J Intell Robot Syst (2013) 70:27–38
DOI 10.1007/s10846-012-9724-3

An Efficient Model Predictive Control Scheme for an Unmanned Quadrotor Helicopter

M. Abdolhosseini · Y. M. Zhang · C. A. Rabbath

Received: 30 June 2012 / Accepted: 12 July 2012 / Published online: 2 August 2012

Abstract In this paper, an efficient Model Predictive Control (eMPC) algorithm deploying fewer prediction points and less computational requirement is presented in order to control a small or miniature unmanned quadrotor helicopter. A model reduction technique associated with the dynamics of an unmanned quadrotor helicopter is also put forward so as to minimize the burden of calculations in application of MPC into an airborne platform. For three-dimensional tracking control of the quadrotor helicopter, simulation results corresponding to the algebraic formulation—presented in this paper—versus the standard MPC formulation commonly found in the literature further illustrate effectiveness of this study. Unsuccessful implementation of the standard formulation on the testbed due to computational burden proves the necessity and advantages of this new approach. Eventually, to demonstrate effectiveness of the developed MPC algorithm, the suggested algebraic-based MPC framework is successfully implemented on an unmanned quadrotor helicopter testbed (known as Qball-X4) available at the Networked Autonomous Vehicles Lab (NAVL) of Concordia University for tracking control of the unmanned aerial vehicle.

M. Abdolhosseini · Y. M. Zhang (✉) · C. A. Rabbath
Concordia University, Montreal, QC, Canada
e-mail: ymzhang@encs.concordia.ca

Keywords Model Predictive Control (MPC) · Unmanned quadrotor helicopter · Experimental flight test

1 Introduction

Unmanned helicopters have become increasingly popular platforms for the study of Unmanned Aerial Vehicles (UAVs) from the control viewpoints. With the abilities such as hovering or vertical take-off and landing, unmanned helicopters substantially extend the scope of potential civilian as well as military applications such as aerial reconnaissance, border patrol, life saving, and forest surveillance or fire fighting where it is highly risky for human pilots to intervene. Successful fulfillment of such missions is closely tied with existence of autopilot control systems. For the control of a quadrotor helicopter, various control techniques have been proposed. Initially starting with linear control algorithms such as LQR control [1] or PID control [2], linear methods are proved not to have good performance for the nonlinear quadrotor system. The problem of nonlinear control design has been addressed using several methods such as feedback linearization [3], sliding mode control [4] and back-stepping control [5]. However, among those nonlinear control methods, none of them are capable of explicitly dealing with operational constraints prevalent in a control system.

"Model Predictive Control, or Model-Based Predictive Control (MPC or MBPC as it is sometimes known), is the only advanced control technique—that is, more advanced than standard PID control—to have had a significant widespread impact on industrial process control" [6]. The capability of routinely dealing with equipment and performance constraints allows for closer operation to a control system's limits, thus achieving the most profitable operation. In addition, expandability of the basic formulation to multivariable plants without any major modification, simplicity of tuning, and the straightforwardness of its underlying idea, are certainly some of the main reasons that render this controller advanced.

Predictive control was developed and used in the industry for nearly 20 years before attracting much serious attention from the academic control community. An extensive study of the literature reveals that the era of model predictive control can be broken down into three decades of developments and achievements. The first decade is characterized by the fast-growing industrial adoption of the technology, primarily in the refining and petro-chemical sectors. The second decade saw a number of significant advances in understanding the MPC from a control theoretician's viewpoint, while the third decade's main focus has been on the development of "fast MPC algorithms" [7].

Recent advances in predictive control have led to its implementation onto faster dynamic systems and unstable plants, providing solutions to bring orders of magnitude improvement in the efficiency of the online computation so that the technology can be applied to systems and plants requiring very fast sampling rates, typical examples of which are frequently appeared in the field of aerospace design and innovation [8]. In addition, there has also been research into various model reduction techniques to minimize computational demands in order to render the MPC applicable to lightweight airborne platforms [9]. This study aims to partially address the main drawback of the MPC design, namely reliance on availability of high computational power to handle burden of online repetitive calculations. To this end, a closed-loop prediction scheme will be offered for calculation of the predicted output y_p. This scheme will essentially reduce the computational load due to its linear structure. A model reduction technique will be adopted based on some simplifying assumptions so that this closed-loop linear prediction scheme can be made use of. Also, as suggested by [9] it will be benefitted from reduced number of prediction points that are not evenly placed along the prediction horizon—as required by the standard MPC variants.

The rest of the paper is organized as follows. Section 2 describes the formulation of an algebraic model predictive control. Section 3 explains mathematical modeling of the quadrotor helicopter, and suggests a scheme for its model reduction to minimize computational demands. An overview of the testbed hardware is brought in Section 4. Section 5 presents simulation results corresponding to the algebraic framework developed, versus the standard MPC formulation commonly found in the literature in order to make comparison. This is followed by successful implementation of the algebraic MPC formulation onto the quadrotor helicopter in the form of an altitude-hold controller. Finally, Section 6 draws the conclusion and outlines the future extensions of this work.

2 An Efficient Model Predictive Control Formulation

2.1 The Idea of "Predictive Control"

In what follows, the basic idea of an algebraic predictive control will be presented. For the sake of simplicity, discussion is confined to the control of a single-input single-output system. The idea and formulation set out herein can be applied to multi-input multi-output systems without loss of generality.

Though a continuous version of this MPC design approach exists as well, a discrete-time setting will be discussed and applied. This makes it easily transformable to an executable code which can be easily programmed on a microcontroller as a prospective airborne flight computer.

