Journal of Guidance, Control, and Dynamics
Vol. 34, No. 6, November–December 2011

Engineering Notes

Efficient Model Predictive Control Algorithm for Aircraft

Peter W. Gibbens* and Eran D. B. Medagoda†
University of Sydney,
Sydney, New South Wales 2006, Australia

DOI: 10.2514/1.52162

I. Introduction

MODEL predictive control (MPC) has received much attention since its development over 40 years ago. Within the past 20 years, its application has steadily grown and is now in widespread use, particularly in the process industry [1] using systems with slow, predictable properties. Recent advances in predictive control have lead to its implementation onto faster dynamics systems [2,3] and unstable systems [4], providing a more reliable method of controlling models with nonlinear properties and constraints. There has also been significant research into the use of MPC in aerospace applications. These include addressing issues of robustness [5], model reduction to minimize computational demands [6] and application into airborne platforms [7].

The aim of MPC is to determine, online, an optimal control sequence that minimizes the cost of reaching a reference condition within a given prediction horizon given knowledge of the system and current state information. MPC can therefore be considered more effective than other control strategies, as it not only attempts to minimize past and current errors, but also considers the effects of future errors. A detailed summary of MPC and its associated issues is presented in [8,9].

The use of state-space models has gained increased attention for their ability to easily and accurately control multivariable processes within an MPC framework [10]. Existing formulations of predictive control using state-space models such as generalized predictive control (GPC) [11] among others rely on the discretization of the system over a fixed period to formulate a suitable discrete state-space model. This discretization period ultimately defines the interval over which the predicted samples are calculated, with the prediction horizon defined as the number of discrete steps into the future the system is to observe [12,13]. To retain model accuracy, the discretization period is kept small in order to model all possible system dynamics. However, small discretization periods, although good for model accuracy, results in poor computational performance as an increased number of samples are needed to observe future outputs for a given prediction horizon. This restricts the applications of MPC into fast dynamic systems (which require small discretization periods to capture all system dynamics), as the online computational burden is too great. If the discretization period was made larger to improve computational performance, the algorithm risks being unable to model fast model dynamics and in some cases, destabilizes the closed-loop performance. Formulations such as those discussed in [14] use continuous-time versions of GPC to control multivariable systems by predicting over specific time intervals as opposed to a fixed number of discrete steps to improve prediction accuracy computational efficiency.

This Note introduces a method of MPC using variable prediction time intervals to reduce the level of computation required without losing model accuracy, based on models formed using the eigenvalues/eigenvector characteristics of the system. This Note discusses in detail the formulation of an algebraic model predictive control (AMPC) algorithm using fewer prediction points to generate smaller prediction matrices. An analysis of the controller on a linear longitudinal aircraft model will be performed using a variety of controller configurations, assessing the effectiveness and controllability of the system using the proposed AMPC formulation compared with standard MPC formulations.

II. Efficient Model Predictive Control Algorithm

A. AMPC Formulation

Let the continuous-time nominal model be defined by

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad y(t) = Cx(t) + Du(t) \tag{1}$$

with the system linearized at time t_0 about the operating point $x(t_0)$ with the initial control set $u(t_0)$.

The evaluation of future states resulting from control inputs can be achieved by applying a convolution integral between points in time along the prediction horizon. These points herein will be referred to as *prediction points*. Formulations of MPC define the prediction horizon P as the length of time the controller is to predict future outputs and controls [14]. The convolution integral for a system evaluating future states from a uniform control input can be expressed as

$$\bar{x}(\delta t_{i+1}) = \Phi(\delta t_{i+1} - \delta t_i)\bar{x}(\delta t_i) + \int_{\delta t_i}^{\delta t_{i+1}} \Phi(\delta t_{i+1} - \tau)B\bar{u}(\delta t_i)\,\mathrm{d}\tau \tag{2}$$

where $\bar{x}(\delta t_{i+1})$ denotes the next predicted state from the current state predicted state $\bar{x}(\delta t_i)$, resulting from the control input $\bar{u}(\delta t_i)$. The predicted state can be determined by evaluating the integral between times δt_{i+1} and δt_i, where $i = 1{:}n$ (with n representing the number of prediction points) corresponds to the current time index along the prediction time vector T:

$$T = [\delta t_1 \quad \delta t_2 \quad \delta t_3 \quad \cdots \quad \delta t_n]^T \tag{3}$$

where $\delta t_n = P$, and δt_i are unevenly distributed. Note that the placement of prediction points can occur at any point along the prediction horizon and is not required to be distributed evenly as standard MPC formulations require [11,14].

The state transition matrix $\Phi(\cdot)$ is formulated over each time interval between prediction points in the prediction time vector and can be formed using a matrix exponential formulated using the eigenvalues and eigenvectors of the stability matrix A. For a given time interval between prediction points, the state transition matrix can be expressed as

$$\Phi(\delta t_{i+1} - \delta t_i) = e^{A(\delta t_{i+1} - \delta t_i)} \tag{4}$$

where

Received 25 August 2010; revision received 16 May 2011; accepted for publication 23 May 2011.

*Senior Lecturer, School of Aerospace, Mechanical and Mechatronic Engineering; peter.gibbens@sydney.edu.au.

