing this function. It can be seen that the optimization wants to keep the spline as close to the reference as possible. An additional component penalizes the actuator usage with α_5 and β_6 weights. The routine also pays attention to the constraints from the beginning (t_0) to the end ($t_0 + t_H$) of the horizon. The following variables are constrained in the optimization:

$$1 \leq \dot{x}(\tau) \leq 2 \left[\begin{matrix} \cdot \\ \cdot \end{matrix} \right]; \quad \dot{\psi}(\tau) \leq 3 \left[\begin{matrix} \cdot \\ \cdot \end{matrix} \right]^2$$

The $damp(\tau)$ and $\dot{\psi}_{damp}(\tau)$ functions are part of the modified reference trajectory. Since the reference trajectory from the route planner and its description by polynomial

functions may differ at every starting point due to the continuity of the splines, an exponential damping function is used to reduce the difference gradually to avoid violating flight envelope margins. It is also suitable to handle changes in the offset between the aircraft, because the damping function can decrease the effect of switches in desired distance between aircrafts.

$$damp(\cdot) = ref(\cdot) + (sp(0) - ref(0)) \cdot e^{-k_x t} \quad (5)$$

In the following I will omit the description of the corresponding y coordinates hence they can be calculated the same way as x terms with respect to the different wind direction terms. The initial values for spline coefficients are calculated from the trajectory received from the route planner, the time stamped values of the knot points are equivalent with the $\dot{ref}(\cdot)$ and $\ddot{ref}(\cdot)$ values at the same time. This estimation is based on the correspondence between the coefficients and the actual values of the splines at given knot points assuming relatively smooth functions.

The problem is solved with MATLAB's *fmincon* Sequential Quadratic Programming based routine with fixed iteration steps. It handles the nonlinear constraints of the problem and because the starting point is accurately selected it converges to the reference quite rapidly. The main problem arises from the non-convexity of the task, it can be very sensitive to the minimum and maximum changes in variables for finite difference gradients especially when the constraints are derivatives of the variables. To avoid using constraints as variable derivatives, we parameterized $\dot{(\cdot)}$ and $\ddot{(\cdot)}$. This requires time consuming integrations, in spite of this the calculations are computational tractable and faster in practice than working with $\dot{(\cdot)}$ and $\ddot{(\cdot)}$.

- In the case of parameterizing velocity, the acceleration is calculated based on that parameterization. Since the constraints on velocity don't limit the values of acceleration significantly, the velocity change within the constraints on speed can be very high between two sampling points. Hence the bounds can be easily violated. This leads to the algorithm not converging directly to the optimum.
- With an acceleration parametrization it is harder to violate the velocity margins. This is due to it being more difficult to reach the maximum velocity with several time step. Hence the probability of having unfeasible solutions with call to the optimization function is lower.

V. MPC CONTROL OF THE FOLLOWERS

The follower receives the route description when it is actually flying its previous horizon. Hence first an estimation is needed to determine the position of the aircraft at the beginning of the next horizon (Fig. 3). It uses the previously calculated command signals corrected with an estimate of the actual wind based on a simplified model of the aircraft. The optimization of the command signals for the next horizon starts from that point. In our problem, the followers calculate their command signals every u sec for a 2 sec long horizon based on measured and previously determined data like the position, velocity and direction of the aircraft and the actual horizon's command signal flow (Fig. 4). To handle the effect of the wind, the synthesized trajectory compensates for its effects. (Fig. 5).

Similar to the trajectory generation, the command signals are also compressed in spline form to connect the relatively low degree of freedom spline knot points with the required

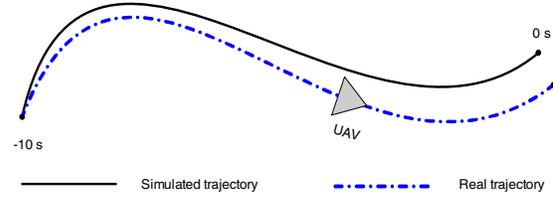


Figure 3: Simulated position of the UAV at the beginning of the next horizon ($u=10$ sec)