As mentioned, a discrete-time setting is assumed, and the current time step is represented by k. A set-point trajectory which is the ideal or expected behavior of the control system is denoted by $s(t)$. Distinct from the set-point trajectory is the reference trajectory that starts at the current output $y(k)$, and defines a second trajectory along which the plant should return to the set-point trajectory after a disturbance has made the system depart from the specific flight conditions. Therefore, the reference trajectory determines an important behavioral aspect of the closed-loop control system. Here, an exponential reference trajectory is assumed with a time constant denoted by T_{ref} specifying the speed of response as:

$$\epsilon(k+i) = e^{-iT_s/T_{\mathrm{ref}}}\epsilon(k) \tag{1}$$

where T_s is the sampling interval for both the discretization of the equations of motion representing dynamics of the quadrotor helicopter as well as the update rate of prediction.

There also exists an internal model which is employed to predict the behavior of the plant ahead of time over a prediction horizon starting at the current time. This predicted behavior is based on the assumed input trajectory $\hat{u}(k+1|k)$, $k = 0, 1, \ldots, H_p - 1$, that is to be applied over the prediction horizon but not applied yet, and the concept behind is to calculate the input that results in the best predicted performance. It has been assumed that the internal model is linear. Selection of a proper sampling interval guarantees an acceptable agreement between the nonlinear model of the quadrotor helicopter and its discretized representation. In order to calculate the input trajectory, current output measurement $y(k)$ is required. This implies that the control system is assumed to be strictly proper; that is to say, $y(k)$ depends on the values of past inputs having already applied but not on the current input $\hat{u}(k|k)$.

The elements of the input trajectory are selected such as to bring the plant output $\hat{y}(k+i)$ to the corresponding value of the reference trajectory $r(k+i)$ at specific time intervals which are not necessarily "evenly distributed". In its simplest form, the input trajectory is chosen so that the plant output coincides with the reference trajectory at the end of the prediction horizon, namely $(k + H_p)$. Further, the input trajectory may be determined such that the plant output comes to the required reference trajectory at all sampling intervals along the prediction horizon at $k+1, k+2, \ldots, k+H_p$. For the case of a single coincidence point there is several input trajectories which achieve this but based on the criteria at hand one can be chosen, for instance the one which minimizes the control effort. With this wide possible range of selection, it is in fact preferable to impose some simple structure on the input trajectory. For example, the values of the input trajectory may be allowed to vary over the first four steps of the prediction horizon, but to remain constant thereafter: $\hat{u}(k+3|k) = \hat{u}(k+4|k) = \cdots = \hat{u}(k+H_p 1|k)$. In this case there exist four parameters to choose, namely $\hat{u}(k|k)$, $\hat{u}(k+1|k)$, $\hat{u}(k+2|k)$, $\hat{u}(k+3|k)$.

In practice, however, it is quite commonplace that there are more coincidence points than parameters to choose; that is to say, more equations to be satisfied than the number of available variables, and consequently impossible to find an exact solution. This implies lack of an exact future input trajectory capable of bringing the plant output to the reference trajectory at all coincidence points. That is the reason why some sort of approximate solution is sought, looking into a specific cost function. This can be a least-squared optimization problem, namely one that minimizes the sum of the squares of the error $\sum_1^i [r(k+i|k) - \hat{y}(k+i|k)]^2$, where i corresponds to the set of coincidence points [6, 10].

2.2 The Internal Linear Model

As the name implies, the centerpiece of a model predictive controller is a mathematical model of the real plant which best represents behavioral characteristics of the control system under study. This internal model is used to predict the free response of the plant; that is the response that would be obtained at the ith coincidence point if the future input trajectory stays at the latest value having already been applied to the plant $u(k_1)$. To this end, for a state-space representation of the internal model, the current values of states or their estimations are needed. Assuming $S(i)$ to be the response of the internal model at some ith coincidence point to a unit step function, as long as a linear time-invariant system is considered, the predicted output at the ith coincidence point is:

$$\hat{y}(k+i|k) = \hat{y}_f(k+i|k) + S(i)\Delta\hat{u}(k|k) \tag{2}$$

where

$$\Delta\hat{u}(k|k) = \hat{u}(k|k) - u(k-1) \tag{3}$$

It is intended to achieve:

$$\hat{y}(k+i|k) = r(k+i|k) \tag{4}$$

Therefore, the optimal change of input is given by:

$$\Delta\hat{u}(k|k) = \frac{r(k+i|k) - \hat{y}_f(k+i|k)}{S(i)} \tag{5}$$

In a slightly complicated pattern for the input trajectory, the input is allowed to change over the first H_u steps of the prediction horizon, $\hat{u}(k|k)$, $\hat{u}(k+1|k), \ldots, \hat{u}(k+H_u-1|k)$; and remains constant thereafter, $\hat{u}(k+H_u-1|k) = \hat{u}(k+H_u|k) = \hat{u}(k+H_u+1|k) = \ldots = \hat{u}(k+H_p-1|k)$. This yields analogues results as obtained for the previous simpler input trajectory structure at the time step $k+P_i$ over the prediction horizon:

$$\begin{aligned}\hat{y}(k+P_i|k) = \hat{y}_f(k+P_i|k) &+ H(P_i)\hat{u}(k|k) \\ &+ H(P_i-1)\hat{u}(k+1|k) + \ldots \\ &+ H(P_i-H_u+2)\hat{u}(k+H_u-2|k) \\ &+ S(P_i-H_u+1)\hat{u}(k+H_u-1|k)\end{aligned} \tag{6}$$

where $H(j) = S(j) - S(j-1)$ is the unit pulse response coefficient of the system after j time steps. The reason why pulse response coefficients appear in this expression rather than step response coefficients is that each of the input values $\hat{u}(k|k), \hat{u}(k+1|k), \ldots, \hat{u}(k+H_u-2|k)$ is to be applied for only one sampling interval. Only the last one, $\hat{u}(k+H_u-1|k)$, remains unchanged until step P_i, and its effect is therefore obtained by multiplying it by the step response coefficient $S(P_i-H_u+1)$.

