To appear, IFAC World Congress, July 2002

MODEL PREDICTIVE CONTROL OF A THRUST-VECTORED FLIGHT CONTROL EXPERIMENT

William B. Dunbar [*,1] **Mark B. Milam** [*] **Ryan Franz** [**]
Richard M. Murray [*]

[*] *Division of Engineering and Applied Science*
Control and Dynamical Systems
California Institute of Technology
Pasadena,CA 91125

[**] *Electrical and Computer Engineering*
University of Colorado, Boulder, CO 80309

Abstract: Model predictive control (MPC) is applied to the Caltech ducted fan, a thrust-vectored flight experiment. A real-time trajectory generation software based on spline theory and sequential quadratic programming is used to implement the MPC controllers. Timing issues related to the computation and implementation of repeatedly updated optimal trajectories are discussed. Results show computational speeds greater than 10 Hz, 2.5 times that of the actuator dynamics. The MPC controllers successfully stabilize a step disturbance applied to the ducted fan and compare favorably to LQR methods.

Keywords: predictive control, model-based control, real-time, flight control, stabilization, constraints

1. INTRODUCTION

This paper is concerned with the application of model predictive control (MPC) to a high-performance flight control experiment shown in Figure 1. In MPC, the current control action is determined by solving a finite horizon open-loop optimal control problem on-line. Each optimization yields a control law that is applied to the plant until the next sampling instant. MPC is traditionally applied to plants with dynamics slow enough to permit computations between samples. It is also one of few suitable methods in applications that can impose constraints on the states and or inputs, as the constraints are directly enforced in the on-line optimal control problem. With the advent of faster modern computers, it has become possible to extend MPC to systems governed by faster dynamics that warrant this type of solution. An example of such a system is the Caltech ducted fan, a thrust-vectored flight

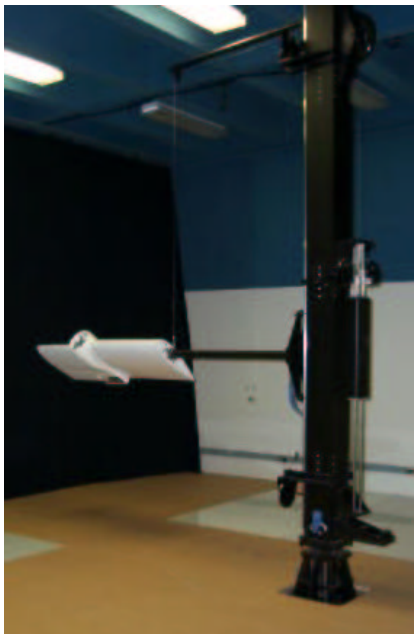

Fig. 1. Caltech ducted fan experiment

[1] Author to whom correspondence should be addressed, email: dunbar@cds.caltech.edu.

control experiment where actuation and spatial constraints are present.

Development and application of MPC originated in process control industries where plants being controlled are sufficiently slow to permit its implementation. This was motivated by the fact that the economic operating point of a typical process lies at the intersection of constraints. An overview of the evolution of commercially available MPC technology is given in (Qin and Badgwell, 1997). Most algorithms use only impulse and step response models. More recent developments in process applications employ MPC based on neural networks (Tan *et al.*, 2001) and fuzzy logic (Huang *et al.*, 2000), (Mutha *et al.*, 1997). A motivation for the fuzzy logic approach is that it can be applied to highly nonlinear systems and attempts to alleviate the computational demand of attempting to use MPC with first principles-based models.

Following the large number of successful industrial applications, academic researchers began to investigate stability of MPC theoretically. A survey of the current state of stability theory of MPC is given in (Mayne *et al.*, 2000). The paper details the various MPC problem formulations that guarantee stability, particularly when the system is modeled by a state equation and constrained. Applications of MPC to systems other than process control problems have begun to emerge over recent years ((Low and Zhuang, 2000)). Neural Network based MPC is applied to an underwater vehicle in (Kodogiannis *et al.*, 1996), robots in (Ollero *et al.*, 1994), a wind-tunnel experiment in (Scott and Pado, 2000) and a helicopter experiment in (Wan and Bogdanov, 2001). In (Singh and Fuller, 2001), MPC is used in simulation to stabilize a linearized simplified UAV helicopter model around an open-loop trajectory, while respecting state and input constraints.