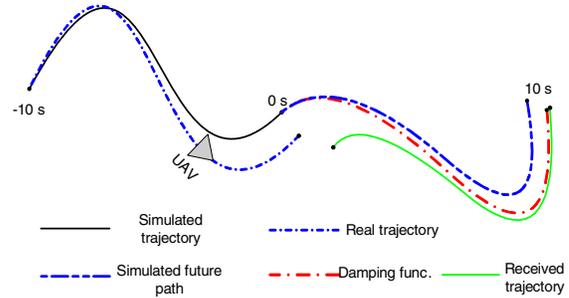


Figure 4: The reference and the intended path ignoring wind ($u=10$ sec)

sampling time for the controller. There are two alternate ways to parameterize the command signals:

- Describe the coordinates of the aircraft $(x(t); y(t))$ via polynomial functions. This method uses dynamic inversion [17] to calculate the required command signals. It is easy to see that in the optimization routine we cannot estimate the bounds of the coordinates. Hence it is hard to find an initial estimate to the coordinates and the problem cannot be well defined. Not to mention that the optimization function contains the 2nd derivative of the command signals which means 4th derivative of the coordinates.
- Describing $\dot{cmd}(\cdot)$ and $\ddot{cmd}(\cdot)$ with splines and then performing the required optimization needs more computational time each iteration step. Though the problem is better formulated and less steps are necessary to obtain the required precision.

In our UAV formation problem, the second method is used. The connection between different horizons is provided by the usage of integration, so the initial value of

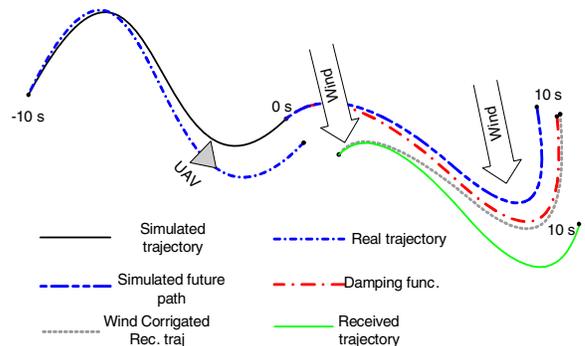


Figure 5: The reference and the intended path including wind ($u=10$ sec)

a given horizon ($\dot{c}_{md}(0)$ and $c_{md}(0)$) is obtained from the previous command signal's value at that time. That leads to a continuous command signal flow, which secure the constraint fulfillment in points between horizons. The horizons are 2 times longer than the refreshing time and all of the coefficients connected to the equivalently divided spline knot points are optimized. The modification in the spline coefficients has an indirect effect on the command signals because an additional integration is needed to obtain $\dot{c}_{md}()$ and $c_{md}()$. The starting point for the simulation of the aircraft's movement with MPC controller ($\dot{a}_{c,approx}()$ and $a_{c,approx}()$) is based on the previously calculated simulation results. Namely applying the last command signal flow, starting from the last measured position. We formulate the optimization function similar to the function described above:

$$\min_{\alpha_{MPC}; \beta_{MPC}} \int_{t_0}^{t_{H,MPC}} \left\| \begin{matrix} \dot{d}() - \dot{model}() \\ d() - model() \end{matrix} \right\|_2^2 + \tau \cdot \dot{c}_{md}()^2 + \delta \cdot c_{md}()^2 \quad (6)$$

Where the $\dot{d}()$, $d()$ damping function is calculated from the received trajectory, and $\dot{model}()$, $model()$ is the simulated behavior of the simplified aircraft model.

A. Past part of the route

First an approximation is needed to estimate the aircraft's position at the beginning of the next horizon (Fig. 3).

$$\dot{a}_{c,approx}() = \dot{v}(c_{md}()) = -0 \quad (7)$$

$$a_{c,approx}() = \chi(c_{md}()) = -0 \quad (8)$$

The effect of wind based on the measurements is included:

$$\dot{a}_{c,approx}() = \dot{a}_{c,approx}() \cdot \cos(\hat{a}_{c,approx}()) + \hat{E}(-) \quad (9)$$

The estimated position of the aircraft is calculated at the point from where the optimization starts:

$$a_{c,approx}() = a_{c,approx}(-) + \int_{-n}^0 \dot{a}_{c,approx}() \quad (10)$$