†Graduate Research Student, School of Aerospace, Mechanical and Mechatronic Engineering; eran.medagoda@sydney.edu.au.

$$e^{A(\delta t_{i+1}-\delta t_i)} = S\begin{bmatrix} e^{\lambda_1(\delta t_{i+1}-\delta t_i)} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2(\delta t_{i+1}-\delta t_i)} & & \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & e^{\lambda_n(\delta t_{i+1}-\delta t_i)} \end{bmatrix} S^{-1} \tag{5}$$

with S defining a set of eigenvectors corresponding to the system eigenvalues λ_1 to λ_n. The eigenvalue/eigenvector method becomes sensitive when the system is ill-conditioned. In these cases, however, any prediction method will be equally sensitive, due to the inherently poor dynamic interactions of the system. By using this formulation of the matrix exponential, no approximations are needed, as a direct time-domain solution of the state transition matrix can be obtained. Any resulting inaccuracies that may stem from the state transition matrix will therefore only be subject to the precision of the model[‡] (A and B). The evaluation of the integral in Eq. (2) results in a state-space form allowing for the prediction of future states from the current operating point:

$$\bar{x}_{i+1} = \Phi(\delta t_{i+1}-\delta t_i)\bar{x}_i + \Gamma(\delta t_{i+1}-\delta t_i)\bar{u}_i \tag{6}$$

with Γ characterizing the forced-response matrix formulated as

$$\Gamma(\delta t_{i+1}-\delta t_i) = A^{-1}[\Phi((\delta t_{i+1}-\delta t_i)) - I]B \tag{7}$$

The predicted output vector $\bar{y}$ can be expressed as a function of the operating point and future control inputs in the following general solution over the prediction horizon P:

$$\bar{y}_i = C\Phi(\delta t_i)x(t_0) + \sum_{j=1}^{i} C\Phi(\delta t_n - \delta t_j)\Gamma(\delta t_j - \delta t_{j-1})\bar{u}_{j-1} + D\bar{u}_i \tag{8}$$

where $x(t_0)$ is the current operating point. The future outputs of the system for all prediction points can therefore be represented as a function of the operating point and a set of hypothesized control inputs expressed compactly as

$$Y = Fx(t_0) + GU = \begin{bmatrix} C\Phi(\delta t_1) \\ C\Phi(\delta t_2) \\ \vdots \\ C\Phi(\delta t_n) \end{bmatrix} x(t_0) + \begin{bmatrix} C\Gamma(\delta t_1) & D & 0 & \cdots & 0 \\ C\Phi(\delta t_2-\delta t_1)\Gamma(\delta t_1) & C\Gamma(\delta t_2-\delta t_1) & D & & 0 \\ \vdots & & & \ddots & \vdots \\ C\Phi(\delta t_n-\delta t_1)\Gamma(\delta t_1) & C\Phi(\delta t_n-\delta t_2)\Gamma(\delta t_2-\delta t_1) & \cdots & C\Gamma(\delta t_n-\delta t_{n-1}) & D \end{bmatrix} \begin{bmatrix} \bar{u}_0 \\ \bar{u}_1 \\ \vdots \\ \bar{u}_n \end{bmatrix} \tag{9}$$

where $x(t_0) = \bar{x}_0$ (the current operating point), and Y is a vector of changes in future outputs evaluated over the prediction horizon with n prediction points and U being a vector of hypothesized changes in input acting over the prediction horizon. The matrices F and G hold the free- and forced-response terms evaluated using the discrete convolution integral to generate outputs at each prediction point along the prediction horizon.

During the cost optimization, the control law can be obtained analytically by minimizing the cost function with respect to the control inputs. (This is for the unconstrained optimization case. When constraints are present, a quadratic programming approach is used.) The resulting optimal control set over the prediction horizon from the cost minimization gives

$$K = (G^T QG + R)^{-1} G^T Q \tag{10}$$

where Q and R are diagonal weighting matrices penalizing state and control moves, respectively.

‡The AMPC formulation is valid for cases where the state sensitivity matrix is full rank and well conditioned.

B. Enhancements to Standard MPC

In standard MPC formulations, the state transition matrix Φ is evaluated over a uniform discretization period ΔT [13]. The state transition matrix is typically evaluated via a matrix exponential, which is a function of the linear state-space system, expressed as $\Phi(\Delta T) = e^{A\Delta T}$. In standard MPC design the conventional evaluation of a matrix exponential is obtained from a Taylor series expansion normally truncated to an Kth-order approximation [14]:

$$e^{A\Delta T} = I + A\Delta T + \frac{A^2\Delta T^2}{2!} + \frac{A^3\Delta T^3}{3!} + \cdots + \frac{A^N\Delta T^K}{K!} \tag{11}$$

The omission of the higher-order terms in this expansion is an acceptable approximation if the discretization period is sufficiently small. However, if larger discretization periods are used, the higher-order terms become more significant and affect the accuracy of the result. This will invariably result in the accumulation of truncation errors when integrated for long periods, leading to reduced accuracy of predictions further into the future.