The main contribution of this paper is to demonstrate an implementation of MPC on a experiment with dynamics of considerable speed, namely flight dynamics, and strict constraints on the inputs. This is possible because of modern computational power and a software package (NTG, for nonlinear trajectory generation) developed at Caltech (Milam *et al.*, 2000). Timing issues that arise from real-time MPC computations are elucidated for this system. A method for applying the generated optimal trajectories while accounting for nontrivial computational time is detailed and implemented. Experimental tests compare various MPC parameterizations and show the success of this methodology for real-time control of the ducted fan. Specifically, the method is compared to a static hover LQR controller and a gain-scheduled controller for stabilization of a step disturbance to the ducted fan. Results show that the MPC controllers have a bigger region of attraction than the static hover LQR controller and perform comparably to the gain-scheduled controller.

This paper is organized as follows: The experimental set-up is detailed in §2, the MPC problem defined in §3, LQR design details in §4, results in §5 and conclusions in §6.

2. FLIGHT CONTROL EXPERIMENT

The Caltech ducted fan is an experimental testbed designed for research and development of nonlinear flight guidance and control techniques for Uninhabited Combat Aerial Vehicles (UCAVs). The fan is a scaled model of the longitudinal axis of a flight vehicle and flight test results validate that the dynamics replicate qualities of actual flight vehicles (Milam and Murray, 1999).

2.1 *Hardware*

The ducted fan has three degrees of freedom: the boom holding the ducted fan is allowed to operate on a cylinder, 2 m high and 4.7 m in diameter, permitting horizontal and vertical displacements. Also, the wing/fan assembly at the end of the boom is allowed to rotate about its center of mass. Optical encoders mounted on the ducted fan, gearing wheel, and the base of the stand measure the three degrees of freedom. The fan is controlled by commanding a current to the electric motor for fan thrust and by commanding RC servos to control the thrust vectoring mechanism. The sensors are read and the commands sent by a dSPACE multi-processor system, comprised of a D/A card, a digital IO card, two Texas Instruments C40 signal processors, two Compaq Alpha processors, and a ISA bus to interface with a PC.

2.2 *Software*

The dSPACE system provides a real-time interface to the 4 processors and I/O card to the hardware. The NTG code resides on one of the alpha processors. A detailed description of NTG as a real-time trajectory generation package for constrained mechanical systems is given in (Milam *et al.*, 2000). The package is based on finding trajectory curves in a lower dimensional space and parameterizing these curves by B-splines. Sequential quadratic programming (SQP) is used to solve for the B-spline coefficients that optimize the performance objective, while respecting dynamics and constraints. The package NPSOL (Gill *et al.*, 1998) is used to solve the SQP problem.

2.3 *Model of the ducted fan*

The full nonlinear model of the fan including aerodynamic and gyroscopic effects is detailed in (Milam and Murray, 1999). For the implementation of the receding horizon approach outlined in this paper, the planar model of the fan will be utilized. The ODE's for the planar ducted fan may be written

$$\begin{aligned} m\ddot{x}\cos\theta - (m\ddot{z} - mg)\sin\theta &= F_{X_b} \\ m\ddot{x}\sin\theta + (m\ddot{z} - mg)\cos\theta &= F_{Z_b} \\ (J/r)\ddot{\theta} &= F_{Z_b}. \end{aligned} \tag{1}$$

The configuration variables x and z represent, respectively, horizontal and vertical inertial translations of the fan while θ is the rotation of the fan about the boom axis. These variables are

measured and their derivatives are computed with a FIR filter. The inputs F_{X_b} and F_{Z_b} are the body-fixed axial and transverse thrust components, respectively. For notational ease later we define the vector variables $\boldsymbol{x} = [x, z, \theta, \dot{x}, \dot{z}, \dot{\theta}]^T$ and $\boldsymbol{u} = [F_{X_b}, F_{Z_b}]^T$.

3. APPLICATION OF MPC

This section outlines the MPC problem and a timing method for updating real-time trajectories while accounting for non-negligible computational time. The adopted control approach is a hybrid of MPC and Control Lyapunov Function (CLF) based ideas (Jadbabaie *et al.*, 1999).

