

# Receding horizon control of vectored thrust flight experiment

M.B. Milam, R. Franz, J.E. Hauser and R.M. Murray

**Abstract:** The application of a constrained receding horizon control technique to stabilise an indoor vectored-thrust flight experiment, known as the Caltech ducted fan, is given. The receding horizon control problem is formulated as a constrained optimal control problem and solved in real time with an efficient, computational method that combines nonlinear control theory, B-spline basis functions, and nonlinear programming. Characteristic issues, including non-zero computational times, convergence properties, choice of horizon length and terminal cost are discussed. The study validates the applicability of real-time receding horizon control for constrained systems with fast dynamics.

## 1 Introduction

In receding horizon control (RHC), an open-loop trajectory is found by solving a finite-horizon constrained optimal control problem starting from the current state. The controls of this trajectory are then applied for a certain fraction of the horizon length, after which the process is repeated.

RHC has found successful applications in the process control industry for some time, where dynamics are relatively slow. However, the algorithm demands tremendous computational power, and can exhibit poor stability properties if not implemented properly. These difficulties have largely prevented its application to stability-critical nonlinear systems with fast dynamics. Increasingly powerful and affordable computing facilities combined with better understanding of receding horizon control's stability properties have revived interest in this area (see Mayne *et al.* [1], Findeisen *et al.* [2, 3] and the references therein for a review of recent work in this field).

To implement the RHC strategy, a constrained nonlinear optimisation problem must be solved online. Owing to the complexity of solving a nonlinear programming problem in real time, the computational delay cannot be ignored. This is particularly important in aerial vehicle applications, where the timescales of the vehicle dynamics (and the requisite control loops) are very short and comparable to the time required to solve a finite-horizon optimisation problem. The papers by Ronco *et al.* [4] and Chen *et al.* [5] examine the effects of computation in receding horizon control for some specific applications.

The application of RHC to aerial vehicles has been proposed and analysed by several researchers. Representative examples include the mixed integer linear programming

approach of Richards *et al.* [6], the LMI framework for receding horizon control of Bhattacharya *et al.* [7], and the work of Singh *et al.* [8], which provide simulation results for stabilisation of an unmanned aerial vehicle about an open loop trajectory using receding horizon control. Applications to missiles have been considered by Kim *et al.* [9] and others. The results here build on the approach developed by Jadbabaie *et al.* [10–12], in which a control Lyapunov function terminal cost is used to increase the speed of the computation while still ensuring stability.

The RHC strategy offers many benefits in this environment, such as the inherent ability to deal with constraints in the control. Examples of such a constraint commonly encountered include saturations on the actuators. However, RHC would serve little practical purpose unless stable and efficient computational techniques are developed to provide real-time solutions to the underlying constrained nonlinear optimal control problems.

The goal of the work in this paper is twofold. The first goal is to address issues of implementation with substantial computation times and fast system dynamics and the second is to provide a validation of theoretical results through implementation on an actual nonlinear experiment. A full nonlinear model of the Caltech ducted fan is used in order to test the viability of this technique on a flight platform. The results are among the first to demonstrate the use of receding horizon control for agile flight in an experimental setting using representative dynamics and computation.

## 2 Theoretical background

RHC, also known as model predictive control, makes use of online model-based optimisation to determine the current control action. In this paper, we explore the practical application of receding horizon control for operating point regulation of nonlinear systems for which a linear quadratic regulator provides the desired local system performance. Using online optimisation, we expect to enlarge the operating region significantly by exploiting knowledge of nonlinearities and control constraints.

Our system is modeled as

$$\dot{x} = f(x, u) \quad (1)$$

where the vector field  $f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  is (at least)  $C^2$  and possesses a linearly controllable equilibrium point at

© IEE, 2005

*IEE Proceedings* online no. 20059031

doi: 10.1049/ip-cta:20059031

Paper first received 12th May 2003 and in revised form 5th January 2005

M.B. Milam, and R. Franz are with the Northrop Grumman Corporation

J.E. Hauser is with the Electrical Engineering Department, University of Colorado, USA

R.M. Murray is with the Control and Dynamical Systems Department, California Institute of Technology, USA

E-mail: milam@cds.caltech.edu

the origin, i.e.  $f(0,0) = 0$  and  $(A,B) := (D_1f(0,0), D_2f(0,0))$  is controllable. Furthermore,  $f$  is affine in the control  $u$  and the control is restricted to a compact convex set  $\bar{U}$  containing the origin in its interior. We assume that  $f$  is such that (1) does not exhibit finite escape time behaviour when driven by bounded inputs. This is a reasonable assumption for the mechanical systems considered in this paper.

For the purpose of regulation, we consider the online solution of the optimal control problem

$$\begin{aligned} & \text{minimise} \quad \int_0^T q(x(\tau), u(\tau)) d\tau + V(x(T)) \\ & \text{subject to} \quad \dot{x}(t) = f(x(t), u(t)), \quad x(0) = x_0 \quad (2) \\ & \quad \quad \quad u(t) \in \bar{U} \end{aligned}$$

where  $x_0$  is the current (measured) state. The incremental cost  $q$  is chosen to be quadratic,  $q(x,u) = \|x\|_Q^2/2 + \|u\|_R^2/2$ , with  $Q = Q^T > 0$  and  $R = R^T > 0$ , so that both state and control are fully penalised (useful for nonlinear regulation) and such that the linear quadratic approximation has desirable behaviour. The terminal cost  $V$  is taken to be the linear quadratic optimal cost-to-go  $V(x) = \|x\|_P^2/2$  where  $P = P^T > 0$  is determined from the Riccati equation for the linearised system.

With this setup, we make a few observations. First, note that, for each initial condition  $x_0$ , there exists an optimal control trajectory  $u^*(\cdot; x_0)$  that achieves the minimum value  $J_T^*(x_0)$  of the cost function. Since there are no state constraints, we conclude, by the Pontryagin Maximum Principle, that each optimal control trajectory  $u^*(\cdot; x_0)$  is a continuous function of time.

Secondly, for  $x_0$  in a neighborhood of the origin, the optimal control is in fact unique and depends continuously (locally  $C^1$  or better) on the initial condition  $x_0$ . On the neighbourhood where  $u^*$  is unique and  $x_0 \mapsto u^*(\cdot; x_0)$  is continuous, we can define an instantaneous optimal state feedback  $k(x_0) := u^*(0; x_0)$ , with  $k(\cdot)$  continuous on its domain of definition and differentiable in the neighbourhood of the origin.

