**Proceedings of the**
**44th IEEE Conference on Decision and Control, and**
**the European Control Conference 2005**
**Seville, Spain, December 12-15, 2005**

# Application of MPC to an active structure using sampling rates up to 25kHz

A. Wills, D. Bates, A. Fleming, B. Ninness, R. Moheimani.

*Abstract*— **In this paper we demonstrate the implementation of model predictive control (MPC) for vibration suppression of the first five bending modes of an active structure. For adequate performance, this requires a 5kHz sampling rate, which is achieved using a standard active-set optimisation technique running on a 200MHz digital signal processor. Experimental results show that MPC offers improved performance for this application when compared with other standard approaches.**

## I. Introduction

In this paper we describe the application of MPC (Model Predictive Control) to an active structure, namely, the cantilever beam illustrated in Figures 1 and 2. This apparatus is a simple representation of many systems experienced in the field of active vibration control. Examples include flexible links, dual stage hard-drives, smart aerospace structures, and high-speed robotics [18]. The common objective is to augment mechanical damping through the use of piezoelectric strain actuators.

In situations where piezoelectric actuators have limited control authority, for example when hitting amplifier voltage limits, then control performance may suffer unduly. Standard techniques for active structural control [5], [21], do not explicitly cater for such a scenario.

In the field of process control, MPC has been successfully applied for the regulation of systems subject to constraints - see e.g. [13], [22] and [26]. Here, we investigate the use of MPC to provide active damping for the first five bending modes of the beam while satisfying input constraints. For performance, this requires sampling rates of 5kHz.

Although this application is of independent interest, the main contribution of this paper is the implementation of MPC using standard active-set optimisation techniques at sampling rates up to 25kHz using an inexpensive 200MHz DSP.

In terms of closed-loop performance, the preferred implementation uses a 12 step-ahead prediction horizon and a control interval of 200$\mu$s. Recorded worst-case solution times show that the QP is easily solved within this time, and the benefits of achieving this are clearly illustrated by experimental results. Furthermore, two sampling intervals, 5kHz and 25kHz, are compared to observe the effects of absolute prediction horizon (in seconds) on closed-loop performance.

Included in this paper is a description of the cantilever beam used in laboratory experiments together with an identified model of the beam - Section II. The state observer and MPC structure are provided in Section III. The Quadratic Programming (QP) structure and active-set solution method are explained in Section IV. Implementation details for the Digital Signal Processor (DSP) are given in Section V. Experimental results are presented in Section VI and Section VII concludes the paper.

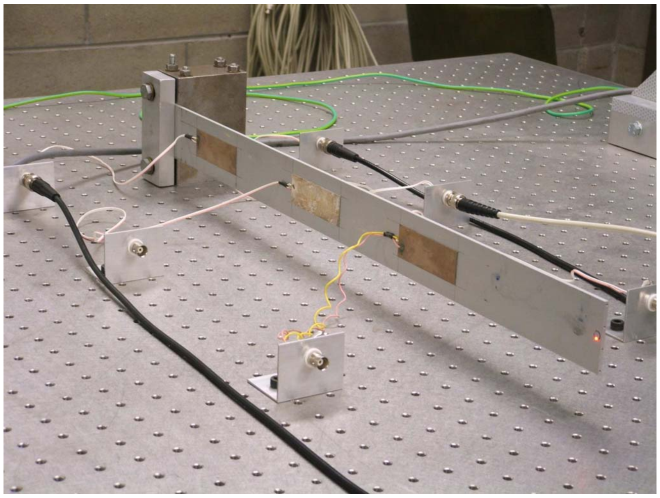

Fig. 1. Experimental apparatus.