B. Required route

Since we want to design a route which is affected by wind, we need to correct the desired path using wind data (Fig. 5):

$$\dot{w}_{c.ref}() = \dot{s}_p() - \hat{E}(-) \quad (11)$$

An exponential damping function is used to gradually decrease the difference between the actual position of the aircraft and the demanded trajectory. In addition, a constant offset between the leader and the follower is added:

$$d() = s_p(0) + \int_0^{2n} \dot{w}_{c.ref}() + (\dot{a}_{c,approx}(0) - \dot{w}_{c.ref}(0)) \cdot e^{-k_x t} + o \quad (12)$$

C. Modelled route of the aircraft

The heart of the MPC algorithm is the part where we design a suitable control signal flow with which the modelled aircraft flies as close to the required path as possible. This part starts from the previously calculated $\dot{a}_{c,approx}(0)$; $a_{c,approx}(0)$ point. The modelled position

of the aircraft is calculated by simulating the aircrafts dynamics \dot{v} and χ with the input signal function received from the objects of optimization ($\dot{c}_{md}(MPC)$ and $c_{md}(MPC)$) after an integration. The starting point of the integration ($\dot{c}_{md}(0)$; $c_{md}(0)$) is equal to the previous horizon's value at $t = 10$:

$$\dot{model} = \dot{v}(c_{md}()); \quad model = \chi(c_{md}()) \quad (13)$$

Then we can obtain the position of the model from previously calculated data:

$$model() = a_{c,approx}(0) + \int_{t_0}^{t_0+2n} \dot{model}() \cos(c_{md}()) \quad (14)$$

The following values must be constrained during flight. The optimization algorithm include these as nonlinear constraints:

$$\max |a_{c,long}| \leq 3; \quad \max |a_{c,lat}| \leq 4 \\ 5 \leq |a_{c}| \leq 6; \quad \max |a_{c}| \leq 7$$

The initial values of the $\dot{c}_{md}()$ and $c_{md}()$ spline coefficients are obtained from the received spline coefficients ($\dot{s}_i()$; $s_i()$) without modifications, because in optimal case (neglecting wind) the coefficients of the route describing spline coefficients and the spline coefficient parameterizing the command signal are close and the iteration does not need more precision to achieve convergence. The required trajectory is corrected to comply with the optimization function. First, the starting point of the horizons is computed based on the last measured position and the actually implemented command signals with the simplified aircraft model. Second, the starting point of the received trajectory and the actual aircraft position are different so an exponential damping function is used to smooth out the error and drive the aircraft. Third, the algorithm deals with wind so the required trajectory and the trajectory to which the optimization focuses must differ by the effect of wind. The wind is estimated at the beginning of each optimization horizon, what requires inertial/air pressure sensors and GPS data [8].

$$\hat{E} = \dot{a}_{c,approx}() \cdot \cos(\hat{a}_{c,approx}()); \quad \hat{N} = \dot{a}_{c,approx}() \cdot \sin(\hat{a}_{c,approx}()) \quad (15)$$

VI. EXAMPLES

Two routes were simulated on 80 long intervals with various order simplified models in the MPC algorithm. All tests were used fixed (approximately 30) iteration steps for both the trajectory generation and the MPC control optimization part. The obtained input was simulated on the high order \mathcal{H}_∞ model of the aircraft affected with wind. The constraints included were:

$$\max |a_{c,long}| = 4.5 \text{ } ^\circ; \quad \max |a_{c,lat}| = 90 \text{ } ^\circ \\ 50 \leq |a_{c}| \leq 300; \quad \max |a_{c}| \leq 36^\circ$$

The first trajectory was the following:

$$\dot{c}_{md}() = \begin{cases} 150 & \text{if } 0 \leq t \leq 30, \\ 130 & \text{if } 30 \leq t \leq 80, \end{cases} \\ c_{md}() = \begin{cases} 0^\circ & \text{if } 0 \leq t \leq 10, \\ -9^\circ & \text{if } 10 \leq t \leq 20, \\ 13.5^\circ & \text{if } 20 \leq t \leq 30, \\ -13.5^\circ & \text{if } 30 \leq t \leq 50, \\ 9^\circ & \text{if } 50 \leq t \leq 80. \end{cases}$$