The optimal value function  $J_T^*(\cdot)$  is a positive definite function that is locally convex, locally smooth ( $C^2$  or better) and quadratically tangent to the optimal LQ cost  $V(\cdot)$  at  $x_0 = 0$ . Therefore the locally defined state feedback  $k(\cdot)$  is, to first order, the LQR state feedback. It follows easily that the origin of the system (1) with instantaneous optimal

feedback  $u = k(x)$  is locally exponentially stable with the optimal value function  $J_T^*(\cdot)$  acting as as Lyapunov function.

We can examine the region of operation for this instantaneous feedback law by looking at the sublevel sets of the optimal value function, defined as

$$\Gamma_r^T := \text{path connected component of } \{x \in \mathbb{R}^n : J_T^*(x) \leq r^2\} \text{ containing } 0$$

where we use the squared radius  $r^2$  (rather than  $r$ ) to reflect the fact that our incremental cost is quadratically bounded from below. For small  $r > 0$ , we may use the linearisation of  $\dot{x} = f(x, k(x))$  to conclude that the set  $\Gamma_r^T$  is positively invariant and attractive. Let  $\bar{r}$  be the largest radius such that the instantaneous optimal feedback  $k(x)$  is uniquely defined for all  $x \in \Gamma_r^T$  with  $r < \bar{r}$ . The value function  $J_T^*(\cdot)$  is  $C^1$  on the interior of  $\Gamma_{\bar{r}}^T$  (there is a potential loss of differentiability due to control saturation) so that the directional derivative,  $\dot{J}_T^*(x) := DJ_T^*(x) \cdot f(x, k(x))$ , is well defined there. Defining  $\bar{r} \leq \bar{r}$  to be that largest radius such that  $\dot{J}_T^*(x) < 0$  for all  $x \in \Gamma_r^T$  with  $r < \bar{r}$ , we see that the interior of  $\Gamma_{\bar{r}}^T$  is contained in the domain of attraction of 0 and is thus a valid region of operation for the instantaneous optimal control law.

Summarising, we have a Lyapunov function  $J^*(\cdot)$  that provides a family of sets  $\Gamma_r^T$ ,  $r < \bar{r}$ , that are positively invariant and attractive under the (possibly saturated) instantaneous optimal feedback  $u = k(x)$ . Moreover,  $J_T^*(\cdot)$  is continuous on the sets  $\Gamma_r^T$ ,  $r < \bar{r}$ . It follows that the stability and stability-like properties of the closed-loop system will persist (on a subset of the ideal operating region) under perturbations in the system vector field (including external disturbances) and in the control law.

We can view the practical application of RHC as a perturbation of the instantaneous optimal feedback law. For example, we might view the receding horizon system as a sampled data system where at each  $t_i = i\delta$  we sample the system state to obtain  $x_i = x(t_i)$ . On the interval  $[t_i, t_{i+1})$ , we apply a control  $\tilde{u}_i(t)$ ,  $t \in [t_i, t_{i+1})$  that is determined in some fashion from our online optimisation.

A standard choice is to use a zero-order hold, sampled data control  $\tilde{u}_i(t) \equiv k(x_{i-1})$ . In the case that control inputs are provided to the system at a much higher rate than the computation time  $\delta$ , however, it is useful to command the system with a portion of the computed optimal input trajectory rather than holding the input constant for

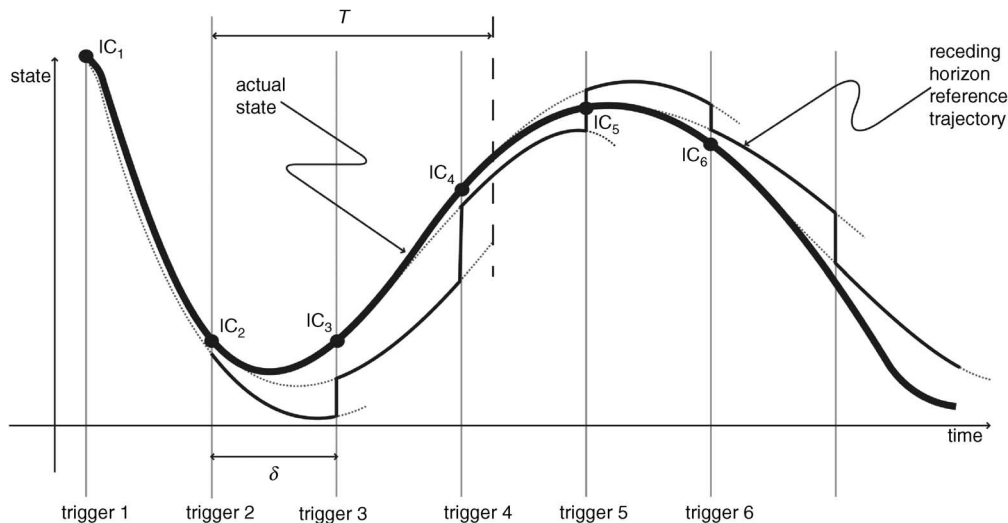
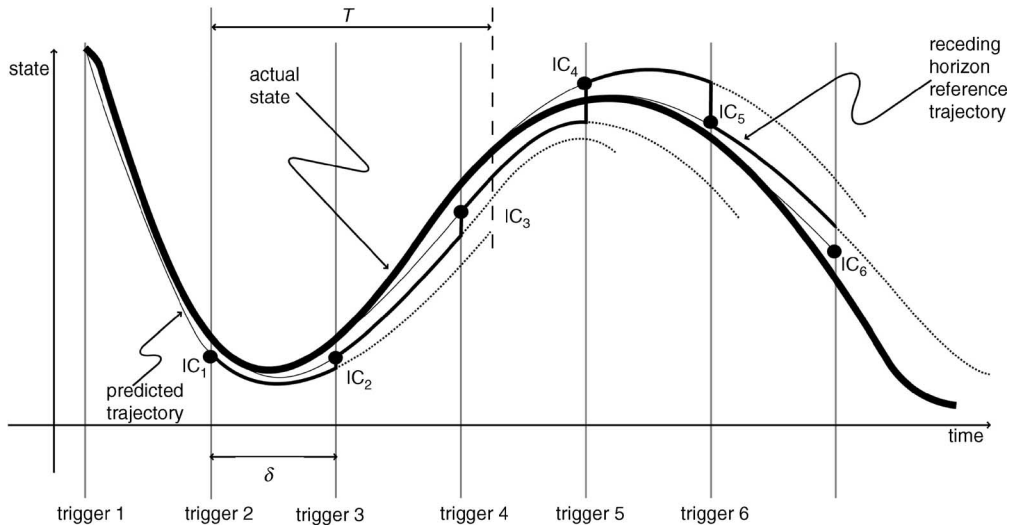