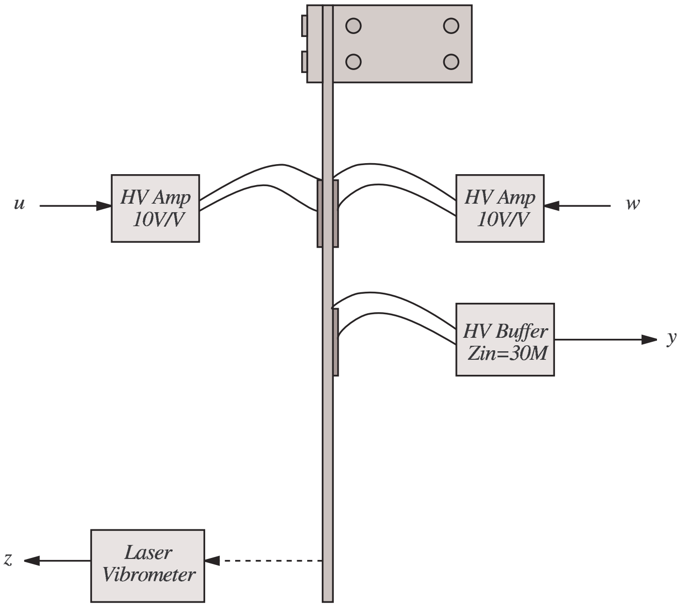

Fig. 2. Plan view schematic of the experimental apparatus.

## II. Plant - Cantilever Beam

As shown in Figures 1 and 2, the experimental setup comprises a uniform aluminium beam, clamped at one end, and free at the other. Such an apparatus is a simple representation

This work was supported by the Australian Research Council.

All authors are with the School of Electrical Engineering and Computer Science, University of Newcastle, NSW Australia. Corresponding author is Adrian Wills. Email: `Adrian.Wills@newcastle.edu.au`.

**0-7803-9568-9/05/$20.00 ©2005 IEEE**

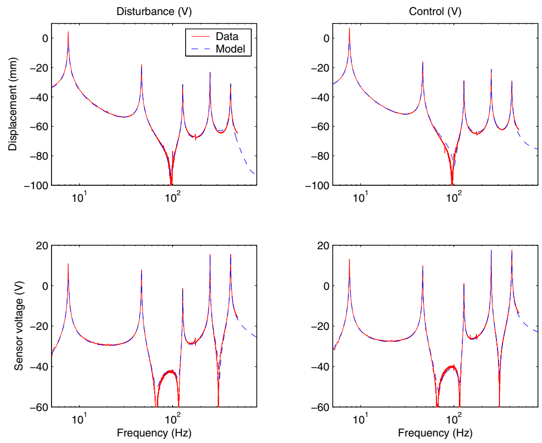

Fig. 3. Magnitude (dB) frequency response of identified beam model versus measured frequency response

of many systems experienced in the field of active vibration control.

Although six piezoelectric transducers are bonded to the front and rear surfaces, only three patches are required in this application. Remaining patches are short circuited to isolate their response from the structural dynamics.

The beam is $550mm$ in length, $3mm$ in thickness, and $50mm$ in width. The transducer centers are mounted 55 and $215mm$ from the clamped base. All transducers are manufactured from *Physik Instrumente* PIC151 piezoelectric ceramic and are $50mm$ in length, $25mm$ in width, and $0.25mm$ in thickness.

The disturbance and control signals, $w$ and $u$ respectively, are applied through high-voltage amplifiers to the base patches. The location of these patches, over an area of high modal strain, affords sufficient authority over all structural modes. The mechanical strain at the beam center, acquired by buffering the induced open-circuit transducer voltage, is utilised as the feedback variable $y$. For the purpose of performance analysis, a Polytec Laser Vibrometer is employed to measure the tip velocity.

### A. System Identification

The frequency domain class of subspace algorithms [15] has proven useful for the identification of high-order multivariable resonant structural and acoustic systems [16]. By applying a periodic chirp to each input successively (while zeroing the remaining input), a matrix of SISO frequency response measurements can be constructed. A total of 908 FFT points from the experimental system were utilised to identify a 14 state two-input two-output model[1]. A satisfactory fit in the frequency domain can be observed in Figure 3.