Since $H(j) = S(j) - S(j-1)$, Eq. 6 can be rewritten as:

$$\begin{aligned}\hat{y}(k+P_i|k) = \hat{y}_f(k+P_i|k) &+ S(P_i)\Delta\hat{u}(k|k) \\ &+ S(P_i-1)\Delta\hat{u}(k+1|k) + \ldots \\ &+ S(P_i-H_u+1)\Delta\hat{u}(k+H_u-1|k)\end{aligned} \tag{7}$$

by writing the same relation for each of the coincidence points and regrouping terms on both sides of the equation, the predicted output at the coincidence points in the matrix-vector form is:

$$Y = Y_f + \Theta\Delta U \tag{8}$$

where

$$Y = \begin{bmatrix} \hat{y}(k+P_1|k) \\ \hat{y}(k+P_2|k) \\ \vdots \\ \hat{y}(k+P_c|k) \end{bmatrix};$$

$$\Delta U = \begin{bmatrix} \Delta\hat{u}(k|k) \\ \Delta\hat{u}(k+1|k) \\ \vdots \\ \Delta\hat{u}(k+H_u-1|k) \end{bmatrix}$$

and

$$\Theta = \begin{bmatrix} S(P_1) & S(P_1-1) & \ldots & S(1) & & 0 \\ S(P_2) & S(P_2-1) & \ldots & S(1) & & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ S(P_c) & S(P_c-1) & \ldots & \ldots & & S(P_c-H_u+1) \end{bmatrix}$$

As stated earlier, it is intended to achieve Eq. 4. In the case of having more equations to be satisfied—corresponding to the number of coincidence points—than the number of available variables to be calculated, the solution of a least-squared optimization problem is sought.

Having determined a future input trajectory, only the first element of that trajectory is to be applied as the input signal to the plant and the rest are neglected. Then the whole series of events being repeated one sampling interval later; that is, *output measurement*, *prediction*, and *input trajectory determination* (Eq. 6). In the whole cycle of calculation, the prediction equations are solved to determine the input trajectory, whereas output measurement is required to obtain the reference trajectory—which is different from the setpoint trajectory—as well as the free response of the plant.

2.3 Realization of Constraints

As is the case for the standard formulation, predictive control is to be readily employed to respect constraints. Considering constraints on the inputs or outputs, the simple "linear least-squared" solution has to be replaced by a "constrained least-squared" solution. Most formulations of predictive control assume linear inequality constraints; that is because even nonlinear constraints can be approximated by one or more linear constraints. For the case of constraints in the form of linear equalities, a quadratic programing problem evolves. This can be solved very reliably and relatively quickly by means of a number of efficient, computationally inexpensive optimization software available to date.

In practice, there are usually three types of constraints existent in a control system. Limitations that should be considered for actuator ranges available for the control effort, those of possible actuator slew rates, and constraints on the controlled variables. That is equivalent to:

$$a_1 < \Delta U(k) < a_2$$

$$b_1 < U(k) < b_2$$

$$c_1 < Z(k) < c_2$$

The following formulation represents the three constraints, respectively;

$$E\begin{bmatrix}\Delta U(k)\\ 1\end{bmatrix} \le 0; \quad F\begin{bmatrix}U(k)\\ 1\end{bmatrix} \le 0; \quad G\begin{bmatrix}Y(k)\\ 1\end{bmatrix} \le 0 \tag{9}$$

where

$$U(k) = \left[\hat{u}(k|k)^T \ \hat{u}(k+1|k)^T \ \ldots \ \hat{u}(k+H_u-1|k)^T\right] \tag{10}$$

Assuming

$$G_c = \frac{1}{2}\Delta U^T \Phi \Delta U + \varphi^T \Delta U \tag{11}$$

as the cost function of a quadratic programing optimization problem with ΔU being its optimized solution, it is required to express all of the three types of constraints in terms of $\Delta U(k)$.

Suppose F has the form

$$F = \left[F_1 \ F_2 \ \ldots \ F_{H_u} \ f\right] \tag{12}$$

therefore, the second inequality of Eq. 9 can be written as:

$$\sum_{i=1}^{H_u} F_i \hat{u}(k+i-1|k) + f \le 0 \tag{13}$$

since

$$\hat{u}(k+i-1|k) = u(k-1) + \sum_{j=0}^{i-1} \Delta\hat{u}(k+j|k) \tag{14}$$

the second inequality of Eq. 9 can be written as:

$$\begin{aligned}\sum_{j=1}^{H_u} F_j \Delta\hat{u}(k|k) + \sum_{j=2}^{H_u} F_j \Delta\hat{u}(k+1|k) + \ldots \\ + F_{H_u}\Delta\hat{u}(k+H_u-1|k) + \sum_{j=1}^{H_u} F_j u(k-1) \\ + f \le 0\end{aligned} \tag{15}$$

By defining $\tilde{F}_i = \sum_{j=i}^{H_u} F_j$ and $\tilde{F} = [\tilde{F}_1, \tilde{F}_2, \ldots, \tilde{F}_{H_u}]$, then the second inequality of Eq. 9 can be written as:

$$\tilde{F}\Delta U(k) \le -\tilde{F}_1 u(k-1) - f \tag{16}$$

where the right-hand side of the inequality is a vector which is known at time k. The same methodology as discussed, can be applied to the third inequality of Eq. 9 so as to convert it into a linear inequality constraint on $\Delta U(k)$ [11].