Fig. 1 Illustration of timing scheme without prediction



**Fig. 2** Illustration of timing scheme with prediction

$\delta$  seconds. The following two options for applying the optimal input trajectory are implementable if the optimisation can be performed in less than  $\delta$  seconds:

- 1 No prediction: apply  $\tilde{u}_i(t) \equiv k_{[\delta, 2\delta]}(x_i)$  over the interval  $t \in [t_{i+1}, t_{i+2}]$ .
- 2 Prediction: apply  $\tilde{u}_i(t) \equiv k_{[0, \delta]}(x_{i+1}^p)$  over the interval  $t \in [t_{i+1}, t_{i+2}]$ . Here  $x_{i+1}^p$  represents the predicted state of the system at  $t_{i+1}$  found by simulating the system model with initial condition  $x_i$  and inputs  $k_{[0, \delta]}(x_i^p)$ .

The goal with any scheme is to keep the receding horizon trajectory as close as possible to the real system trajectory in the face of significant computational delays, modelling errors and disturbances. Figures 1 and 2 show graphically the timing paradigm for options 1 and 2, respectively.

Using the Lyapunov function properties discussed above, one may easily prove

*Proposition 1:* Given  $r < \bar{r}$ , there exists a  $\delta_0 > 0$  such that, for each  $\delta \in (0, \delta_0]$ , the sampled data RHC law  $\tilde{u}_i(t)$ ,  $t \in [t_i, t_{i+1})$  with  $t_i = i\delta$ , provides exponential stability of (1) with a region of attraction that includes  $\Gamma_r^T$ .

Key in the proof of the above result is showing that  $J_T^*(\cdot)$  can be used as a discrete time Lyapunov function.

This approach to RHC will be effective provided that the optimal control problem in (2) can always be solved in  $\delta$  seconds. Since the optimal control trajectory depends continuously on the state in the region of operation, one may hope to solve the problem for  $x_{i+1}$  quickly, given the solution for  $x_i$ , by using a warm start strategy.

### 3 Technique for solving RHC problem

The resulting optimal control problem in (2) will be solved using the Nonlinear Trajectory Generation (NTG) software package developed by Milam [13].

There are three primary components to NTG. The first is to determine a parameterisation (output) such that the equations of motion can be mapped to a lower dimensional space (output space). The idea is to map dynamic constraints in (1) to algebraic ones. Once this is done, the cost and constraints in the optimal control problem in (2) are mapped to the output space. The second component of NTG is to parameterise each component of the output in terms of an appropriate B-spline polynomial. Finally, nonlinear

programming is used to solve for the coefficients of the B-splines.

The key to the approach is to map (1) to a lower dimensional output space, the idea being that it will be computationally more efficient to solve a lower dimensional problem. In most cases it is desirable to find an output  $z = z_1, \dots, z_q$  of the form

$$z = A(x, u, u^{(1)}, \dots, u^{(r)}) \quad (3)$$

such that  $(x(t), u(t))$  can be determined completely from

$$(x, u) = B(z, z^{(1)}, \dots, z^{(s)}) \quad (4)$$

where  $x^{(i)}$  denotes the  $i$ th time derivative of  $x$ . By doing this, the differential constraints in (1) are automatically satisfied and (1) is said to be differentially flat. A necessary condition for the existence of such an output can be found in Fliess *et al.* [14]. Examples of differentially flat systems can be found in Fliess *et al.* [15]. In general, finding the flat output may be very difficult even if one could prove it does exist. In the case that we cannot determine a flat output or no flat output exists, we will map the system dynamics in (1) to the lowest dimensional space possible. Therefore,  $(x(t), u(t))$  will be completely determined from

$$(x, u) = C_1(z, z^{(1)}, \dots, z^{(s_1)}) \text{ and} \\ C_2(z, z^{(1)}, \dots, z^{(s_2)}) = 0 \quad (5)$$

Once chosen, the outputs are parameterised in terms of piecewise polynomial B-spline basis functions. B-splines are chosen as basis functions for their flexibility and ease of enforcing continuity across knot points. A complete treatment of B-splines can be found in [16]. The outputs are written in terms of finite dimensional B-spline curves as

$$z_1 = \sum_{i=1}^{p_1} B_{i,k_1}(t) C_i^1 \text{ for the knot vector } \mathbf{t}_1 \\ z_2 = \sum_{i=1}^{p_2} B_{i,k_2}(t) C_i^2 \text{ for the knot vector } \mathbf{t}_2 \\ \vdots \\ z_q = \sum_{i=1}^{p_q} B_{i,k_q}(t) C_i^q \text{ for the knot vector } \mathbf{t}_q \\ \text{and } p_j = l_j(k_j - m_j) + m_j$$

where  $B_{i,k_j}(t)$  is a B-spline basis function defined in [16] for the output  $z_j$  with order  $k_j$ ,  $C_i^j$  are the coefficients of the B-spline,  $l_j$  is the number of polynomial segments and  $m_j$  is

the number of smoothness conditions at the knots. After the desired B-spline properties of the outputs have been chosen and all constraints are mapped to output space, the coefficients of the B-spline basis functions are found numerically by numerical solution of a nonlinear programming problem.

The nonlinear programming problem may be stated in the form

$$\min_{y \in R^M} F(y) \quad \text{subject to} \quad lb \leq c(y) \leq ub \quad (6)$$

where  $y = (C_1^1, \dots, C_{p_1}^1, C_1^2, \dots, C_{p_2}^2, \dots, C_1^q, \dots, C_{p_q}^q)$ ,  $M = \sum_{i=1}^q p_i$  and  $F(y)$  is the discrete approximation in output space to the objective in (2). The vector  $c(y)$  contains the input constraints in output space from (2) and any other

constraints represented in output space, as a result of not choosing the flat output. Note that the trajectory constraints will be satisfied at the collocation points chosen by the user. The lower and upper bounds for the constraints are denoted by the vectors  $lb$  and  $ub$ .

See [17] for detailed analysis of the efficiency of this approach.