3.1 *MPC formulation*

In MPC, the current optimal control $\boldsymbol{u}_T^*(\tau; \boldsymbol{x}_0)$, $\tau \in [t_0, t_0 + T]$ for current state $\boldsymbol{x}_0$ at time t_0 is the argument that respects the following scalar objective:

$$\inf_{\boldsymbol{u}(\cdot)} \int_{t_0}^{t_0+T} q(\boldsymbol{x}(\tau), \boldsymbol{u}(\tau))\, \mathrm{d}\tau + V(\boldsymbol{x}(t_0+T)), \quad (2)$$

$$\text{s.t. } \dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}),\ \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad (3)$$
$$\boldsymbol{a} \le \boldsymbol{\psi}(\boldsymbol{x}(t_0), \boldsymbol{u}(t_0)) \le \boldsymbol{b},$$
$$\boldsymbol{c} \le \boldsymbol{\phi}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \le \boldsymbol{d}.$$

The vector $\boldsymbol{\phi}$ is a trajectory constraint (enforced over the entire time interval) while $\boldsymbol{\psi}$ is an initial time constraint. The control objective is to steer the state to an equilibrium point, usually the origin. No terminal constraint is enforced in this study. In theory, the resulting control $\boldsymbol{u}_T^*(\cdot)$ is instantaneously applied until a new state update occurs, usually at a prespecified sampling interval of time δ seconds. Repeating these computations yields a feedback control law.

For the ducted fan problem, differential equation (3) corresponds to equation (1) and the equilibrium point of interest is hover:

$$\boldsymbol{x}_{eq} \triangleq [0, 0, \pi/2, 0, 0, 0]^T, \quad \boldsymbol{u}_{eq} \triangleq [mg, 0]^T. \quad (4)$$

The vector $\boldsymbol{\psi}$ contains 6 initial conditions, or equality constraints, on the state $\boldsymbol{x}$ and two constraints on the inputs. The inputs are initially constrained to be within 0.25 N of the previously computed optimal inputs, at the appropriate instant of time. This amount of time will be detailed in the next section that describes how the timing of the MPC process is done.

The sole trajectory constraint on the state is $-1 \le z \le 1$. For the tests considered in this paper, the fan does not hit the boundaries of this constraint, so it is not included in the optimization problem. Trajectory constraints on the inputs are

$$\begin{bmatrix} 0 \\ -F_{X_b}^{max}/2 \end{bmatrix} \le \begin{bmatrix} F_{X_b} \\ F_{Z_b} \end{bmatrix} \le \begin{bmatrix} F_{X_b}^{max} \\ F_{X_b}^{max}/2 \end{bmatrix}, \quad (5)$$

where $F_{X_b}^{max}$ is 11 N and mg is 7.3 N. With respect to equation (2), the cost is defined as

$$q(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{2}\hat{\boldsymbol{x}}^T \boldsymbol{Q} \hat{\boldsymbol{x}} + \frac{1}{2}\hat{\boldsymbol{u}}^T \boldsymbol{R} \hat{\boldsymbol{u}}, \quad (6)$$
$$V(\boldsymbol{x}) = \gamma \hat{\boldsymbol{x}}^T \boldsymbol{P} \hat{\boldsymbol{x}}, \quad \text{where}$$
$$\hat{\boldsymbol{x}} \triangleq \boldsymbol{x} - \boldsymbol{x}_{eq} = [x, z, \theta - \pi/2, \dot{x}, \dot{z}, \dot{\theta}]^T,$$
$$\hat{\boldsymbol{u}} \triangleq \boldsymbol{u} - \boldsymbol{u}_{eq} = [F_{X_b} - mg, F_{Z_b}]^T,$$
$$\boldsymbol{Q} = diag\{4, 15, 4, 1, 3, 0.3\},$$
$$\boldsymbol{R} = diag\{0.5, 0.5\},$$

$\gamma = 0.075$ and P is the unique stable solution to the algebraic Riccati equation corresponding to the linearized dynamics of equation (1) at hover and the weights $\boldsymbol{Q}$ and $\boldsymbol{R}$. Note that if $\gamma = 1/2$, then $V(\cdot)$ is the CLF for the system corresponding to the LQR problem. Instead V is a relaxed (in magnitude) CLF, which achieved better performance in the experiment. In either case, V is valid as a CLF only in a neighborhood around hover since it is based on the linearized dynamics. We do not try to compute off-line a region of attraction for this CLF. Experimental tests *without* any terminal cost and/or the input constraints leads to *instability*. The results in this paper show the success of this choice for V for stabilization. An inner-loop PD controller on $\theta, \dot{\theta}$ is implemented to stabilize to the open-loop MPC states $\theta_T^*, \dot{\theta}_T^*$. The θ dynamics are the fastest for this system and although most MPC controllers were found to be nominally stable without this inner-loop controller, small disturbances could lead to instability.