Standard MPC formulations are subject to computational constraints depending on the discretization period used and the length of the prediction horizon, particularly when long prediction horizons are used [8]. The difference between the AMPC formulation and standard MPC formulations is that the placement of prediction points can occur at any point in time along the prediction horizon and is not restricted to integer multiples of a uniform discretization period between prediction points [11,14]. This can significantly reduce the size of the prediction matrices F and G, allowing for greater computational efficiency during the evaluation of the prediction matrices and the control optimization process. Therefore this process implicitly encompasses the blocking-of-inputs methodologies [15,16] typically used to improve the efficiency of standard MPC algorithms.

The primary difference between the proposed AMPC and standard MPC formulations is that the state transition matrix Φ and forced-response matrix Γ are no longer evaluated over integer multiples of a fixed discretization periods but algebraically over unevenly distributed time intervals along the prediction horizon. The advantage of using unevenly distributed time intervals is that there is no longer a restriction on where prediction points can be placed or when predicted outputs are to be evaluated. By allowing unrestricted placement of prediction points, the predicted outputs can be evaluated at a few critical times in the future based on the nature of the systems open-loop response. It should be noted that the control law is still evaluated at each control update interval.

Figure 1 illustrates the distribution of prediction points along the prediction horizon and the subsequent hypothesized control signal. Note that the placement of prediction points can occur at any point along the prediction horizon, dictated by the prediction time vector T [Eq. (2)] and are not required to be distributed evenly as standard MPC formulations require [11,14]. The hypothesized control signal is formulated to force the predicted output to the reference within the prediction horizon. The control signal between each prediction point is held constant over that time interval as required by the convolution integral in Eq. (1).

This can significantly reduce the size of the prediction matrices F and G, allowing for greater computational efficiency during the evaluation of the prediction matrices and the control evaluation process. The size of the intervals between prediction points will have no effect on the accuracy of the predictions since the state

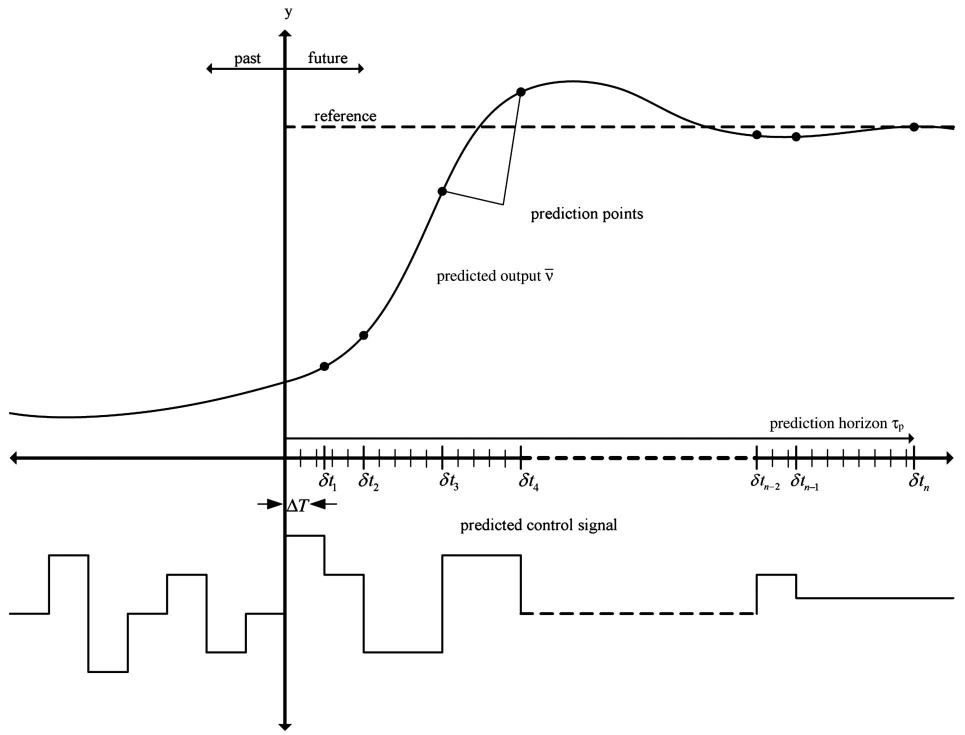

Fig. 1 Prediction-point distribution.

transition matrices are found via a direct algebraic solution using the eigenvalue/eigenvector method. This in turn improves the efficiency of the optimization process without loss in accuracy or generality.

III. Tuning Guidelines

There are various methods available in tuning the AMPC algorithm in order to obtain a desired level of system performance. Aside from modifying tracking error and control weights in Q and R, additional parameters are available such as the number and distribution of prediction points and prediction horizons.

A. Number and Distribution of Prediction Points

Prediction points are specific points along the prediction horizon that can be used to shape the predicted response of tracked outputs resulting from a hypothesized set of control actions. These points can be considered as tuning parameters, with the placement along the prediction horizon based on their influence on the system and the resulting nature of the closed-loop response.

The nature of the predicted output response is dependent on the level and duration of inputs along the prediction horizon and hence is related directly to the time intervals between prediction points as defined in the prediction time vector T. Since the outputs are computed as a convolution of impulse responses between prediction points, the level of response of the output will fundamentally depend on the magnitude and duration of the inputs. For example, consider a case where only a few prediction points are used, distributed unevenly along the prediction horizon using large time intervals between prediction points. In this case, only a few control inputs are computed to force the system to the desired reference trajectory. The resulting predicted response will be the convolution of each control input over the duration of the prediction horizon. Now consider a case where many prediction points are present, distributed unevenly with smaller time intervals between points. In this case, the overall predicted response will be the sum of the responses due to the application of all the many control inputs acting over the prediction horizon.