#### 4 Experimental setup and mathematical model

The RHC theory and supporting numerical algorithms are validated using the Caltech ducted fan experiment. The Caltech ducted fan implements the longitudinal flight dynamics of a typical highly manoeuvrable aircraft and was carefully designed to be representative of modern flight

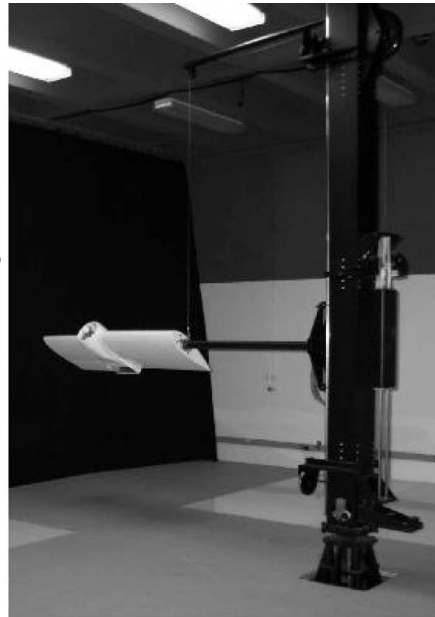
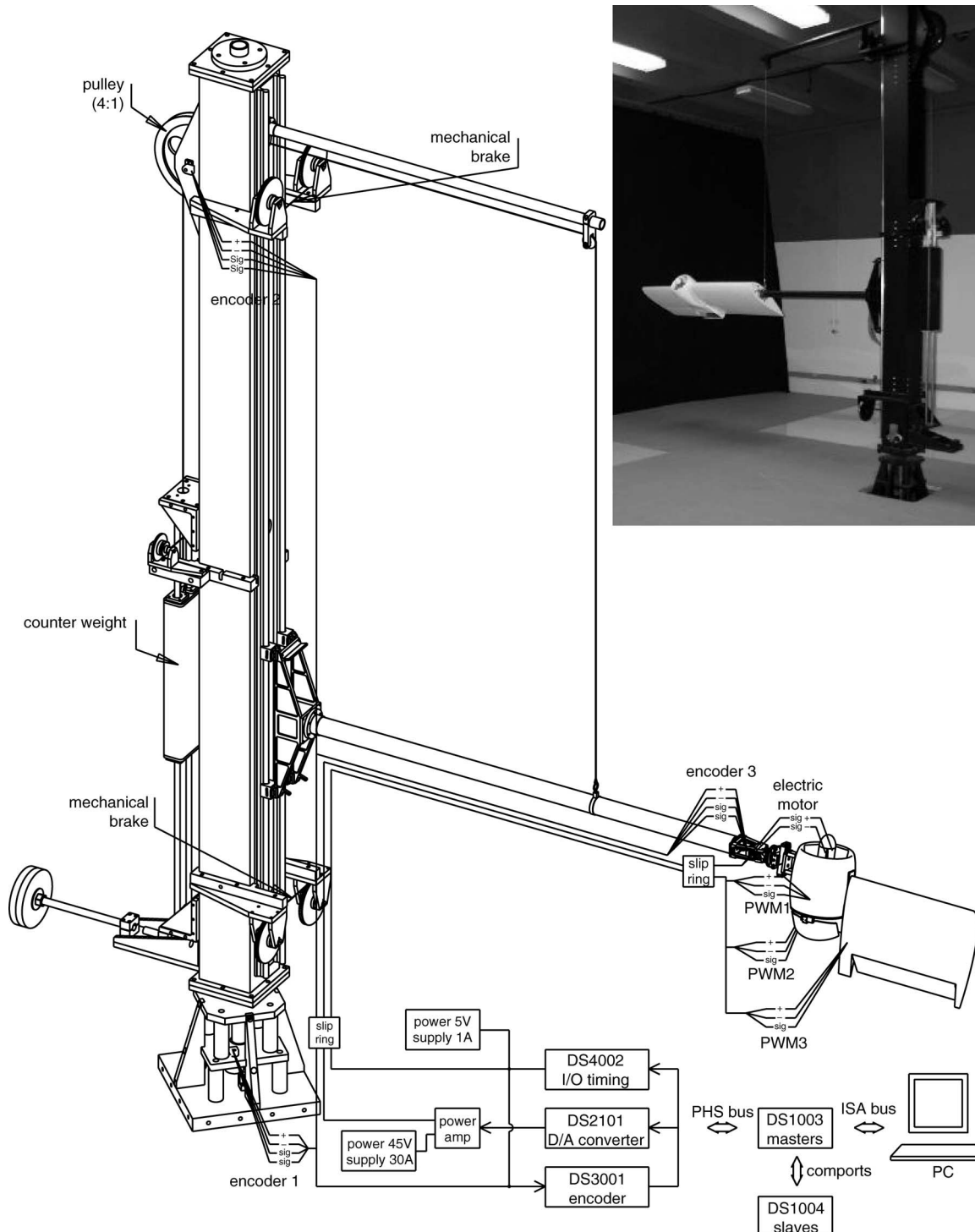


Fig. 3 The Caltech ducted fan testbed

systems [18]. Its dynamics include aerodynamic effects from lift and drag, actuation saturation, and computational delays. All these features are also present in full-scale flight systems and must be handled by flight control systems.

Figure 3 shows an overview of the Caltech ducted fan experiment. The experiment consists of a vertical stand and a horizontal boom, which holds a ducted fan and wing. This setup enables flight on a cylinder of height 2.5 m and radius 2.35 m. Because of a mass of 12.5 kg and a maximum thrust of only 13 N, a counterweight is attached to the boom via a cable and pulleys, which reduces the effective gravity to  $mg_{eff} = 7$  N. This allows the system to attain sizable vertical accelerations, while minimising the force of potential crashes. Mechanical brakes in the vertical direction are used as well to aid in crash landings. The sensors are read and the commands are sent by a multiprocessor system, comprising a D/A card, an optical encoder card, a digital IO card, two Texas Instruments C40 signal processors, two Alpha processors (500 and 600 MHz), and an ISA bus to interface with a PC. The RHC strategy used in this paper resides on the 500 MHz Alpha processor. Actuation of the ducted fan is accomplished in two ways: by controlling the

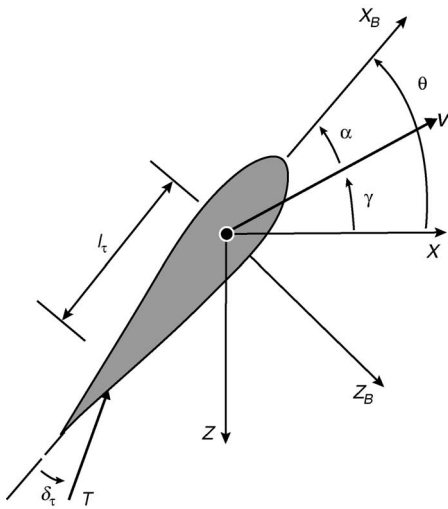


Fig. 4 Ducted fan co-ordinate frames

current to the ducted fan, and by vectoring the resulting thrust via a servo-controlled bucket. The bandwidth of the ducted fan motor is 1 Hz and the bandwidth of the bucket servos is 4 Hz.