Suppose G has the form

$$G = \left[G_1 \ G_2 \ \ldots \ G_{P_c} \ g\right] \tag{17}$$

therefore, the third inequality of Eq. 9 can be written as:

$$G_1 Y_f(k+1|k) \sum_{i=0}^{H_u-1} S(P_1 - i)\Delta\hat{u}(k+i|k)$$
$$+ G_2 Y_f(k+2|k) \sum_{i=0}^{H_u-1} S(P_2 - i)\Delta\hat{u}(k+i|k) + \ldots$$
$$+ G_{P_c} Y_f(k+P_c|k) \sum_{i=0}^{H_u-1} S(P_c - i)\Delta\hat{u}(k+i|k)$$
$$+ g \leq 0$$

By defining $\tilde{G}_j = \sum_{i=0}^{H_u-1} S(P_j - i)$ and $\tilde{G} = [\tilde{G}_1, \tilde{G}_2, \ldots, \tilde{G}_{P_c}]$, then the third inequality of Eq. 9 can be written as:

$$\tilde{G}\Delta U(k) \leq -\sum_{j=1}^{P_c} G_j Y_f(k+j|k) - g \tag{18}$$

where the right-hand side of the inequality is a vector which is known at time k. By transforming the first inequality of relations 9 into $W\Delta U(k) \leq w$ and assembling this with Eqs. 16 and 18, the problem becomes that of

$$\min\ G_c = \frac{1}{2}\Delta U^T \Phi \Delta U + \varphi^T \Delta U \tag{19}$$

subject to

$$\begin{bmatrix} \tilde{F} \\ \tilde{G} \\ W \end{bmatrix} \Delta U \leq \begin{bmatrix} -\tilde{F}_1 u(k-1) - f \\ -\sum_{j=1}^{P_c} G_j Y_f(k+j|k) - g \\ w \end{bmatrix} \tag{20}$$

3 Dynamical Equations of Quadrotor Helicopter

3.1 Nonlinear Model of a Quadrotor Helicopter

Based on the balance of forces and moments as detailed in [12], equations of motion governing dynamics of a quadrotor helicopter with respect to an earth-fixed coordinate system can be represented as follows:

$$\ddot{x} = \frac{(\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)u_1 - K_1\dot{x}}{m} \tag{21}$$

$$\ddot{y} = \frac{(\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi)u_1 - K_2\dot{y}}{m} \tag{22}$$

$$\ddot{z} = \frac{(\cos\phi \cos\theta)u_1 - K_3\dot{z}}{m} - g \tag{23}$$

$$\ddot{\phi} = \frac{u_3 l - K_4\dot{\phi}}{I_z} \tag{24}$$

$$\ddot{\theta} = \frac{u_2 l - K_5\dot{\theta}}{I_y} \tag{25}$$

$$\ddot{\psi} = \frac{u_4 c - K_6\dot{\psi}}{I_z} \tag{26}$$

where $K_i, i = 1, 2, \ldots, 6$ are the drag coefficients associated with the aerodynamic drag force, l is the distance between the center of gravity of the quadrotor and the center of each propeller, and c is the thrust-to-moment scaling factor. Note that the drag coefficients are negligible at low speeds. Also, I_x, I_y, and I_z represent the moments of inertia along x, y, and z directions. For computational convenience the inputs to the system $u_i, i = 1,\ 2,\ 3,\ 4$ are defined as:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 \\ 1 & -1 & 1 & -1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{27}$$

The actuators of the quadrotor helicopter are brushless DC motors. The relation between the PWM input applied and the thrust produced is:

$$F_i = K_{\text{motor}} \frac{w_{\text{motor}}}{s + w_{\text{motor}}} u_{PWM} \tag{28}$$

where K_{motor} is a positive gain and w_{motor} represents the actuator bandwidth. Table 1 contains the nominal values of the quadrotor helicopter's system parameters.

Table 1 System specifications

Parameter	Value	Unit
m	1.4	kg
I_x	0.03	kg.m^2
I_y	0.03	kg.m^2
I_z	0.04	kg.m^2
l	0.25	m
c	1	–
K_{motor}	120	N
w_{motor}	15	rad/s

3.2 Model Reduction to Minimize Computational Demands

As stated earlier, due to the relatively high rate of update desired for fast dynamic systems, success of predictive control in aerospace applications is highly dependent on the real-time computational power of the airborne computer. Since in almost all such applications the available onboard computational capacity is limited, partially due to weight considerations, any effort to reduce the burden of calculations is crucial to render application of the MPC to aerial systems—specifically unmanned vehicles—feasible.

To this end, it has been tried to decouple the six-degree-of-freedom equations of motion governing dynamics of the quadrotor so that the system is described by four second-order differential equations, in which the translational longitudinal displacement x is coupled with the rotational pitching motion θ, the translational lateral displacement y is coupled with the rotational rolling motion ϕ, and the translational vertical displacement along the normal axis z is treated separately and independently of the other two. That is to say:

$$\ddot{x} = \frac{u_1 \sin\theta}{m}; \quad \ddot{y} = \frac{u_1 \sin\phi}{m}; \quad \ddot{z} = \frac{u_1}{m} - g \tag{29}$$

$$\ddot{\phi} = \frac{u_3 l}{I_x}; \quad \ddot{\theta} = \frac{u_2 l}{I_y}; \quad \ddot{\psi} = \frac{u_4 c}{I_z} \tag{30}$$

This way, dimensions of the system matrices involved in the iterative calculations and that of the optimization over a single time step lasting for a fraction of a second, will be of the order of one-third or less; otherwise, direct consideration of a six-DOF motion corresponding to a quadrotor helicopter includes matrices of the order of fourteen (two corresponding to each degree of freedom plus those of DC motors). This individual treatment of the modes of motion greatly affects the execution time of onboard calculation. Also, regarding the yawing motion ψ, it has been assumed that a zero yaw angle is maintained at all times; this can be achieved by integration of a separate reaction-wheel mechanism—apart from the four DC motors—to take over control of the yawing motion.