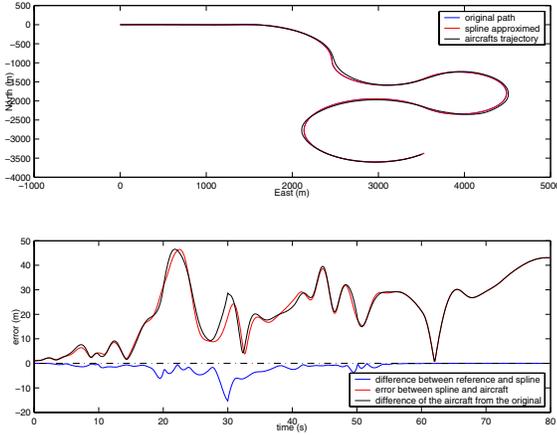


Figure 6: The UAV's path relative to the reference

It was used to verify the performance of the controller and to compare the different order simplified models used for MPC control, and to see the effect of the wind on the performance. To improve the quality of the control method weighting functions were included into the cost functions of the optimizations. This enhanced the importance of the spline functions' fitting in the optimization near to the connection points between horizons, causing improved conjunctions. The modified cost function for trajectory generation:

$$\int_{t_0}^{t_0+3n} \left\| \begin{matrix} sp(\cdot) - damp(\cdot) \\ sp(\cdot) - damp(\cdot) \end{matrix} \right\|_2^2 (\mathbf{W}(\cdot)) + 5 \cdot 2 \cdot sp(\cdot) + 6 \cdot 2 \cdot sp(\cdot) \quad (16)$$

$$\mathbf{W}(\cdot) = 2.5 \cdot 10^{-7} \cdot 3 - 1.5 \cdot 10^{-4} \cdot 2 + 2.25 \cdot 10^{-2} + 1$$

The modified cost function for MPC control:

$$\int_{t_0}^{t_0+2n} \left\| \begin{matrix} d(\cdot) - model(\cdot) \\ d(\cdot) - model(\cdot) \end{matrix} \right\|_2^2 \mathbf{W}(\cdot) + 7 \cdot 2 \cdot cmd + 8 \cdot 2 \cdot cmd \quad (17)$$

$$\mathbf{W}(\cdot) = -0.01 \cdot 2 + 0.2 + 1$$

It can be seen in Fig.6 that the maximum error is less than 50 m in case of first order model, ignoring wind. The error is defined as the distance between the aircraft's position and the desired point on the given trajectory at a certain time.

The values of the constrained variables are shown on Fig.7. These values are always lower than their maximal value, but in some cases like in the longitudinal acceleration and in the flight angle rate are far from their maximal value which affects the performance. This is due to the penalization of the actuator usage in the cost function with $\gamma = 100$ and $\delta = \gamma(180^\circ)^2$ weights. With lower cost on the performance can be improved with higher energy invest. The effect of the high refreshing time (10 s) can be seen on Fig.8, where the continuity of the constrained behavior of the system in the transition points is also demonstrated.

Fig.9 shows the effects of a random wind simulated at speeds: $N = 10.5 \pm 0.5$; $E = 15.75 \pm 0.75$. It can be seen that the first estimate for wind was false. The first measurements can be built in into the second horizon where the wind can be estimated precisely. The tracking properties are similar with the previous case with a slight difference in the error, because of the uncertainty