For the purposes of control, only the transfer function between input $u$ and voltage measurement $y$, and the transfer function between disturbance $\omega$ and measurement $y$ are considered. The corresponding discrete-time state-space relationship between $u$, $\omega$ and $y$ is given by

$$\begin{aligned} \xi_{t+1} &= A_p\xi_t + B_u u_t + B_\omega \omega_t, \\ y_t &= C_p \xi_t + D_u u_t + D_\omega \omega_t + \nu_t. \end{aligned}$$

In order to adequately observe and control the high-frequency content of $y$, sample rates of 5kHz (approximately 10 times the frequency of the highest mode under consideration) and 25kHz are tested. Comparisons are provided in Section VI.

The disturbance $\omega_t$ and measurement noise $\nu_t$ are assumed to be Gaussian distributed with zero mean and respective covariances of $\sigma_\omega$ and $\sigma_\nu$ with no cross-covariance, i.e.

$$\begin{bmatrix} \omega_t \\ \nu_t \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_\omega & 0 \\ 0 & \sigma_\nu \end{bmatrix} \right).$$

## III. Observer and MPC Structure

In this application, we are interested in rejecting disturbances $\omega$ using feedback control between the measurement $y$ and input $u$. An important consideration is that the control action $u$ has limited authority due to voltage bounds on the amplifier. Although this presents an ideal situation for using MPC it is challenging in light of the desired sample rates of 5kHz and 25kHz (i.e. sample intervals of $200\mu s$ and $40\mu s$ respectively).

In what follows we provide some background material on MPC and the specific structure used in this application. More detail on MPC can be found in many surveys including those by [1], [2], [6], [13], [14] and [19]. Surveys of industrial applications can be found in [22].

As the name suggests, MPC requires a model (see Section II) and a method of predicting future states/outputs of the plant. These predictions are used in determining an optimal (in terms of a control objective function) control action over a prediction horizon $N$.

Before considering the estimation problem, we discuss an embellishment of the plant model that is pertinent to closed-loop performance. From Section II we see that the model incorporates the first five modes of the beam only. Since higher frequency modes exist, but are unmodeled, their effect on closed-loop performance can be devastating, as indeed observed during experimental trials. To compensate for this we include a penalty on high frequency control action by augmenting the plant model to generate a high-pass filtered version of the input signal, and include a penalty on this term in the control objective function.

More precisely, we constructed a fourth order discrete-time Butterworth high-pass filter with 3dB roll-off point at 450Hz (*c.f.* frequency response in Figure 3). The state-space matrices corresponding to this filter are denoted by $(A_f, B_f, C_f, D_f)$ and the augmented system is given by

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + F\omega_t, \\ z_t &= Cx_t + Du_t + G\omega_t + H\nu_t. \end{aligned}$$

where

$$A = \begin{bmatrix} A_p & 0 \\ 0 & A_f \end{bmatrix}, \ B = \begin{bmatrix} B_u \\ B_f \end{bmatrix}, \ C = \begin{bmatrix} C_p & 0 \\ 0 & C_f \end{bmatrix},$$
$$D = \begin{bmatrix} D_u \\ D_f \end{bmatrix}, \ F = \begin{bmatrix} B_\omega \\ 0 \end{bmatrix}, \ G = \begin{bmatrix} D_\omega \\ 0 \end{bmatrix}, \ H = \begin{bmatrix} I \\ 0 \end{bmatrix}.$$

Combined with the 14 states used to represent the first five modes, this results in an 18 state model.

In terms of predicting the states we use a steady state Kalman predictor, which under the Gaussian assumptions on $\omega_t$ and $\nu_t$, provides optimal (in minimum variance sense)

[1] An implementation of McKelveys algorithm [15] is freely available by contacting Andrew.Fleming@newcastle.edu.au.