With this new subset of equations, $\sin\theta$, $\sin\phi$, and $\frac{u_1}{m}g$ will be taken as manipulated variables or inputs of their corresponding equations (Eq. 29). That is to say, u_1 is initially calculated by means of the third equation of Eq. 29 required for steady and level flight. Then this value is substituted in both the first and the second equations of Eq. 29 as constant (over the prediction horizon), remaining $\sin\theta$ and $\sin\phi$ as the only manipulated variables. Next, the new versions of equations are discretized with a proper discretization time step, preserving dynamics of the quadrotor system. This rate can vary from one equation to the other depending on how agile the system acts along that axis.

3.3 Validation of the Simplified Decoupled Model vs. the Elaborate Coupled Model

In the previous section, based on some simplifying assumptions, a model reduction technique was used. However, the obtained decoupled model holds as long as those underlying simplifying assumptions are met; that is to say, the pitch angle as well as the roll angle are maintained within the vicinity of zero or thereabouts at all times. In other words, there is a flight envelope inside which the quadrotor is bound to stay over the course of a flight.

As stated, in order to arrive at the simplified decoupled equations of motion it is required to keep the rotational angles—roll, pitch, and yaw—as small as possible. But this is a qualitative image of the requirement. However, for the purpose of controller design this requirement should be precisely specified quantitatively as well. The controller that is designed based on the simplified model will not be functioning properly once the plant passes across or violates the boundaries of the pitch and roll angles determined to be respected for the validity of the employed model reduction technique. This is also referred to as flight envelope.

In this section, instead of conducting a set of simulations to determine the flight envelope, a single simulation is set up to reveal the validity range of the decoupled model throughout a flight. In this flight test the quadrotor is guided through a series of consecutive square trajectories of increasing sides. By increasing the sides of square trajectories while keeping the flight time over the sides constant, setpoint changes will be increased abruptly as the dimensions of the square trajectories become bigger. In order to accommodate such abrupt changes of setpoint, the controller outputs an input signal of increasing amplitude. Since the controller's outputs/the input signals to the plant are $\sin\theta$ and $\sin\phi$, soon they will reach a point beyond which the decoupled model does not conform to the coupled full order model. This point should be marked as the bottom-line of design.

As suggested in Fig. 1, if the quadrotor receives a setpoint variation of 2 meters or more along the longitudinal axis, the longitudinal controller will output a pitch angle greater than 0.2618 rad (15°) to accommodate the setpoint change. This is the maximum acceptable change of pitch angle, if the decoupled simplified equations are to be used for the purpose of design. The lateral dynamics exhibits less sensitivity to variations in the roll angle. This is illustrated in Fig. 2.

As the yaw angle does not contribute much to the cross coupling of equations of motion, its variations will be neglected in the process of controller design. In addition, it has been assumed that a separate controller is employed to maintain a zero yaw angle essentially at all times; this is pretty manageable in practice.