Comparing the two cases, the control inputs between successive prediction points are compiled over a longer period in the case where fewer points are used. Since both predicted responses are expected to converge upon the reference within the prediction horizon, the overall level of predicted control energy to the system is similar, provided output and control weight are adjusted accordingly.

B. Prediction Horizons

For most stable systems, the length of the prediction horizon depends on three key times; system time constants τ_c, settling time τ_s (generally for a type 0 system), and nonminimum phase time τ_{nmp}. The system time constant is a measure of rate of decay (or growth) of a particular system output from its initial value. It essentially defines the natural speed of the system response. Settling time is the earliest time taken for an output to settle within a certain percentage bound of its steady-state value.

The other period that needs to be considered is the nonminimum phase time.[§] If the selected prediction horizon falls within the nonminimum phase time, the predicted system output will be increasing the error between itself and the reference value. If such a case was implemented into an MPC framework, the inputs provided into the system will progressively increase the error between the reference and the current output, causing the system to be unstable. As a general guideline, the prediction horizon should lie between the nonminimum phase time and settling time to ensure that the system approaches and settles at the reference. Ideally, the prediction horizon should lie around the system time constant; however, shorter prediction horizons will generally give rise to faster transient response, though at the expense of increased control activity, while longer prediction horizons have been known to guarantee stability [17].

C. Single-Point Prediction

The ability to formulate the predictive matrices F and G using AMPC with larger and irregular time intervals between prediction points allows for the size of these prediction matrices to be significantly reduced. As a result, the computational efficiency can also be improved, with smaller matrices being used in the optimization process to formulate the optimal control set. However, having defined a set of general guidelines that define the relative length of the

[§]The nonminimum phase time defines the time where the initial response of a nonminimum phase system is in the opposite direction to desired response, i.e., the response is initially 180 deg out of phase with the final response.

prediction horizon based on a system's open-loop response, it is possible to implement a predictive controller that only relies on a single prediction point, placed at the end of the prediction horizon.[¶]

This can be achieved by formulating a state transition matrix over a time interval equal to the length of the desired prediction horizon. This will reduce the size of G (which in turn reduces the sizes of Q and R), meaning that the optimization will only need to minimize a cost function returning a single, optimal control value that will be implemented into the system. Since only the first optimized control value is ever applied to the system, the unused optimal control values that may otherwise be formulated when multiple prediction points are used no longer need to be calculated, thus reducing the execution time of the optimization. In the single-point case only a single control is evaluated across the entire prediction horizon and applied to the system *at each control update interval* ΔT.

D. Closed-Loop System and Stability

Stability of the closed-loop system can also be inferred by determining the location of the closed-loop poles. If the closed-loop poles can be shown to exist on the left hand side of the s plane, then the closed-loop system can be considered stable and can be established a priori for a linear unconstrained case.

The primary tool for stability analysis in MPC is Lyapunov theory, first shown by Keerthi and Gilbert [19]. The Lyapunov candidate function becomes the cost function of the optimization for all algorithms in this theory. In this process, it can be shown that by observing successive iterations of cost following the application of the optimal control solution at the current time step, the cost at the next time step decreases. This condition of decreasing cost over time allows for the inference of some level of stability, at least in the sense of Lyapunov, allowing for the successful use of the cost function as a Lyapunov candidate function [20]. For the proposed AMPC and standard MPC methods, the formulation of the optimal cost function used in the Lyapunov method are equivalent, with the differences stemming only from the formulation of the prediction matrices. The stability analysis presented in Muske and Rawlings [18] establishes stability in both cases.

IV. Simulation Results

To illustrate the implications of controlling a multi-input/multi-output system, the AMPC algorithm was applied to a linear longitudinal aircraft model in the following form:

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} -0.026 & 0.074 & -0.804 & -9.809 & 0 \\ -0.242 & -2.017 & 73.297 & -0.105 & -0.001 \\ 0.003 & -0.135 & -2.941 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ -0.011 & 1 & 0 & -75 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} 4.594 & 0 \\ -0.0004 & -13.735 \\ 0.0002 & -24.410 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_t \\ \delta_e \end{bmatrix} + \begin{bmatrix} -0.028 & 0.074 & 0 \\ 0.020 & -2.014 & -1.698 \\ 0.003 & -0.135 & -2.941 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_g \\ w_g \\ q_g \end{bmatrix} \tag{12}$$

with u and w representing forward and normal airspeed components in the body frame, q representing pitch rate, θ representing pitch angle and h representing altitude. The control inputs for this system are throttle δ_t and elevator δ_e. The variables u_g, w_g, and q_g represent longitudinal, normal and rotational turbulence components acting on the aircraft in unsteady atmospheric conditions. The linear model for this example was formed using a numerical perturbation process to linearize a nonlinear simulation about a nominal equilibrium trim condition. The aircraft is representative of a military training aircraft and was trimmed at an airspeed of 75 m/s at an altitude of 500 metres.