The optimal control problem is set-up in NTG code by parameterizing the three position states (x, z, θ), each with 8 B-spline coefficients. Over the receding horizon time intervals, 11 and 16 breakpoints were used with varying horizon lengths. Breakpoints specify the locations in time where the differential equations and any constraints must be satisfied, up to some tolerance. The value of $F_{X_b}^{max}$ for the input constraints is made conservative to avoid prolonged input saturation on the real hardware. The logic for this is that if the inputs are saturated on the real hardware, no actuation is left for the inner-loop theta controller and the system can go unstable. The value used in the optimization is $F_{X_b}^{max} = 9$ N.

3.2 *Timing set-up*

Computation time is nonnegligible and must be considered when implementing the optimal trajectories. The computation time varies with each optimization as the current state of the ducted fan changes. The following notational definitions will facilitate the description of how the timing is set-up.

i - Integer counter of MPC computations.
t_i - Value of current time when MPC computation i started.
$\delta_c(i)$ - Computation time for computation i.

$\boldsymbol{u}_T^*(i)(t)$ - Optimal output trajectory corresponding to computation i, with time interval $t \in [t_i, t_i + T]$.

A natural choice for updating the optimal trajectories for stabilization is to do so as fast as possible. This is achieved here by constantly resolving the optimization. When computation i is done, computation $i+1$ is immediately started, so $t_{i+1} = t_i + \delta_c(i)$. Figure 2 gives a graphical picture of the timing set-up as the optimal input trajectories $\boldsymbol{u}_T^*(\cdot)$ are updated. As shown in the

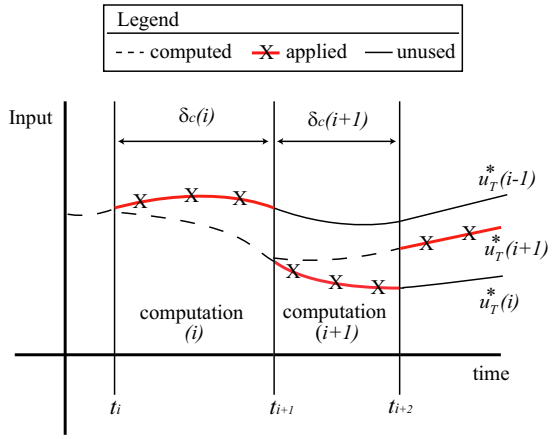

Fig. 2. Receding horizon input trajectories

figure, any computation i for $\boldsymbol{u}_T^*(i)(\cdot)$ occurs for $t \in [t_i, t_{i+1}]$ and the resulting trajectory is applied for $t \in [t_{i+1}, t_{i+2}]$. At $t = t_{i+1}$ computation $i+1$ is started for trajectory $\boldsymbol{u}_T^*(i+1)(\cdot)$, which is applied as soon as it is available ($t = t_{i+2}$). For the experimental runs detailed in the results, $\delta_c(i)$ is typically in the range of [0.05,0.25] seconds, meaning 4 to 20 MPC computations per second. Each optimization i requires the current measured state of the ducted fan and the value of the previous optimal input trajectories $\boldsymbol{u}_T^*(i-1)$ at time $t = t_i$. This corresponds to, respectively, 6 initial conditions for state vector $\boldsymbol{x}$ and 2 initial constraints on the input vector $\boldsymbol{u}$. Figure 2 shows that the optimal trajectories are advanced by their computation time prior to application to the system. A dashed line corresponds to the initial portion of an optimal trajectory and is not applied since it is not available until that computation is complete. The figure also reveals the possible discontinuity between successive applied optimal input trajectories, with a larger discontinuity more likely for longer computation times. The initial input constraint is an effort to reduce such discontinuities, although some discontinuity is unavoidable by this method. Also note that the same discontinuity is present for the 6 open-loop optimal state trajectories generated, again with a likelihood for greater discontinuity for longer computation times. In this description, initialization is not an issue because we assume the MPC computations are already running prior to any test runs. This is true of the experimental runs detailed in the results.