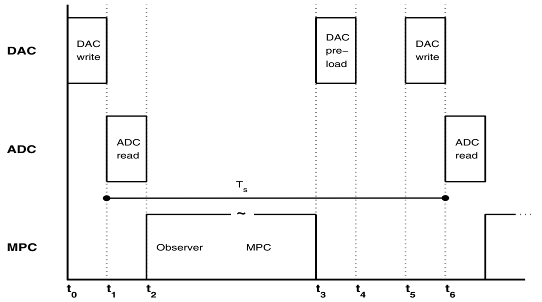

Fig. 4. Controller timing: $T_s$ is constant and $t_1$ and $t_6$ are the sampling "instants". $t_0$ (and $t_5$) mark the beginning of a DAC write operation, which preempts a delay so that writing the input and reading the output occur within tolerable margins.

prediction of states and outputs. This results in a Kalman gain matrix $L$ such that

$$
\begin{aligned}
\widehat{x}_{t+1|t} &= A\widehat{x}_{t|t-1} + Bu_t + \begin{bmatrix} L(y_t - \widehat{y}_{t|t-1}) \\ 0 \end{bmatrix}, && (1)\\
\widehat{y}_{t|t-1} &= [C_p \quad 0]\,\widehat{x}_{t|t-1} + D_u u_t, &&\\
L &= \left(A_p X C_p^T + Z\right)\left(C_p X C_p^T + V\right)^{-1}, && (2)\\
X &= W + A_p X A_p^T - L\left(C_p X C_p^T + V\right)L^T, &&\\
W &= B_\omega \sigma_\omega B_\omega^T, &&\\
V &= \sigma_\nu + \sigma_\omega D_\omega^2, &&\\
Z &= B_\omega \sigma_\omega D_\omega &&
\end{aligned}
$$

We actually require estimates of the state/output over the entire prediction horizon from time $t+1$ until time $t+N$. Since we already have $\widehat{x}_{t+1|t}$, the optimal estimates from time $t+1$ to $t+N$ can be obtained as follows (for more detail regarding optimal prediction see e.g. [11]).

$$
\begin{aligned}
\widehat{x}_{t+k+1|t} &= A\widehat{x}_{t+k|t} + Bu_{t+k}, && (3)\\
\widehat{z}_{t+k|t} &= C\widehat{x}_{t+k|t} + Du_{t+k}. && (4)
\end{aligned}
$$

Of course, the above predictions depend on future inputs $u_{t+1}, \dots, u_{t+N}$, which are free variables. This is important because it enables a search for the best (according to some control objective function) sequence of future inputs.

Concerning the control objective function $J$, we use the following structure

$$
J(\widehat{x}_{t+1|t}, \mathbf{u}_t) \triangleq \sum_{k=1}^{\infty} ||\widehat{z}_{t+k|t}||_Q^2 + ||u_{t+k}||_R^2,
$$

where $\mathbf{u}_t$ denotes an input sequence $\{u_{t+1}, u_{t+2}, \dots\}$ and $\widehat{z}_{t+k|t}$ denotes the controlled output estimate at time $t+k$ given input and output measurements up to and including time $t$. Of course this can be replaced by a finite horizon cost function with a penalty on the terminal state as follows

$$
J(\widehat{x}_{t+1|t}, \mathbf{u}_t) = ||\widehat{x}_{t+N+1|t}||_P^2 + \sum_{k=1}^{N} ||\widehat{z}_{t+k|t}||_Q^2 + ||u_{t+k}||_R^2,
$$

where $N$ is called the prediction horizon and $P$ is the solution to the following DARE,

$$
P = C^T QC + A^T PA - K\left(B^T PB + R + D^T QD\right)K^T, \quad (5)
$$

$$
K = -\left(A^T PB + C^T QD\right)\left(B^T PB + R + D^T QD\right)^{-1}. \quad (6)
$$