Figure 4 shows the inertial and body co-ordinate frames used in this paper. In the inertial frame, the axes are fixed to the ground, and the  $x$  and  $z$  directions represent horizontal and vertical inertial translations. In the body frame, the  $X_B$  and  $Z_B$  axes are fixed to the vehicle.  $\theta$  represents the rotation of the ducted fan about the boom axis. All three of these variables are measured via rotary encoders, and the resulting signals are routed to the computing platform via slip rings.

The equations of motion of the experiment are given by

$$\begin{aligned} m\ddot{x} + F_{X_a} - F_{X_b} \cos \theta - F_{Z_b} \sin \theta &= 0 \\ m\ddot{z} + F_{Z_a} + F_{X_b} \sin \theta - F_{Z_b} \cos \theta &= mg_{eff} \\ J\ddot{\theta} - M_a + \frac{1}{r_s} I_p \Omega \dot{x} \cos \theta - F_{Z_b} l_\tau &= 0, \end{aligned} \quad (7)$$

where

$$F_{X_a} = D \cos \gamma + L \sin \gamma, \quad F_{Z_a} = -D \sin \gamma + L \cos \gamma$$

are the aerodynamic forces, see [18] for a complete derivation of these equations. We chose a spatial representation of the equations of motion so that we can consider both hover and forward flight modes, which couples translation and rotation of the flight dynamics.  $F_{X_b}$  and  $F_{Z_b}$  are thrust vectoring body forces. There exists an invertible nonlinear transformation between  $F_{X_b}$  and  $F_{Z_b}$  and the set thrust ( $T$ ) and the thrust vector bucket angle  $\delta_\tau$ .  $I_p = 2 \times 10^{-5} \text{ kg m}^2$  and  $\Omega = 1300 \text{ rad/s}$  are the moment of inertia and angular velocity of the ducted fan propeller, respectively.  $J = 0.25 \text{ kg m}^2$  is the moment of inertia of the ducted fan about the boom, and  $l_\tau = 0.35 \text{ m}$  is the distance from the centre of mass along the  $X_b$  axis to the effective application point of the thrust vectoring force. The angle of attack  $\alpha$  is related to the pitch angle  $\theta$  and the flight path angle  $\gamma$  by  $\alpha = \theta - \gamma$ , where the flight path angle can be derived from the spatial velocities by  $\gamma = \arctan(-\dot{z}/\dot{x})$ . The lift ( $L$ ), drag ( $D$ ) and moment ( $M$ ) are given by  $L = qSC_L(\alpha)$ ,  $D = qSC_D(\alpha)$  and  $M = \bar{c}SC_M(\alpha)$ , respectively. The dynamic pressure is given by  $q = \frac{1}{2} \rho V^2$ . The norm of

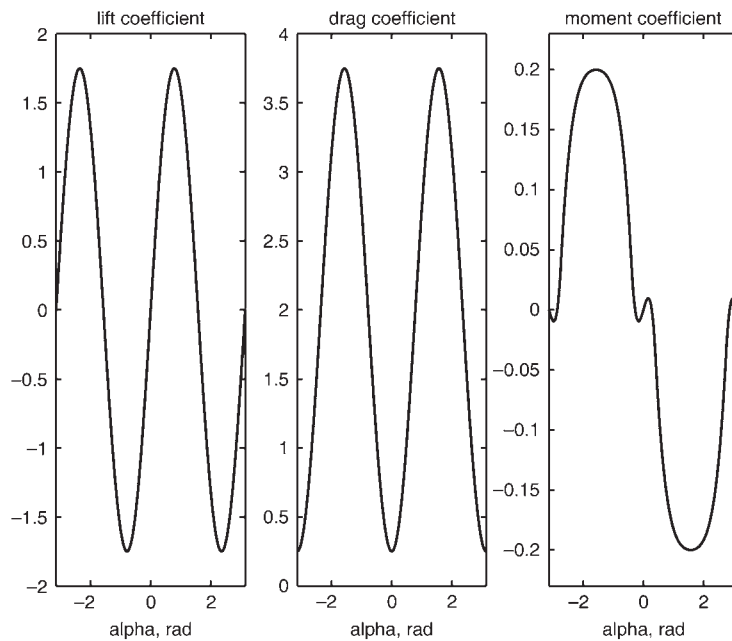


Fig. 5 B-spline curve fits to wind tunnel and flight test results for the  $C_L(\alpha)$ ,  $C_D(\alpha)$  and  $C_M(\alpha)$  aerodynamic coefficients

the spatial velocity is denoted by  $V$ , the wing chord is  $\bar{c}$ , the wing surface area is  $S$  and  $\rho$  is the atmospheric density. The coefficients of lift ( $C_L(\alpha)$ ), drag ( $C_D(\alpha)$ ) and moment ( $C_M(\alpha)$ ) were determined from a combination of wind tunnel and flight testing.

It is easily checked that the linearisation is controllable at the equilibrium operating condition.

Figure 5 shows the coefficients of lift ( $C_L(\alpha)$ ) and drag ( $C_D(\alpha)$ ) and the moment coefficient ( $C_M(\alpha)$ ).

## 5 RHC problem formulation

We first state explicitly the cost functions used in this paper, as defined in (2):

$$\begin{aligned} q(x(t), u(t)) &= \frac{1}{2}x_{err}^T(t)Qx_{err}(t) + \frac{1}{2}u_{err}^T(t)Ru_{err}(t) \\ V(x(T)) &= \frac{1}{2}x_{err}^T(T)Px_{err}(T) \\ x_{err} &\equiv x - x_{eq} = [x - x_{cmd}, z - z_{cmd}, \theta - \pi/2, \dot{x}, \dot{z}, \dot{\theta}]^T \\ u_{err} &\equiv u - u_{eq} = [F_{X_b} - mg_{eff}, F_{Z_b}]^T \\ Q &= \text{diag}[4, 3, 15, 3, 4, 0.3] \\ R &= \text{diag}[0.5, 0.5] \end{aligned} \quad (8)$$

where the equilibrium point of interest is hover:

$$x_{eq} \equiv [x_{cmd}, z_{cmd}, \pi/2, 0, 0, 0]^T, \quad u_{eq} \equiv [mg_{eff}, 0]^T$$

We choose  $Q$  and  $R$  to be the same as weights used to generate LQR gains with good performance, and  $P$  to be the corresponding solution to the algebraic Riccati equation resulting in a control Lyapunov function (CLF) terminal cost around a hover equilibrium.