4. LQR CONTROLLER DESIGN

For comparison, the MPC approach is compared to two LQR designs. One is a simple static LQR controller, designed with the planar ducted fan model equation (1) linearized around hover equation (4). The weights chosen are

$$\boldsymbol{Q} = diag\{4, 15, 4, 1, 3, 0.3\}, \ \boldsymbol{R} = diag\{0.5, 0.5\},$$

corresponding to the quadratic integrated penalty on the state and input vectors, respectively. These are the same weights used in equation (6).

The second design is a gain-scheduled LQR. Using the full aerodynamic/gyroscopic model of the ducted fan, equilibrium forces and angle of attack for forward flight at constant altitude z and forward velocity $\dot{x}$ are identified. In this case θ is the angle of attack, and it is possible to linearize the full model around the forward flight equilibrium values for $\dot{x}$ and θ, where the value for all other states is zero. The gain-scheduling weights chosen are

$$\boldsymbol{Q} = diag\{1, 1, 15, 30, 4, 0.3\}, \ \boldsymbol{R} = diag\{0.5, 0.5\}.$$

The relaxed weights on $(x, \dot{x})$ and increased weights on $(z, \dot{z})$ were chosen specifically to improve stability in the presence of the x disturbance investigated in the results. Both LQR controllers require no computational effort; the gain scheduling is done by table look-up on the current θ measurement.

5. RESULTS

The experimental results show the response of the fan with each controller to a 6 meter horizontal offset, which is effectively engaging a step-response to a change in the initial condition for x. The following subsections detail the effects of different MPC parameterizations, namely as the horizon changes, and the responses with the different controllers to the induced offset.

5.1 *Varying the Horizon length T*

The first comparison is between different MPC controllers, where time horizon is varied to be 1.5, 2.0, 3.0, 4.0 or 6.0 seconds. Each MPC controller uses 16 breakpoints. Figure 3 shows a comparison of the average computation time as time proceeds. For each second after the offset was initiated, the data corresponds to the average run time over the previous second of computation. There is a clear trend towards shorter average computation times as the time horizon is made longer. There is also an initial transient increase in average computation time that is greater for shorter horizon times. In fact, the 6 second horizon controller exhibits a relatively constant average computation time. One explanation for this trend is that, for this particular test, a 6 second horizon is closer to what the system can actually do. After 1.5 seconds, the fan is still far from the desired hover position and

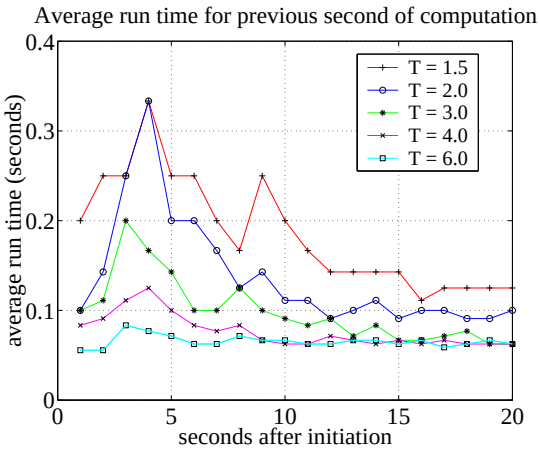

Fig. 3. Moving 1 second average of computation time for MPC implementation with varying horizon time

the terminal cost CLF is large, likely far from its region of attraction. Figure 4 shows the measured x response for these different controllers, exhibiting a rise time of 8-9 seconds independent of the controller. So a horizon time closer to the rise time results in a more feasible optimization in this case.

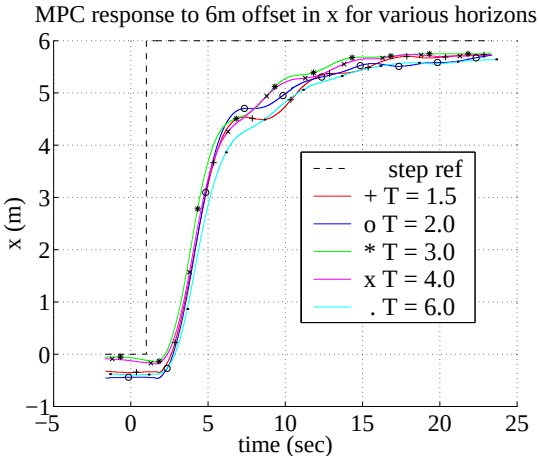

Fig. 4. Response of MPC controllers to 6 meter offset in x for different horizon lengths

As noted in §3.2, smaller computation time naturally leads to better tracking performance as shown in Figure 5. The figure shows the tracking of θ for the 6 and 1.5 second horizon controllers, respectively. The dotted lines (o) represent the computed open-loop optimal trajectories θ^*_T and the solid lines (x) are the measured θ. Discontinuity in θ^*_T for $T = 1.5$ sec reaches 0.25 radians ($t = 2.5$ sec) while for $T = 6.0$ all states are as smooth as θ^*_T shown.