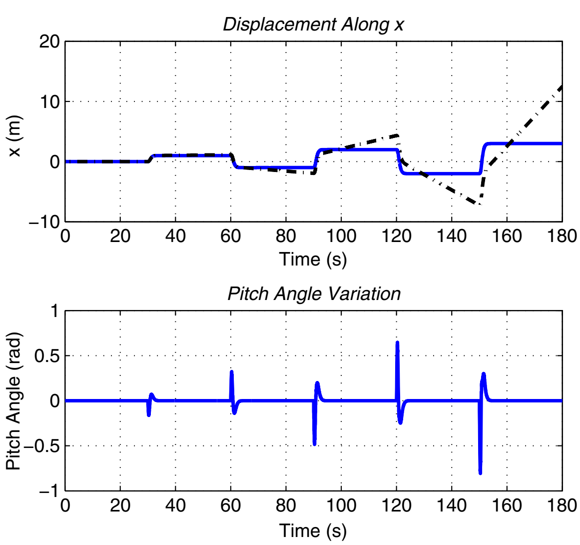

Fig. 1 Validation of the simplified decoupled model along x

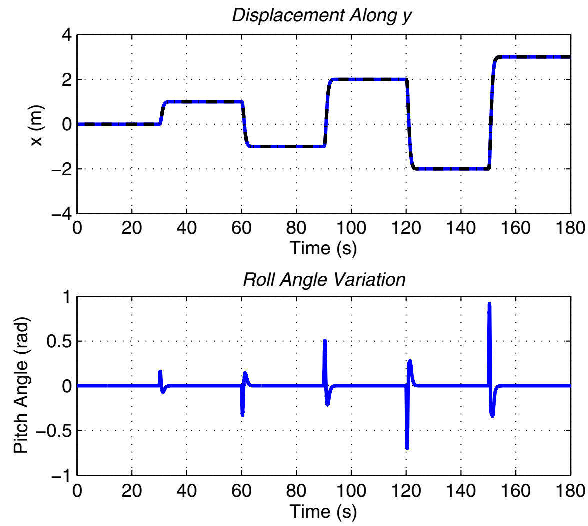

Fig. 2 Validation of the simplified decoupled model along y

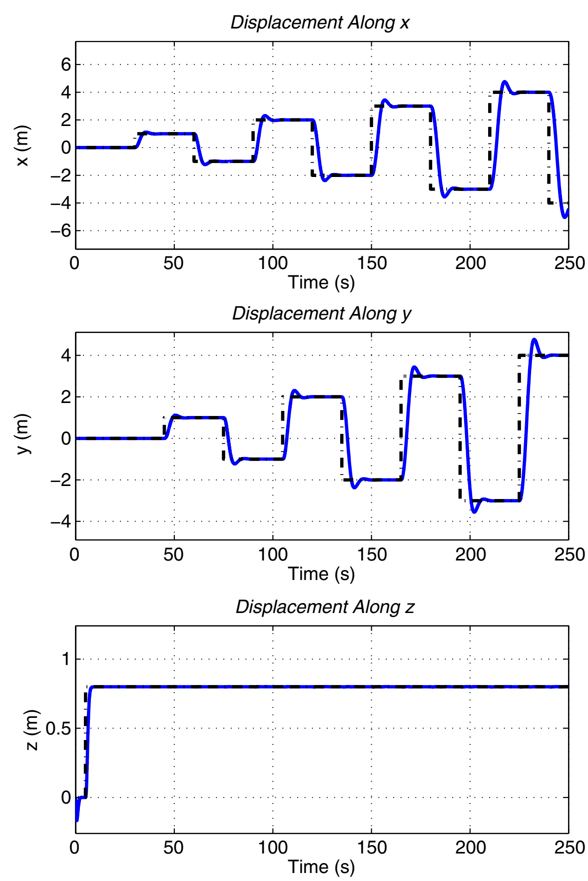

Fig. 3 Abrupt setpoint variations of great amplitude-constrained MPC

Therefore, a maximum of 2-m setpoint change of translational longitudinal or lateral motion corresponding to the 0.2618 rad (15°) change of either pitch or roll angle specify the boundaries of the pertinent flight envelope. However, this should not be interpreted as an operational restriction for the developed MPC control system, yet a shortcoming of all other controllers rather than MPC. As mentioned previously, MPC is one of the rarest control techniques that can explicitly deal with operational constraints. Therefore, once the boundaries of $\sin\theta$ and $\sin\phi$ are given to the controller as constraints on the manipulated variable or U, setpoint variations of whatever magnitude may be applied to the control system of the quadrotor. That is possible because the constrained MPC controller will never output a control signal less than −0.2618 rad (−15°) or greater than +0.2618 rad (+15°), maintaining $-15° < \theta < +15°$ and $-15° < \phi < +15°$ at all times. This is what distinguishes MPC from other controllers. That is illustrated in Fig. 3. In spite of abrupt setpoint variations of great amplitude the developed controller keeps the quadrotor on the trajectory.

4 QBALL-X4: An Unmanned Quadrotor Helicopter

The Quanser's Qball-X4, as shown in Fig. 4, is a quadrotor helicopter platform suitable for a wide variety of UAV research and development applications. This innovative rotary-wing vehicle design is propelled by four DC motors fitted with 10 inch propellers. The entire system is enclosed within a spherical protective carbon fiber cage of 68 cm. The protective cage is a crucial feature of such a system, since this unmanned aerial vehicle is designed for use in an indoor laboratory where there are typically many close-range hazards (including other vehicles).

To have on-board sensor measurements and drive the motors, the Qball-X4 utilizes Quanser's onboard avionics data acquisition card (DAQ), the HiQ, and the embedded single-board computer, Gumstix. The HiQ DAQ integrates a high-resolution Inertial Measurement Unit (IMU) and an avionics Input/Output (I/O) card designed to accommodate a wide variety of research applications. In addition, the on-board flight computer's open-architecture hardware and extensive Simulink blocksets provide users with powerful controls development tools.

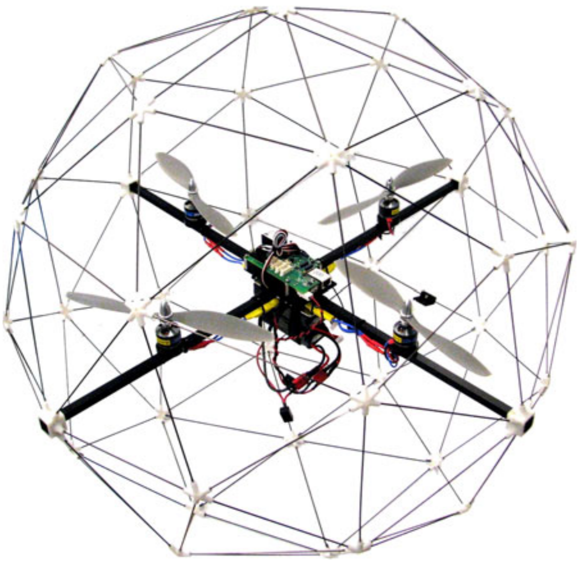

Fig. 4 The Quanser Qball-X4: a quadrotor helicopter platform

QuaRC, Quanser's real-time control software, the interface to the Qball-X4 in MATLAB/Simulink environment, allows researchers and developers to rapidly develop and test controllers on actual hardware through the MATLAB/Simulink interface. QuaRC can target the Gumstix embedded computer automatically, generating the code and executing the designed controllers on-board the vehicle. In other words, the controllers are developed in Simulink with QuaRC on the host computer. Next, these models are coded, compiled into executable codes, and eventually uploaded on the target (Gumstix) seamlessly [13].

During flights, while the controller is executing on the Gumstix, users can tune parameters in real time and observe sensor measurements from a host ground station computer (PC or laptop). System's main components include:

- Qball-X4: as explained previously;
- HiQ: QuaRC aerial vehicle data acquisition card (DAQ);

- Gumstix: The QuaRC target computer. An embedded, Linux-based system with the QuaRC runtime software installed;
- Batteries: Two 3-cell, 2500 mAh Lithium-Polymer batteries; and
- Real-Time Control Software: The QuaRC-Simulink control system development software.