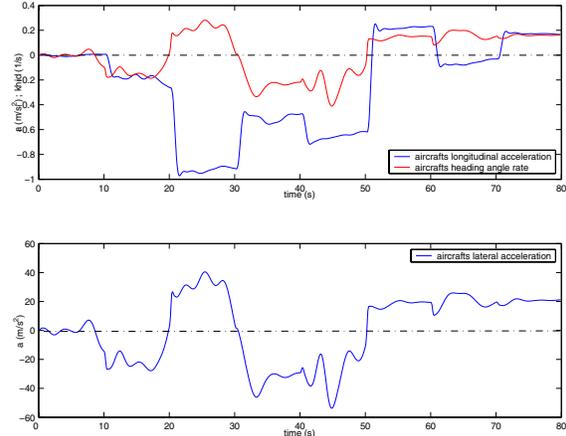


Figure 7: The values of the constraints

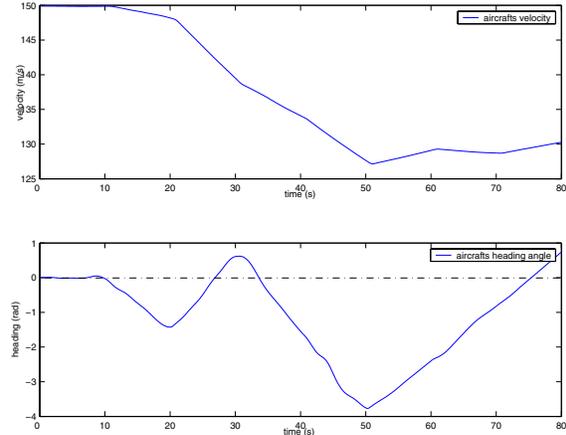


Figure 8: Velocity and heading angle of the UAV

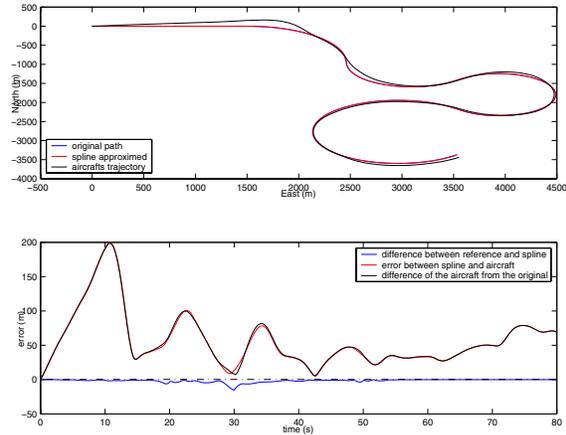


Figure 9: The UAV's path relative to the reference with wind

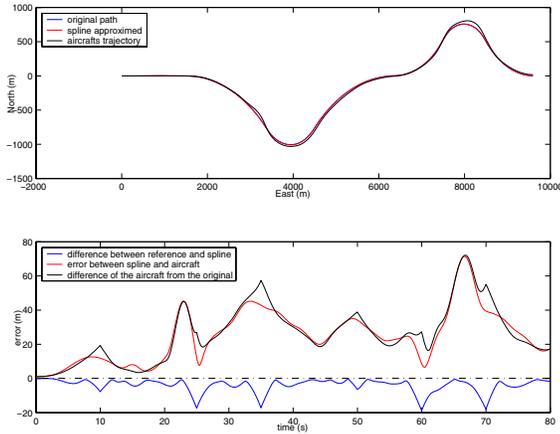


Figure 10: The UAV's path relative to the reference in the second trajectory

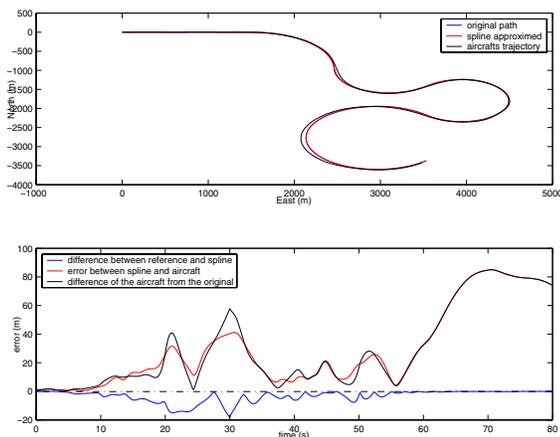


Figure 11: 80 s long tracking problem with corrected 3rd order model

of the wind. The method was verified on a different trajectory Fig.10 in which two possible obstacle avoidance manoeuvres were performed in the region of $\dot{\alpha}_c = 110 - 150$ and $|\alpha_c| = 0 - 15^\circ$. The last figure Fig.11 demonstrates the apparent opposition between higher order models and performance. It is easy to see that in the first order simplified dynamics case, where the rank of the state space is 1, all the states can be measured at the starting point of the approximating function. But when 3rd order models are used at least two states of each systems are uncertain. If $\dot{\alpha}_c(\cdot)$ and $\alpha_c(\cdot)$ are measured, the initial values of the two remaining states of each models can not be estimated. This leads to less precision. Fig.11 shows the case when the initial states for modelling the approximated aircraft position are corrected by the measured velocity and heading angle, but the other states are coming from the last uncertain simulation results not from measurements. Based on the simulation results, the higher order models have no advantages versus the first order models, unless the additional states can be predicted precisely. It is also noticeable that there is just a slight difference between the corrected and non corrected systems.