The input constraints are

$$\begin{bmatrix} 0 \\ -F_{X_b}^{max}/2 \end{bmatrix} \leq \begin{bmatrix} F_{X_b} \\ F_{Z_b} \end{bmatrix} \leq \begin{bmatrix} F_{X_b}^{max} \\ F_{X_b}^{max}/2 \end{bmatrix} \quad (9)$$

where  $F_{X_b}^{max}$  is 13 N and  $mg_{eff}$  is 7.0 N.

## 6 NTG setup

By using NTG, we can sufficiently reduce the dimension of the nonlinear programming problem to make real-time computation possible.

For our system we will choose as outputs  $z_1 = x(t)$ ,  $z_2 = z(t)$ ,  $z_3 = \theta(t)$  in solving the problem posed in (2). Given these outputs, their derivatives and the control trajectory can be computed easily.

Note that the system in (7) is not necessarily differentially flat. An offline comparison of computation times and probability of convergence from a random initial guess for several different outputs motivated our parameterisation. By choosing this parameterisation, one equality constraint will need to be satisfied over the entire trajectory. In the case of only forward flight, it would be possible to choose a parameterisation that contains no equality constraints, see [19] for a complete discussion of this topic. In general, it is best to find a parameterisation that eliminates all equality constraints.

The optimal control problem is set up in NTG code by parameterising the three position states ( $x, z, \theta$ ), each with eight B-spline coefficients. Over the receding horizon time intervals, 21 collocation points were used with horizon lengths of 2.0 s. Collocation points specify the locations in

time where the differential equations and any constraints must be satisfied, up to some tolerance.

## 7 Results

### 7.1 Timing method selection

In proposition 1, we proposed that there exists a sampling rate  $\delta_0$  such that sampling rates less than  $\delta_0$  will provide a stable neighbourhood of the origin. We investigated through simulation and experimental testing the  $\delta_0$  for the ducted fan with different timing methods and horizon lengths. Table 1 shows results of identifying the highest acceptable  $\delta$  for different combinations of timing mode and horizon length. The simulation allows us to explore many different configurations without fear of damaging the hardware. The test used for these results was a 20 m step in  $x$ , a fairly demanding request, which puts the fan into a forward flight state to test out the full features of the model. We were unable to design a gain-scheduled LQR controller that could perform this manoeuvre in an acceptable fashion. Acceptable results are defined as stable and with few qualitative differences from the best results. Next, Table 2 shows horizon lengths and timing methods that were acceptable on the real experiment. One difference from these runs for the simulation was that on the real experiment we used a smaller step of 5 m in  $x$  in order to prevent damage to the apparatus. Another difference from the simulation is that the fixed period chosen is only a lower bound on the actual period. The majority of calculations remain below the fixed period in all the runs, but there are still some that exceed the value owing to limited computing power. The prediction timing method produced larger run times on average, and appeared more sensitive to the model used in NTG. Note that 0.0 s refers to the 'fast as possible' scheme. This scheme allows a varying  $\delta$ , so that, whenever a computation completes, the next computation is started immediately. The no prediction timing method with a 2 s horizon running in 'fast as possible' mode was chosen for the example test cases shown in this paper.

### 7.2 Further considerations

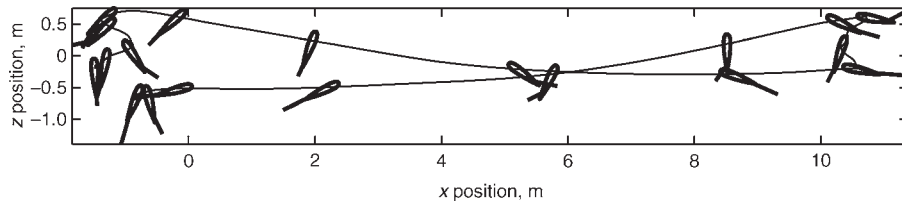
Another consideration involves non-convergent trajectory computations. Unfortunately, receding horizon trajectory computations are not guaranteed to converge. For the warm

**Table 1: Maximum acceptable  $\delta$  as determined in simulation**

Horizon	Predict	No predict
1.0 s	0.4 s	0.15 s
1.5 s	0.5 s	0.2 s
2.0 s	0.65 s	0.3 s
2.5 s	0.6 s	0.4 s
3.0 s	0.5 s	0.4 s

**Table 2: Maximum acceptable  $\delta$  as determined on the real experiment**

Horizon	Predict	No predict
1.0 s	0.0 s	0.2 s
2.0 s	0.1 s	0.2 s
3.0 s	0.2 s	0.2 s



**Fig. 6** Plot depicting actual attitude and position of ducted fan throughout manoeuvre

start strategy, each computation is given the last computed trajectory as an initial guess, which is sometimes not good enough. Also, some combinations of initial constraints and cost function are simply degenerate. If a computation returns signs of failure, the last good trajectory is simply continued and another computation is triggered. This will certainly fail if non-convergence happens frequently or repeatedly, as it has the effect of greatly increasing the sample time. In practice, this has not been a problem.

A characteristic of the spline representation used to solve the optimal control problem is that, between enforcement points, the values of the states, their derivatives and the controls may not be consistent with the equations of motion for the system. Because of this, a point on the trajectory is, in general, not suitable as an initial equality constraint for successive computation. Experience showed us that some sort of effort in minimising large jumps in at least the forces is worthwhile. To deal with this, we introduce a degree of freedom on the accelerations by eliminating their initial constraints. We are most interested in minimising jumps in the controls, so we enforce an inequality constraint

$$|u_{k+1}(0) - u_k(t)| < a \quad (10)$$

for some  $a$ . If a fixed  $\delta$  is used,  $t$  is equal to  $\delta$ . If a 'fast as possible' rule is used,  $t$  is taken as an average of the past  $n$  run times. This approach is compatible with both timing schemes discussed above; graphically, control trajectories always start near the previous trajectory.