5 Simulation and Experimental Results on Qball-X4

In this section, simulation and experimental results corresponding to the two MPC design approaches are illustrated: firstly based on the standard formulation employed in [14] and secondly based on the formulation detailed in this paper.

5.1 Simulation Results

In order to evaluate performance of the controllers, Qball-X4 simulation model has been given a rectangular trajectory to follow. Figures 5 and 6 demonstrate performance of the quadrotor helicopter to this input trajectory under the control of a standard MPC proposed in [14] and the one developed in this paper, respectively. As suggested by the figures, due to the simplifications made on the MPC algorithm and the internal decoupled model of the quadrotor helicopter used, three-dimensional tracking performance of the developed MPC algorithm is relatively slow in response to the reference trajectory changes, but accurate steady-state performance is still well achieved.

Even though the time response of the standard MPC algorithm is satisfactory in terms of performance specifications such as rise time, settling time, and overshoot, this controller is highly demanding in terms of the airborne computational power. The execution time of the non-real-time simulation running on a desktop computer featuring an Intel Core(TM) 2Duo CPU, 2.20GHz processor shows that it requires more than three times as much time as that required by the proposed MPC algorithm in this paper.

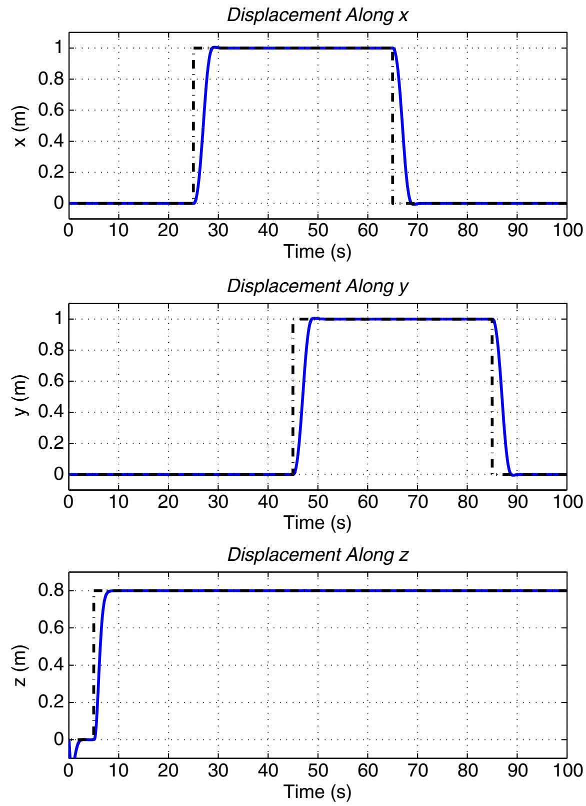

Fig. 5 System performance: standard MPC

5.2 Experimental Testing Results

Next is implementation of the controllers on the Qball-X4 to assess performance of each design approach. This is done by putting in parallel the developed controllers with the baseline controller of the Qball-X4 which is a combination of PID and LQR controllers. Having designed the controller in the environment of QuaRC for a single degree of freedom (height), each altitude-hold controller is built and uploaded to the onboard flight computer to take control of the vehicle. However, due to the quadrotor's limited airborne computational power, implementation of the standard MPC is not successful and it brings the system to a halt due to the limited computational power of the on-board Gumstix single-chip micro-computer embedded in the Qball-X4. In contrast, the proposed efficient MPC in this paper successfully controls the vehicle along a rectangular trajectory, as shown in Fig. 7. In this flight test, controller's design parameters are as specified in Table 2.