In addition to minimising the control objective function $J$, it is desirable that the input signal $u$ (see Figure 2) satisfies certain hard constraints, in this case upper and lower bounds due to amplifier voltage limits. Such constraints enter MPC in a natural manner as side conditions on the optimal control action calculation. More precisely, the optimal control action $\mathbf{u}_t^*$ over the prediction horizon $N$ is obtained by solving the following quadratic program given $\widehat{x}_{t+1|t}$

$$
\begin{aligned}
(\mathcal{MPC}): \quad \mathbf{u}_t^* \;&=\; \arg\min_{\mathbf{u}_t} \quad J(\widehat{x}_{t+1|t}, \mathbf{u}_t)\\
&\qquad\quad \text{s.t.} \quad \mathbf{u}_t \in \mathbb{U},
\end{aligned}
$$

where $\mathbb{U}$ is a, preferably non-empty, polyhedron. The first element of $\mathbf{u}_t^*$, namely $u_{t+1}$, is the optimal control move to be applied at the next time interval (i.e. $t_6$ in Figure 4). At the next time interval we obtain new information about the plant output and repeat the process - see Section V for more details.

## IV. Quadratic Programming Solver

In this Section we provide some detail on the construction, and method for solving the quadratic program ($\mathcal{MPC}$) online.

To simplify subsequent notation we define a stacked version of the predicted output and terminal state and the future inputs as follows.

$$
Z_t \triangleq \begin{bmatrix} \widehat{z}_{t+1|t} \\ \vdots \\ \widehat{z}_{t+N|t} \\ \widehat{x}_{t+N+1|t} \end{bmatrix}, \qquad U_t \triangleq \begin{bmatrix} u_{t+1} \\ \vdots \\ u_{t+N} \end{bmatrix}.
$$

From the relationships (3) and (4)

$$
Z_t = \Lambda \widehat{x}_{t+1|t} + \Phi U_t, \quad (7)
$$

where

$$
\Lambda \triangleq \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{N-1} \\ A^N \end{bmatrix}, \; \Phi \triangleq \begin{bmatrix} D & & & & \\ CB & D & & & \\ CAB & CB & D & & \\ \vdots & & & \ddots & \\ CA^{N-2}B & \cdots & \cdots & CB & D \\ A^{N-1}B & \cdots & \cdots & AB & B \end{bmatrix}.
$$

Using these definitions, the cost function $J$ can be expanded to offer a more convenient quadratic form in terms of $\widehat{x}_{t+1|t}$ and $U_t$ as

$$
J(\widehat{x}_{t+1|t}, U_t) = U_t^T H U_t + 2U_t^T f + c.
$$

Here $c$ is a constant term that may be safely ignored and the terms $H$ and $f$ are given by

$$
H = \Phi^T \bar{Q} \Phi + \bar{R}, \quad f = \Gamma \widehat{x}_{t+1|t}, \quad \Gamma = \Phi^T \bar{Q} \Lambda, \quad (8)
$$

with

$$
\bar{Q} \triangleq \begin{bmatrix} Q & & & \\ & \ddots & & \\ & & Q & \\ & & & P \end{bmatrix}, \quad \bar{R} \triangleq \begin{bmatrix} R & & & \\ & R & & \\ & & \ddots & \\ & & & R \end{bmatrix}
$$

It is important to note that $H$ depends on certain matrices, namely $A, B, C, D, Q, R$ and $P$ that change infrequently. This means that $H$ may, and should, be computed off-line. Furthermore, the size of this matrix is $Nn_u \times Nn_u$, but only half the entries need to be stored since it is symmetric by construction. On the other hand, only part of $f$ can be computed off-line, namely $\Gamma$, since $\widehat{x}_{t+1|t}$ is likely to change every control interval. Nevertheless, $f$ can be computed online using a matrix vector multiplication given in (8).

In light of the above definitions, $(\mathcal{MPC})$ can be equivalently stated as (where $\widehat{x}_{t+1|t}$ subsumed within $f$)

$$(\mathcal{MPC}): \quad U_t^* \quad = \quad \arg\min_{U_t} \quad U_t^T H U_t + 2U_t^T f \\ \text{s.t.} \quad U_t \in \bar{\mathbb{U}}$$

Under the assumption that $\bar{\mathbb{U}}$ can be constructed from linear equalities and linear inequalities (i.e. $\bar{\mathbb{U}}$ is a polyhedron), then $\mathcal{MPC}$ may be solved using many quadratic programming routines - see e.g. `http://www.numerical.rl.ac.uk/qp/qp.html`. However, for time-critical online optimisation it is often necessary to adapt these tools in order to exploit problem structure and circumvent inapplicable preprocessing of the problem instance.