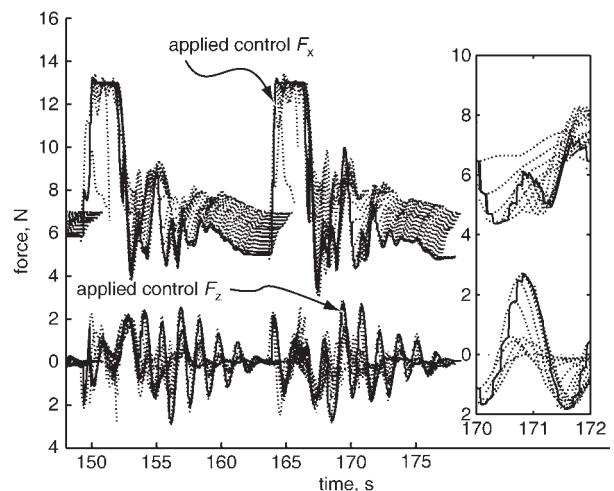
Since this constraint is applied to the finite dimensional problem, it does not affect the stability properties of the system as the constraint is not active in a small neighbourhood of the equilibrium point.

### 7.3 Ducted fan flight test results

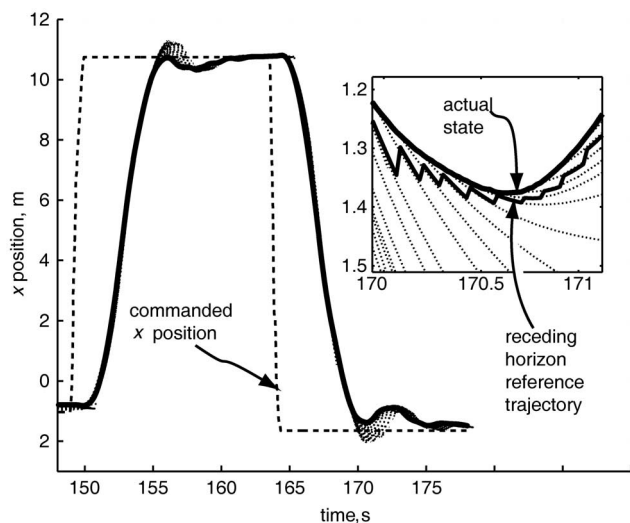
In this Section we present the result of commanding a large change in the equilibrium of the system using the cost in (8) and the constraints in (9) and (10). This aggressive command results in highly nonlinear motion of the system.

The two test cases that are investigated are aggressive manoeuvring, using a series of step commands, and operation of the ducted fan near a state constraint. The desired commands to the experiment are input with joysticks. They are set up so that the user can change in real time the  $x$  and  $z$  equilibrium positions of the experiment.

The first test case is an 11 m step command in  $x$  followed by a  $-11$  m step. Figure 6 shows an animation of the translation and rotation of the ducted fan as well as the angle of the thrust vectoring bucket and the force being applied on the system. The commanded forces are shown in Fig. 7. The RHC at  $t_k$  is denoted by a dotted line. The insert picture illustrates with a solid line the portion ( $\delta$ ) of the RHC that is being commanded to the experiment. The allowable jump in the control at  $t_{k+1}$ , given by  $|u_{k+1}(0) - u_k(\delta)|$ , was bounded by 0.25 N. Figure 8 illustrates that the system quickly responds to the step commands and then settles at the



**Fig. 7** Controls in body frame



**Fig. 8** Horizontal position of ducted fan

desired commanded location. The receding horizon reference trajectory in Fig. 8 is the predicted state resulting from the control applied in Fig. 7 applied over each  $\delta$ . Moving in and out of stall on several occasions, the attitude of the ducted fan changes significantly over the course of the run, as shown in Fig. 9. The velocity of the ducted fan is shown in Fig. 10. The RHC strategy provides very aggressive and responsive flight qualities. Each RHC trajectory in this run converged to an optimal solution. The computation times for each trajectory are shown in Fig. 11. The largest computation times occur when the system is far from the commanded equilibrium. NTG is started with a random initial guess. However, recall that the RHC solution at  $t_{k+1}$  uses the solution from  $t_k$  as the initial guess. The forces in are acceptably within the prescribed bounds, as shown in Fig. 11. The density of the collocation points determines how well the controls stay within the prescribed bounds.

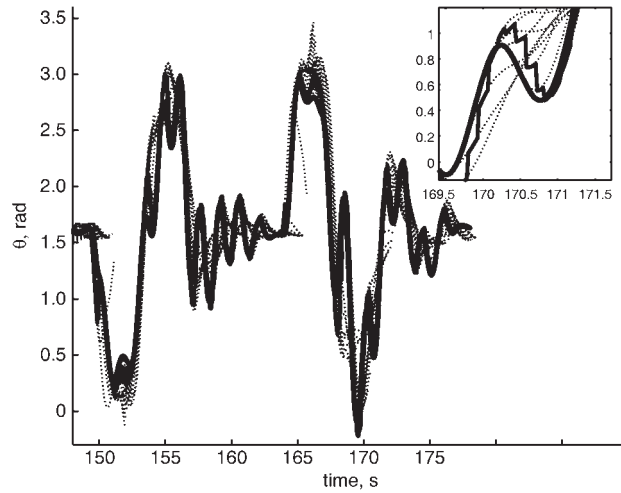


Fig. 9 Attitude of ducted fan

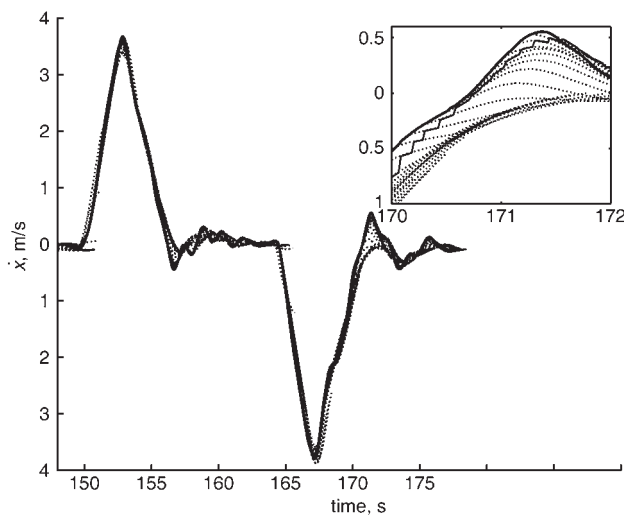


Fig. 10 Spatial horizontal velocity of ducted fan

## 8 Conclusions

The results presented in this paper demonstrate the potential of real-time RHC for constrained systems with fast dynamics. Real-time RHC represents a revolutionary alternative to the traditional linear or nonlinear controller design with many benefits.

The primary advantage of RHC over more traditional techniques is its ability to deal explicitly with input constraints. While the theoretical conditions for convergence in these cases require computation of terminal costs that satisfy these constraints (difficult to do in general), in practice RHC provides excellent stability and performance characteristics with very little tuning. In the experiments described here, the controller effectively provided global convergence to the desired equilibrium point.