VII. CONCLUSION

The control algorithm presented provides a real-time feasible solution to trajectory tracking with constrained dynamical behavior. The present work solved many feasibility problems of a trajectory tracking problems in connection with the communication and computational time delay. It also pays attention to the boundary points and transitions between horizons. The simulation includes the high order dynamics of the aircraft not just only the properties of the control method with simplified low order dynamics. The simplified aircraft models used in the MPC prediction are also extracted from the high order aircraft model. The main issue still is the significant computational capacity needed by each optimization cycle. An effective way of increasing precision is to shorten the update time between horizons, or increase iteration steps but this would require more performance from the computer processor. Another way is to develop more efficient Sequential Quadratic Programming methods, one promising product is Tomlab's Minos [18] software, which is 10 times faster than MATLAB's `fmincon` routine. The article also presented a way of trajectory tracking control solution which can be the starting point for further development.

REFERENCES

- [1] G. Papageorgiou and D. Cofer, "Coordinated control of uninhabited air vehicles with communication and processing power limitations," in *American Institute of Aeronautics and Astronautics*, 2003.
- [2] T. Keviczky, F. Borrelli, and G. J. Balas, "A study on decentralized receding horizon control for decoupled systems," in *Proc. American Control Conf.*, Boston, Massachusetts, June 2004.
- [3] W. B. Dunbar and R. M. Murray, "Model predictive control of coordinated multi-vehicle formations," in *IEEE Conference on Decision and Control*, 2002.
- [4] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials and coordinated control of groups," in *IEEE Conf. Decision and Control*, 2001.
- [5] J. A. Fax and R. M. Murray, "Information flow and cooperative control of vehicle formations," in *IFAC World Congress*, 2002.
- [6] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in *21st Benelux Meeting on Systems and Control*, 2002.
- [7] M. B. Milam, R. Franz, and R. M. Murray, "Real-time constrained trajectory generation applied to a flight control experiment," in *IFAC World Congress*, 2002.
- [8] J. S. Dittrich, "Design and integration of an unmanned aerial vehicle navigation system," Master's thesis, Georgia Institute of Technology, 2002.
- [9] G. J. Balas, J. C. Doyle, K. Glover, A. Packard, and R. Smith, *μ -Analysis and Synthesis Toolbox User's Guide*. The Mathworks Inc., 2001.
- [10] B. Kulcsár, "Robust control design for F-16," lecture note, 2002.
- [11] J. Bokor, "State space approach to H_∞ optimal control," May 2003, lecture notes.
- [12] —, "Structured singular values μ -analysis," May 2003, manuscript.
- [13] K. E. Thomson, "F-16 uninhabited air combat vehicles," Master's thesis, Air Command and Staff College, Air University, 1998.
- [14] T. Gausz, "Flight mechanics (in hungarian)," lecture notes.
- [15] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*. New York, NY: John Wiley & Sons Inc., 1992.
- [16] C. D. Boor, *A practical guide to splines*. New York, NY: Springer-Verlag, 1978.
- [17] N. Petit, M. B. Milam, and R. M. Murray, "Inversion based constrained trajectory optimization," Control and Dynamical Systems, Caltech, Tech. Rep., 2000.
- [18] B. A. Murtagh and M. A. Saunders, "Minos 5.5 user's guide," Department of Operations Research Stanford University.
- [19] R. Bhattacharya, G. J. Balas, M. A. Kaya, and A. Packard, "Nonlinear receding horizon control of an F-16 aircraft."
- [20] G. J. Balas, "Flight control law design: An industry perspective," in *European Journal of Control*, 2003.
- [21] K. Zhou and J. Doyle, *Essentials of Robust Control*. New Jersey: Prentice Hall, 1998.
- [22] I. Szászi and P. Gáspár, "Robust servo control design using the H_∞/μ method," in *Periodica Polytechnica*, vol. 27, no. 1-2, 1999, pp. 3-16.