The following results demonstrate the AMPC algorithms ability in controlling multiple outputs of the linear longitudinal aircraft model. The performance of the algorithm will be assessed by analyzing the aircraft response in tracking an altitude change Δh, while regulating forward airspeed Δu, climb rate ΔV_s, and pitch rate Δq, where

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 75 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad D = [0 \quad 0] \tag{13}$$

The system weights used in this example are $Q = \mathrm{diag}(\frac{1}{0.1^2} \quad \frac{1}{0.01^2} \quad \frac{1}{0.1^2} \quad \frac{1}{0.017^2})$ and $R = \mathrm{diag}(\frac{1}{0.5^2} \quad \frac{1}{0.1745^2})$. The weighting elements have been tuned and set according to the scale of the outputs that are being tracked. This is done because the units of the tracked outputs differ between distances, speeds, and angular rates. This is also performed for control inputs, with throttle having unitary range (0 to 1) and elevator units in radians. The simulation for the following analysis is executed at a frequency of 100 Hz, corresponding to a control update interval of 0.01 s, with control calculations performed at each time step. This high frequency was used in order to capture all possible dynamics of the model and to maintain simulation fidelity.

A. Longitudinal Flight Control

Figure 2 illustrates the responses of the altitude and forward airspeed output responses in reacting to an altitude reference change. They compare the AMPC technique with standard MPC formulations. The standard MPC formulations are based on the methods discussed in [14,18] with a prediction horizon for each case set at $P = 3$ s and a control update interval of 0.01 s (100 Hz). A prediction horizon of 3 s was used based on the open-loop characteristics of the short-period phugoid mode, which induces altitude responses with time constants in the order of 2 s. In Fig. 2, it can be observed that the AMPC response is able to accurately track the desired reference with a smooth, well damped trajectory only using a single prediction point predicting 3 s into the future.

Comparing this response with those of the standard MPC formulations, it can be seen that when a few prediction points are used ($n = 6$, corresponding to a uniform model discretization period of 0.5 s),[**] the altitude tracking is adequate; however, forward airspeed regulation shows a reduction in performance with a greater divergence from zero. The performance of the standard MPC method can be improved by increasing the number of prediction points used. When $n = 20$ points are used (corresponding to a uniform model discretization period of 0.15 s) the altitude and forward airspeed responses exhibit improved performance and converge upon AMPC response, which was established as showing good performance characteristics.

Figure 3 illustrates the elevator and throttle activity of AMPC and standard MPC during the described altitude reference change. From the results it can be seen that standard MPC case using 6 prediction points generates larger control signals (particularly in throttle response). As the number of prediction points in the standard MPC

[¶]For systems with, at most, one unstable pole. For systems with more than one unstable pole, additional prediction points are needed [18].

[**]Note that the model discretization period differs from the control update interval in that the discretization period is used to formulate state transition matrix Φ, which is subsequently used to determine the prediction matrices F and G used in the control optimization process.

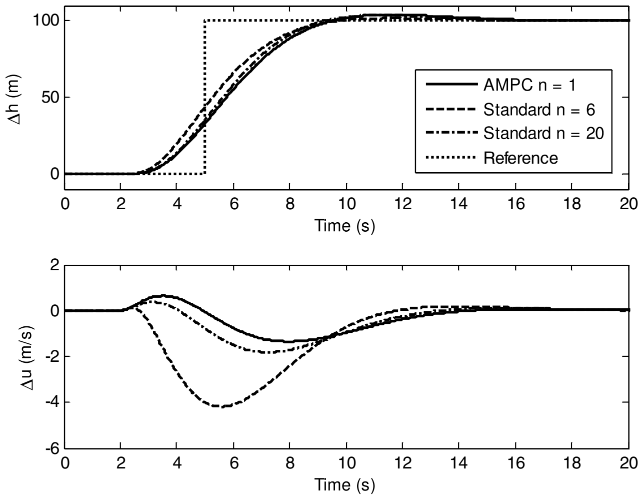

Fig. 2 Altitude and forward airspeed responses of AMPC and standard MPC.

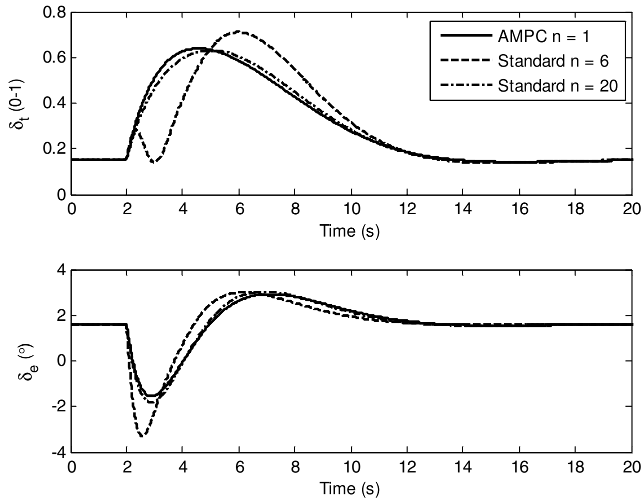

Fig. 3 Throttle and elevator activity of AMPC and standard MPC.

method increases to 20, the control activity becomes smoother and more stable, approaching the control signals of the AMPC method using only a single prediction point. This result can be attributed to a reduction in modeling errors in the prediction process when smaller discretization periods are used (Secs. II.A and II.B) and highlight the accuracy benefits of AMPC.