5.2 *LQR vs. MPC*

Position responses due to the horizontal offset for the static hover LQR controller, the gain-scheduled LQR controller and two MPC controllers are shown in Figure 6. The MPC 6.0 sec-

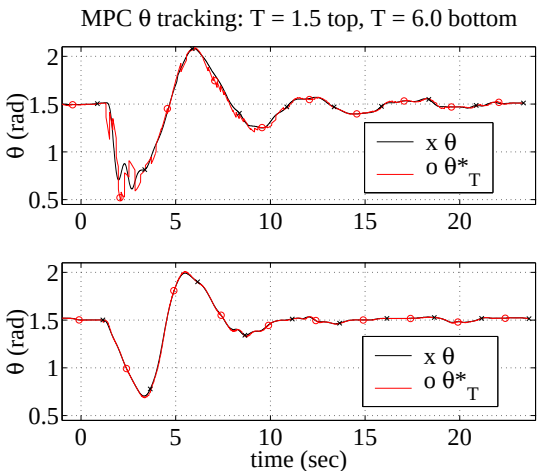

Fig. 5. Position tracking for 6 second horizon MPC controller

ond horizon response is the same as in the previous section. The 1.0 second horizon controller uses 11 breakpoints instead of 16, thereby reducing the computational demand. The LQR hover controller

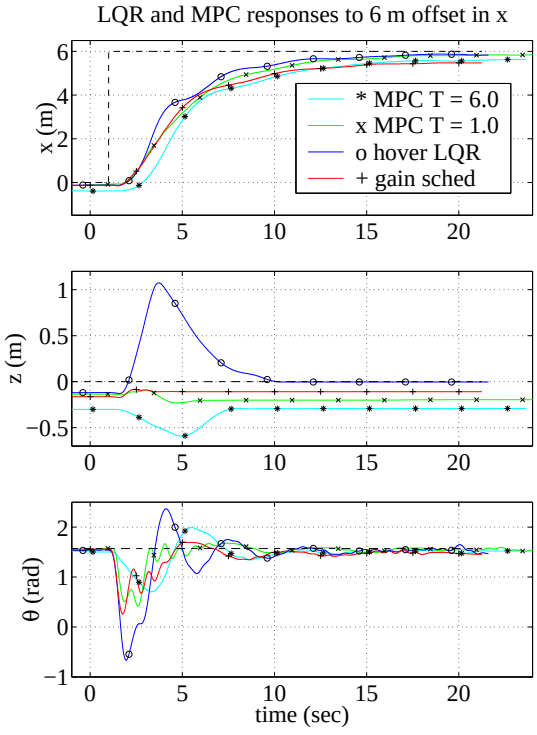

Fig. 6. Response of LQR and MPC controllers to 6 meter offset is x

could not stabilize the fan for an offset in x bigger than 4 meters. The unstable response to the 6 meter offset is shown in the figure as z exceeds the constraint set of $[-1, 1]$. In this case, the fan crashed and then redirected to the desired position. All other controllers were stabilizing, with remarkable similarity between the gain-scheduled LQR and MPC 1 second horizon controllers. For the MPC controllers, only the response of the fan is shown and not the open-loop optimal trajectories that were being tracked. In reference to Figure 3, the average computation profile for the 1 second horizon MPC controller with 11 breakpoints looks

like the 3 second horizon controller with 16 breakpoints. By reducing the number of breakpoints and hence the computational demand tracking is improved. The fan is thus stabilized along a path much different than the longer horizon path (observe the different θ responses). Clearly, there is a highly coupled interaction between horizon length and the number of breakpoints.