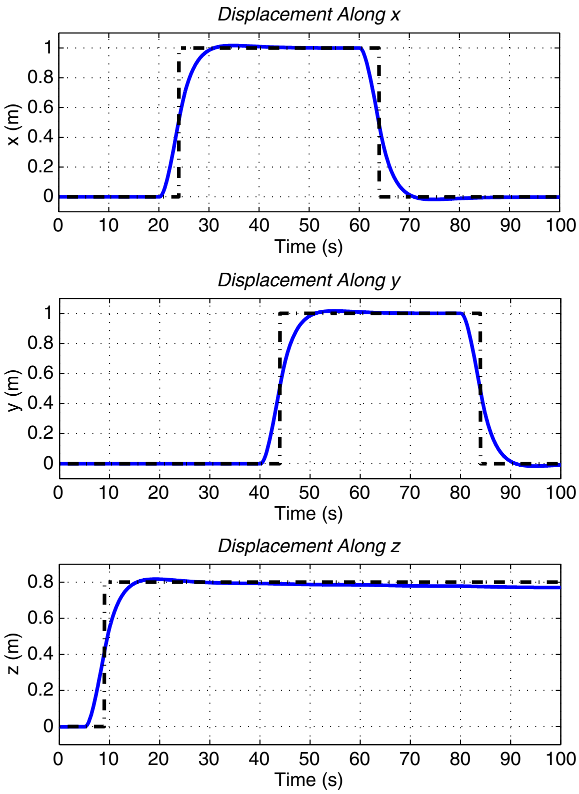

Fig. 6 System performance: developed efficient MPC

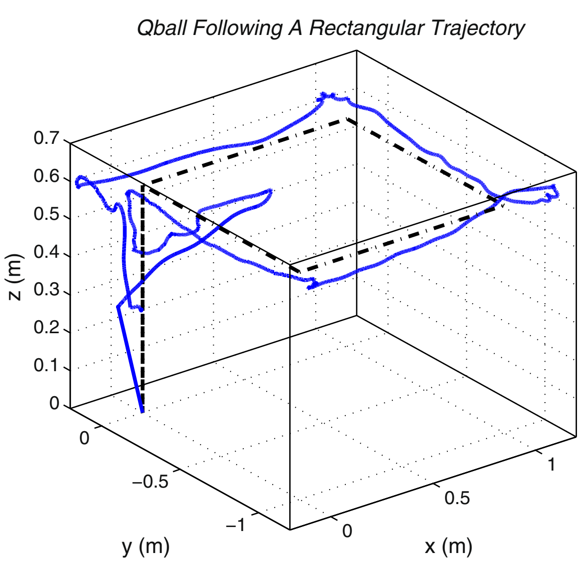

Fig. 7 Qball-X4 real-time performance developed MPC

Table 2 Controller's design parameters

Parameter	Value
T_{ref}	5
T_s	$0.01\,T_{\mathrm{ref}}$
Prediction horizon	80
Discretization rate	0.05

Comparing with the simulation result presented in Fig. 6, there are some small discrepancies in trajectory tracking due to the effects of measurement noises and disturbances added on the Qball-X4 during flights in the experimental testing environment.

5.3 Analysis and Comparison

In this paper, effort has been made to lower the burden of online calculations, in order to make the MPC applicable to unmanned aerial systems with limited airborne computational power. To this end, a linear closed-loop prediction scheme is offered for calculation of the future output y_p of the plant based on a linear internal model. This scheme has essentially reduced the computational load due to its linear structure.

To further reduce the real-time computational requirement, as suggested in [9], a reduced number of prediction points that are not evenly distributed along the prediction horizon—as required by the standard MPC variants—is adopted. Such a strategy lowers the number of coincidence points required to obtain an accurate predicted response; this can indeed be reduced to a single point. As can be seen from the simulation results, Figs. 5 and 6, conducted in the environment of MATLAB/Simulink, the overall steady-state tracking performance of the suggested MPC framework is the same as the standard MPC, although the time response associated with the linear internal model is slightly slower compared with the standard MPC.

Due to the high computational requirement needed for the standard MPC, experimental test with the Qbal-X4 testbed cannot be carried out successfully, while successful flight tests for a rectangular trajectory with the proposed MPC controller proves efficiency of the design. This is shown in Fig. 7.

6 Conclusions

In this paper an efficient Model Predictive Control (eMPC) strategy has been developed and tested on an unmanned quadrotor helicopter testbed Qball-X4 to address the main drawback of standard MPC with high computational requirement. The purpose was to make the MPC-type advanced control algorithms applicable to fast dynamics small/miniature Unmanned Aerial Vehicles (UAVs) with limited airborne computational power. To this end, a closed-loop prediction scheme is offered for calculation of the future output of the vehicle based on a linear internal model. This scheme has essentially reduced the computational load due to its linear structure. In addition, based on the assumption that during most flight maneuvers translational motions can be treated independently from each other, model reduction has been practiced to minimize calculations. Reduction of the on-board execution time was also achieved by a reduced number of prediction points that are not evenly distributed but placed along the prediction horizon. Trajectory tracking control capability has been validated through both simulation and experimental testing with the Qball-X4 UAV. Satisfactory tracking performance has been achieved with the proposed eMPC, whereas the standard MPC fails to work.

Future work will be focused on expansion of the controller so that not only altitude but also other elements of translational motion, i.e. along the longitudinal as well as the lateral axis are all collectively controlled by either a single or multiple MPC controllers.

Acknowledgment This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a Strategic Project Grant and a Discovery Project Grant.

References

1. Oner, K., Cetinsoy, E., Unel, M., Aksit, M., Kandemir, I., Gulez, K.: Dynamic model and control of a new quadrotor UAV with tilt-wing mechanism. In: World Academy of Science, Engineering and Technology (2008)
2. Erginer, B., Altug, E.: Modeling and PD control of a quadrotor VTOL vehicle. In: IEEE Intelligent Vehicles Symposium, pp. 894–899 (2007)
3. Efe, M.O.: Robust low altitude behavior control of a quadrotor rotorcraft through sliding modes. In: Mediterranean Conf. on Control and Automation (2007)
4. Bouadi, H., Bouchoucha, M., Tadjine, M.: Sliding mode control based on backstepping approach for an UAV type quadrotor. In: World Academy of Science, Engineering and Technology (2007)
5. Madani, T., Benallegue, A.: Backstepping control for a quadrotor helicopter. In: International Conference on Intelligent Robots and Systems, pp. 3255–3260 (2006)
6. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall (2000)
7. Lee, J.H.: Model predictive control: review of the three decades of development. Int. J. Control Autom. Syst. **9**(3), 415–424 (2011)
8. Kale, M.M., Chipperfield, A.J.: Reconfigurable flight control strategies using model predictive control. In: IEEE International Symposium on Intelligent Control, Taiwan, pp. 43–48 (2002)
9. Gibbens, P.W., Medagoda, E.D.B.: Efficient model predictive control for aircraft. J. Guid. Control Dyn. **34**(7), 1909–1915 (2011)
10. Murray, R.M.: Optimization-Based Control. California Institute of Technology (2010)
11. Wang, L.: Model Predictive Control System Design and Implementation Using MATLAB. Springer, London (2009)
12. Lai, L.C., Yang, C.C., Wu, C.J.: Time-optimal control of a hovering quad-rotor helicopter. J. Intell. Robot. Syst. **45**(2), 115–135 (2006)
13. Quanser's Rapid Control Prototyping (QuaRC): http://www.quanser.com/quarc
14. Liu, Y.C.: Model predictive control with integral control and constraint handling for mechatronic systems. In: International Conference on Modelling, Identification and Control, Taiwan, pp. 424–429 (2010)