B. Computational Performance

There are two elements of computational performance of consequence for a linear time-invariant system; formulation of the control gains before the commencement of control operation and the recurrent control evaluation required at each control update interval. The performance of the AMPC formulation can be determined by monitoring the CPU execution time required to formulate the prediction matrices required in Eq. (9). The speed of the process is determined by the number of prediction points used, which determines the number of control moves that need to be calculated in the optimizations. Table 1 gives the computational time needed to evaluate the prediction matrices F and G for a given number of prediction points for both AMPC and standard methods. Calculations were performed in MATLAB on a computer operating an Intel Core 2 1.86 GHz (2 GB RAM). Note that for a linear system, this operation only needs to be performed once before the commencement of the recurrent control process.

Figure 2 illustrates the advantages of using AMPC over standard MPC methods. It shows that a single prediction point is more than sufficient to generate a satisfactory closed-loop response with minimal steady-state error, whereas the standard method requires a larger number of prediction points to achieve a similar level of performance.

Comparing the optimization computation times for these two cases (Table 1), AMPC when $n = 1$ requires 0.000814 s, whereas the standard method requires 0.008716 s when $n = 20$. Standard MPC therefore takes 10.7 times longer to perform the computation in order to achieve a similar level of performance as AMPC. With AMPC using a single prediction point, the size of the optimal controller K [Eq. (10)] is a 1×4 array. Hence, only seven arithmetic operations (four multiplications and three additions) are needed to determine the optimal control values in any given time interval. For the accurate standard MPC case, 20 prediction points are needed to obtain a similar level of performance to the proposed AMPC method (Fig. 2). As a result, the optimal controller becomes a 20×80 matrix, requiring 3180 arithmetic operations (1600 multiplications, 1580 additions) in order to determine the optimal control activity applied to the system. This highlights the added efficiency of AMPC in obtaining a satisfactory closed-loop response using fewer points than standard MPC methods. The recurrent computational burden during each control evaluation is significantly reduced, allowing for fast, efficient control of highly dynamic systems. This outcome leads to the following important observations;

1) The shorter computation times of AMPC offer the opportunity to shorten the control update interval if desired.

2) Despite this, the closed-loop stability is no longer tied to the control update interval, and hence the same (or better) closed-loop performance is achievable with longer time intervals, reducing computational demands further (limited only by Nyquist frequency).

The results indicate that the transient response of the system can be shaped by shifting the position of the intermediate prediction points to obtain a desired level of performance. In terms of steady-state response, the settling time can be defined by the length of the prediction horizon.

C. Prediction-Point Distribution Effects

As discussed in Secs. II.A and II.B, the difference between AMPC and standard MPC variants is the flexibility in placing prediction points at any point in time along the prediction horizon. In AMPC, the placement of prediction points is no longer defined by integer samples of a uniform control update interval since the algebraic formulation of the prediction matrices allows for variable length intervals without loss of accuracy. The intermediate predictions along the horizon can therefore be considered as tuning parameters, with their position along the horizon used to shape the nature of the closed-loop response, generalizing the input blocking method addressed in [15,16].

The following results relate to the effects of uneven distribution of prediction points along the prediction horizon (Sec. III.A.). In this example, $n = 2$ predictions points were used, with the position of the first point used as a tuning variable and the second point used to define the end of the prediction horizon. The system weights used in this example were the same as those provided in Sec. IV.A.

Figure 4 shows the response of the tracked aircraft outputs resulting from changes in prediction-point distribution with Fig. 4 showing the associated control activity. In the first case, the prediction time vector T [Eq. (3)] places the first prediction point at 1.5 s, midway along the 3 s prediction horizon. The first case was established as a baseline response for comparison to Sec. IV.A. The second case sees the first prediction point shifted closer to the current

Table 1 Time to perform prediction matrix evaluation comparing AMPC and standard methods

n	AMPC, s	Standard, s
1	0.000814	0.000562
6	0.006452	0.000964
20	0.123837	0.008716

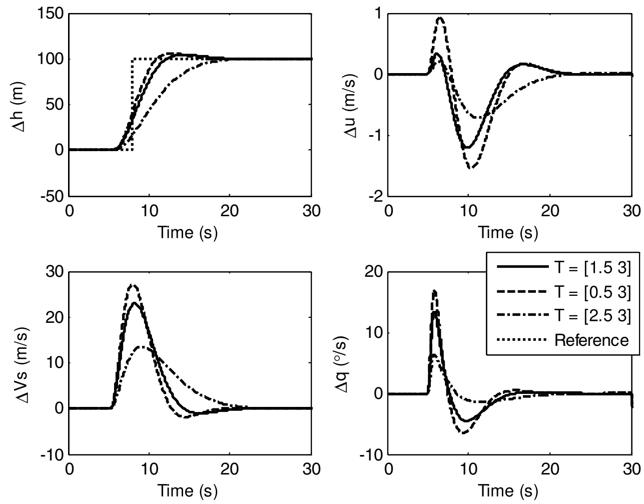

Fig. 4 Prediction-point distribution effects on output response.