In addition, RHC provides improved flexibility for trajectory generation and reconfigurability. RHC effectively integrates the inner and outer loops usually found in aircraft

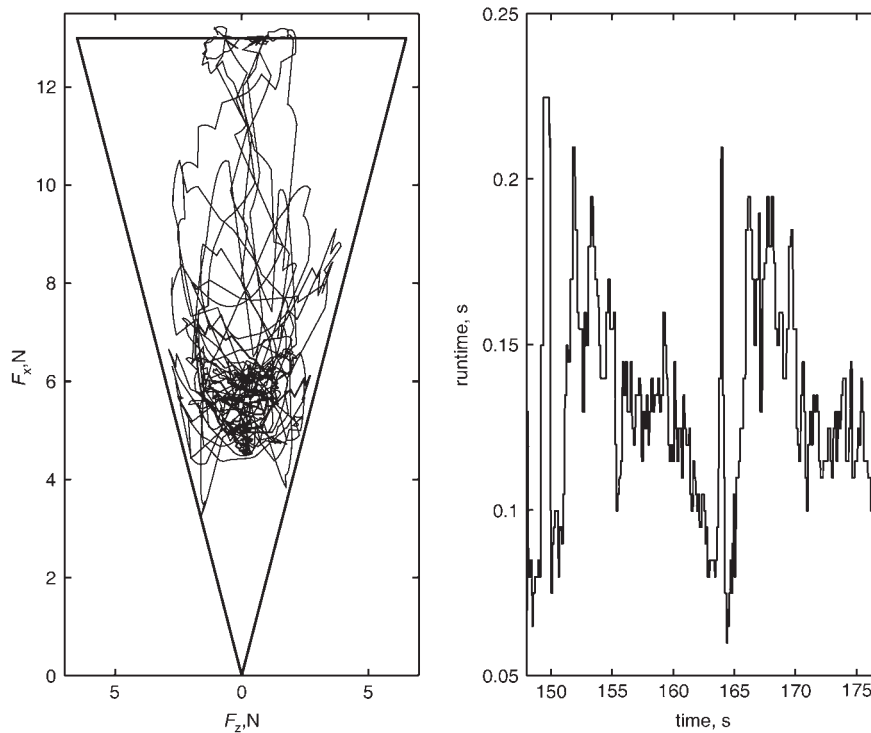


Fig. 11 Force constraints and computation times



control design, allowing the navigation function to be better integrated with stability and performance characteristics. In addition, the controller can be reconfigured online by changing the model, cost functions or constraints that are used in the receding horizon optimisation. This provides interesting opportunities for integrating vehicle management functions within the controller architecture, allowing the vehicle to reconfigure its operation in the presence of component failure or degradation.

In this paper we have addressed RHC at the vehicle dynamics level. Future research includes extending RHC for robust nonlinear optimisation at the mission level. Merits of different timing methods should also be examined through rigorous mathematical investigation and numerical simulation.

## 9 References

- 1 Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M.: 'Constrained model predictive control: stability and optimality', *Automatica*, 2000, **36**, pp. 789–814
- 2 Findeisen, R., and Allgöwer, F.: 'An introduction to nonlinear model predictive control'. 21st Benelux Meeting on Systems and Control, Veldhoven, 2002
- 3 Findeisen, R., and Allgöwer, F.: 'Computational delay in nonlinear model predictive control'. Proc. Int. Symp. Advance Control of Chemical Processes, ADCHEM'03, 2004
- 4 Ronco, E., Arsan, T., and Gawthrop, P.J.: 'Open-loop intermittent feedback control: practical continuous-time GPC', *IEE Proc., Control Theory Appl.*, 1999, **146**, pp. 426–434
- 5 Chen, W.-H., Ballance, D.J., and O'Reilly, J.: 'Model predictive control of nonlinear systems: computational burden and stability', *IEE Proc., Control Theory Appl.*, 2000, **147**, pp. 387–394
- 6 Richards, A., and How, J.: 'Model predictive control of vehicle maneuvers with guaranteed completion time and robust feasibility'. Proc. American Control Conf., 2003
- 7 Bhattacharya, R., and Balas, G.J.: 'Implementation of online control customization within the open control platform'. 'Software enabled control: information technologies for dynamical systems' (IEEE Press, 2003)
- 8 Singh, L., and Fuller, J.: 'Trajectory generation for a UAV in urban terrain, using nonlinear MPC'. Proc. American Control Conf. 2001
- 9 Kim, M.J., Kwon, W.H., Kim, Y.H., and Song, C.H.: 'Autopilot design for btt missiles using receding horizon predictive control scheme', *J. Guid. Control Dynam.*, 1997, **20**, (5), pp. 1248–1254
- 10 Jadbabaie, A., and Hauser, J.: 'Control of a thrust-vectoring flying wing: a receding horizon/LPV approach', *Int. J. Robust and Nonlinear Control*, 2002, **9**, pp. 869–896
- 11 Jadbabaie, A., and Hauser, J.: 'On the stability of unconstrained receding horizon control with a general terminal cost', *IEEE Trans. Autom. Control*, 2004, (to be published)
- 12 Jadbabaie, A., Yu, J., and Hauser, J.: 'Unconstrained receding-horizon control of nonlinear systems', *IEEE Trans. Autom. Control*, 2001, **46**, pp. 776–783
- 13 Milam, M.B.: 'Real-time optimal trajectory generation for constrained dynamical systems'. PhD Thesis, Cal. Inst. of Tech., 2003
- 14 Fliess, M., Lévine, J., Martin, P., and Rouchon, P.: 'Flatness and defect of non-linear systems: introductory theory and examples', *Int. Journal Control*, 1995, **6**, pp. 1327–1360
- 15 Fliess, M., Lévine, J., Martin, P., and Rouchon, P.: 'A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems', *IEEE Trans. Autom. Control*, 1999, **44**, (5), pp. 928–937
- 16 de Boor, C.: 'A practical guide to splines' (Springer-Verlag, 1978)
- 17 Petit, N., Milam, M.B., and Murray, R.M.: 'Inversion based constrained trajectory optimization'. 5th IFAC Symp. on Nonlinear Control Systems, 2001
- 18 Milam, M., and Murray, R.M.: 'A testbed for nonlinear flight control techniques: the Caltech ducted fan'. Proc Conf. on Control Applications, 1999
- 19 Martin, P.: 'Aircraft control using flatness'. Proc. European Control Conf. 1996, 194–199