6. CONCLUSIONS

The Caltech ducted fan has been successfully realized as an experimental testbed for real-time MPC design. The real-time trajectory generation package NTG made it possible to run MPC controllers at speeds necessary for closed-loop stabilization of a system with considerable dynamics and strict input constraints. Specifically, the thrust vectoring mechanism dynamics are modeled as a second order filter with frequency 4 Hz and the 6 second horizon MPC controller ran faster than 10 Hz, a factor of 2.5 times greater than the actuator dynamics. Results show the success of different MPC controllers for stabilizing a step offset in x. A timing set-up based on "compute as fast as possible" accounts for nonnegligible computation time as it affects the application of the repeatedly updated optimal trajectories. The region of attraction of MPC controllers is shown to be larger than that of the static hover LQR controller. Moreover, the performance of some MPC controllers is close to that of the gain-scheduled LQR controller.

Extensions of this work could include a parametric study to better understand the nontrivial coupled relationship between the horizon length and number of breakpoints. An attempt could also be made to remove the inner-loop controller on the θ dynamics. It seems logical to apply a higher density of breakpoints over the time interval for which the optimal trajectories are applied. Another recent work (Franz *et al.*, 2002) details a different timing approach where the full aero/gyro model is used to estimate the state of the fan T' seconds ahead of its current state, where T' can either be chosen as a constant or can be taken as an average based on previous runs. The MPC problem is solved for the predicted state and applied when time has advanced by T' seconds. The work also considers how different timing approaches influence current theoretic stability results.

Acknowledgements

The authors greatly appreciate the contributions of Nicolas Petit for his insight and help with implementation. Thanks also goes to Mario Sznaier and John Hauser for their advice. This work was funded by DARPA grant F33615-98-C-3613.

7. REFERENCES

Franz, R., M.B. Milam and J. Hauser (2002). Applied receding horizon control of the caltech ducted fan. In: *Accepted: American Control Conference*.

Gill, P., W. Murray, M. Saunders and M. Wright (1998). *User's guide for NPSOL 5.0: A fortran package for nonlinear programming*. Systems Optimization Laboratory, Stanford University. Stanford, CA 94305.

Huang, Y.L., H.H. Lou, J.P. Gong and T.F. Edgar (2000). Fuzzy model predictive control. *IEEE Transactions on Fuzzy Systems* **8**(6), 665–678.

Jadbabaie, A., J. Yu and J. Hauser (1999). Receding horizon control of the caltech ducted fan: A control lyapunov function approach. In: *IEEE Conference on Control Applications*.

Kodogiannis, V.S., P.J.G. Lisboa and J. Lucas (1996). Neural network modelling and control for underwater vehicles. *Artificial Intelligence in Engineering* **10**(3), 203–212.

Low, K.S. and H.L. Zhuang (2000). Robust model predictive control and observer for direct drive applications. *IEEE Transactions on Power Electronics* **15**(6), 1018–1028.

Mayne, D.Q., J.B. Rawlings, C.V. Rao and P.O.M. Scokaert (2000). Contrained model predictive control: Stability and optimality. *Automatica* **36**, 789–814.

Milam, M.B. and R.M. Murray (1999). A testbed for nonlinear flight control techniques: the caltech ducted fan. In: *1999 Conference on Control Applications*.

Milam, M.B., K. Mushambi and R.M. Murray (2000). A new computational approach to real-time trajectory generation for constrained mechanical systems. In: *2000 Conference on Decision and Control*.

Mutha, R.K., W.R. Cluett and A. Penlidis (1997). Nonlinear model-based predictive control of control nonaffine systems. *Automatica* **33**(5), 907–913.

Ollero, A., A. Garciacerezo and J.L. Martinez (1994). Fuzzy supervisory path tracking of mobile robots. *Control Engineering Practice* **2**(2), 313–319.

Qin, S.J. and T.A. Badgwell (1997). An overview of industrial model predictive control technology. In: *Fifth International Conference on Chemical Process Control* (J.C. Kantor, C.E. Garcia and B. Carnahan, Eds.). pp. 232–256.

Scott, R.C. and L.E. Pado (2000). Active control of wind-tunnel model aeroelastic response using neural networks. *Journal of Guidance Control and Dynamics* **23**(6), 1100–1108.

Singh, L. and J. Fuller (2001). Trajectory generation for a uav in urban terrain, using nonlinear mpc. In: *Proceedings of the American Control Conference*.

Tan, Y., A.R. Van Cauwenberghe and M. Saif (2001). Neural network based direct optimizing predictive control with on-line pid gradient optimization. *Intelligent Automation and Soft Computing* **7**(2), 107–123.

Wan, E.A. and A.A. Bogdanov (2001). Model predictice neural control with application to a 6 dof helicopter model. In: *Proceedings of the American Control Conference*.