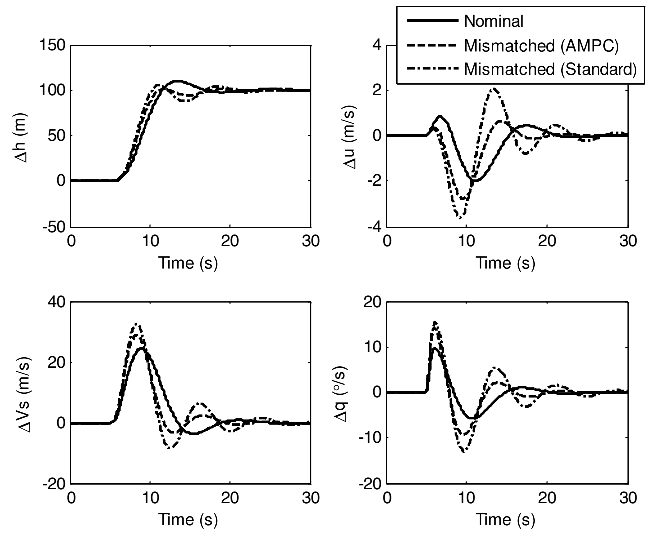

Fig. 6 AMPC nominal and mismatched model responses.

time to a value of 0.5 s, followed by a 2.5 s interval to the end of the horizon.

In this case, a reduction in the first prediction interval sees a faster transient in the altitude response in approaching the reference, which is also apparent in the control activity. The faster climb can also be observed in the climb rate (Fig. 4, bottom left) and pitch rate (Fig. 4, bottom right) responses. Shifting the first prediction point closer to the current time, however, results in inferior forward airspeed tracking (Fig. 4, top-right), with an airspeed reduction of approximately 1.5 m/s during the climb rather that 1 m/s in the baseline case. By increasing the time of the first prediction point to 2.5 s in the third case, forward airspeed regulation is improved, while the climb response becomes slower.

This effect is again apparent in the control activity, shown in Fig. 5, with smaller, slower control action generated in reaching the reference altitude. Note that the settling time of the responses in all three cases are the same, defined by the prediction horizon, which remains constant at 3 s.

D. Robustness

The issue of controller robustness is significant as the model and environmental operating conditions in reality are not known perfectly. A controller can be considered robust if the desired objectives can be achieved while remaining within the performance capabilities of the system and stability is maintained in the presence of model uncertainty.

The following results relate to the influence of a mismatch between the plant used to formulate the controller and the true system. The mismatch was introduced by modifying the value of pitch stiffness C_{M_α} between the model and true system. Pitch stiffness was chosen in this case as it is the primary aerodynamic derivative associated with the aircraft's center-of-gravity position and defines the aircraft's longitudinal stability. Variations in this parameter can significantly affect the response of the aircraft and can potentially destabilize the system.

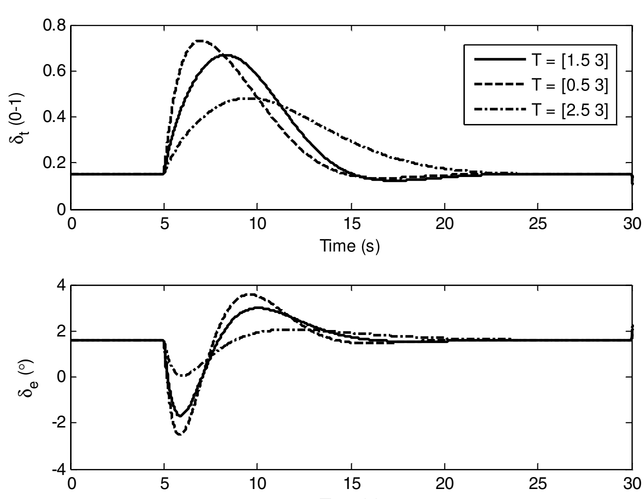

Fig. 5 Prediction-point distribution effects on control activity.

Figure 6 shows the tracked aircraft output responses resulting from a pitch stiffness mismatch. In this case, the model used by AMPC uses a pitch stiffness value equal to $C_{M_\alpha} = -1.4035$, 75% higher than the true $C_{M_\alpha} = -0.802$ value, leading to the control system overestimating the stability of the aircraft. As a result, the AMPC algorithm produces a controller that is weak in controlling the aircraft, which can be observed in the responses showing more oscillatory behavior responses across all outputs. The overall response, however, is still stable, suggesting that the AMPC algorithm has an inherent level of robustness that can accommodate large parameter variations in key derivatives.

The reason for the difference between the two cases stems from the differences in the formulation of the prediction matrices. Since AMPC uses an algebraic method, no errors are incurred due to the prediction method when formulating F and G. The standard MPC method, however, does incur prediction errors in the formulation of the prediction matrices (Sec. II.B.), which invariably introduces errors in the prediction and the formulation of the optimal control. When plant/model mismatches occur, this effect is compounded, resulting in a closed-loop response that is less stable than the AMPC equivalent. By eliminating the errors in the prediction process using an algebraic formulation, the allowable margin of uncertainty in the model can be increased, which in turn increases the overall robustness of the closed-loop system.

V. Conclusions

This Note introduces an efficient and accurate method of controlling an aircraft through model predictive control. By using the AMPC formulation based on the direct algebraic solution of the state transition matrix, the prediction points placed along the prediction horizon are no longer required to be at integer multiples of a control update interval as required by standard MPC variants. This in turn reduces the number of prediction points required to obtain an accurate predicted response and can indeed be reduced to a single point. This has been demonstrated in constrained and unconstrained situations. In general a small number of prediction points can be used to capture the weighted performance typical in standard MPC while retaining freedom of prediction-point placement and solution efficiency. The computational benefits of AMPC are also apparent as smaller prediction matrices are used during the optimization process leading to significantly reduced computation time in recurrent control evaluations. This is because fewer prediction points are required to obtain an accurate predicted response.

The efficiency and accuracy of the AMPC formulation lends itself as an appropriate method for controlling constrained and unconstrained nonlinear systems. By only requiring a few prediction points, the continual reoptimization of the controller can be achieved without hampering the speed of the control system operation.

References

[1] Qin, S. J., and Badgwell, T. A., "A Survey of Industrial Model Predictive Control Technology," *Control Engineering Practice*, Vol. 11, 2003, pp. 733–764.
doi:10.1016/S0967-0661(02)00186-7

[2] Kale, M. M., and Chipperfield, A. J., "Reconfigurable Flight Control Strategies Using Model Predictive Control," *Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, 2002, pp. 43–48.

[3] Anderson, P., and Stone, H., "Predictive Guidance and Control for a Tail-Sitting Unmanned Aerial Vehicle," *Information, Decision and Control, 2007 (IDC '07)*, 2007, pp. 148–153.

[4] Guemghar, K., Srinivasan, B., Mullhaupt, P., and Bonvin, D., "Predictive Control of Fast Unstable and Nonminimum-Phase Nonlinear Systems," *Proceedings of the 2002 American Control Conference*, Vol. 6, 2002, pp. 4764–4769.

[5] Richards, A., Breger, L., and How, J. P., "Analytical Performance Prediction For Robust Constrained Model Predictive Control," *International Journal of Control*, Vol. 79, 2006, pp. 877–894.
doi:10.1080/00207170600699740

[6] Hovland, S., Gravdahl, J. T., and Willcox, K. E., "Explicit Model Predictive Control for Large-Scale Systems via Model Reduction," *Journal of Guidance, Control, and Dynamics*, Vol. 31, 2008, pp. 918–926.
doi:10.2514/1.33079

[7] Slegers, N., Kyle, J., and Costello, M., "Nonlinear Model Predictive Control Technique for Unmanned Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, 2006, pp. 1179–1188.
doi:10.2514/1.21531

[8] Henson, M. A., "Nonlinear Model Predictive Control: Current Status and Future Directions," *Computers and Chemical Engineering*, Vol. 23, 1998, pp. 187–202.
doi:10.1016/S0098-1354(98)00260-9

[9] Garcia, C. E., Prett, D. M., and Morari, M., "Model Predictive Control: Theory and Practice; A Survey," *Automatica*, Vol. 25, 1989, pp. 335–348.
doi:10.1016/0005-1098(89)90002-2

[10] Lee, J. H., Morari, M., and Garcia, C. E., "State-Space Interpretation of Model Predictive Control," *Automatica*, Vol. 30, 1994, pp. 707–717.
doi:10.1016/0005-1098(94)90159-7

[11] Ordys, A. W., and Clarke, D. W., "A State-Space Description for GPC Controllers," *International Journal of Systems Science*, Vol. 24, 1993, p. 1727.
doi:10.1080/00207729308949590

[12] Camacho, E. F., and Bordons, C., *Model Predictive Control*, Springer, New York, 1999.

[13] Maciejowski, J. M., *Predictive Control with Constraints*, Prentice–Hall, Upper Saddle River, NJ, 2002.

[14] Demircioglu, H., and Gawthrop, P. J., "Continuous-Time Generalized Predictive Control (CGPC)," *Automatica*, Vol. 27, 1991, pp. 55–74.
doi:10.1016/0005-1098(91)90006-N

[15] Ricker, N. L., "Use of Quadratic Programming for Constrained Internal Model Control," *Industrial and Engineering Chemistry Process Design and Development*, Vol. 24, 1985, pp. 925–936.
doi:10.1021/i200031a006

[16] Ricker, N. L., "Model Predictive Control with State Estimation," *Industrial and Engineering Chemistry Research*, Vol. 29, 1990, pp. 374–382.
doi:10.1021/ie00099a013

[17] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Vol. 36, 2000, pp. 789–814.
doi:10.1016/S0005-1098(99)00214-9

[18] Muske, K. R., and Rawlings, J. B., "Model Predictive Control with Linear Models," *AIChE Journal*, Vol. 39, 1993, pp. 262–287.
doi:10.1002/aic.690390208

[19] Keerthi, S. S., and Gilbert, E. G., "Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations," *Journal of Optimization Theory and Applications*, Vol. 57, 1988, pp. 265–293.
doi:10.1007/BF00938540

[20] Muske, K. R., and Rawlings, J. B., "Linear Model Predictive Control of Unstable Processes," *Journal of Process Control*, Vol. 3, 1993., pp. 85–96.
doi:10.1016/0959-1524(93)80004-U