For this application we have implemented[2] an active-set method based on the work of [8], [20] and [23]. This method requires a positive definite Hessian matrix $H$; which is automatically satisfied, for example, whenever $R$ is positive definite. It does not require a feasible initial point (in the primal space), which simplifies the algorithm. It handles equality, general inequality and simple bound constraints in a straightforward manner. It caters for "hot starting", i.e. where previous solutions are used to initialise the algorithm (although this feature is not exploited in our implementation). Furthermore, it has been refined in the open literature and open source community for some 20 years.

The method is based on maintaining two matrices $Z_a$ and $R_a$ such that $Z_a Z_a^T = H^{-1}$ and $Z_a^T A_a = R_a$, where the columns of $A_a$ hold the normals to the active constraints and $R_a$ is an upper triangular matrix. Note that since $H$ is computed offline then $Z_a$ can also be computed offline.

Each iteration of the algorithm involves the addition of a violated constraint - if any remain - to the active-set of constraints. There is also the possibility of dropping an already active constraint if no longer needed (i.e. an associated negative Lagrange multiplier). The implementation uses Givens rotations (see e.g. [9]) to update the matrices $Z_a$ and $R_a$ in a numerically robust fashion.

A common difficulty when using online optimisation algorithms is the uncertainty over solution time. Some methods offer better theoretical complexity limits than others - see e.g. [25] - however, in practice the efficacy and efficiency of an algorithm often depends on the problem instance (see e.g. [24]). In order to provide some idea of the performance of this algorithm we include a histogram of actual completion times for different horizon lengths in Section VI. Although a suitable interior-point algorithm was developed in C[3], further development into assembly language is an outstanding task.

## V. Implementation on DSP

Part of the philosophy when developing this controller was to ensure that the MPC algorithm including state observer and quadratic programming solver could be implemented on a standard DSP. The particular hardware used for the MPC experiments is an Analog Devices ADSP-21262 [3] evaluation kit connected to an ancillary board containing a single channel ADC (Analog-to-Digital Converter) and a single channel DAC (Digital-to-Analog Converter).

The ADSP-21262 is a 32-bit floating point DSP running at a clock speed of 200MHz. All instructions are single cycle (except division which is closer to eight) with the capability of performing 2 instructions in parallel; although this feature is not being used. All software used for online purposes was developed manually in assembler in order to minimise overhead. Furthermore, memory requirements for the algorithm fall well within the available limits of on-chip memory for this device.

For this application, as is often the case, MPC can be split into offline and online calculations. Offline calculations proceed as follows.

**Procedure 1** *Given the plant model in terms of the state-space system matrices $A, B, C, D$, compute the following:*

1) *Determine noise covariance terms $\sigma_\omega$ and $\sigma_\nu$.*
2) *Calculate $L$ according to Equation (2).*
3) *Choose the prediction horizon $N$ and state and input weighting matrices $Q$ and $R$ according to acceptable performance. Calculate $P$ according to (5).*
4) *Construct $H$ and $\Gamma$ according to (8).*
5) *Construct the constraint set $\mathbf{U}$ according to physical limitations and desired operating ranges.*

Once the above procedure is complete, MPC is implemented on the DSP according to the following algorithm, which is also depicted in Figure 4.

**Algorithm V.1** *At each time interval $t$ (corresponding to $t_0$ in Figure 4), complete the following tasks,*

1) *Apply the previously calculated control move $u_t$ to the system (calculated as $u_{t+1}$ in the previous iteration).*
2) *Measure the system output $y_t$.*
3) *Estimate the current state $\widehat{x}_{t+1|t}$ using the measured outputs and inputs according to (1).*
4) *Compute $f$ according to (8).*
5) *Compute the next control move $u_{t+1}$ by selecting the first control move from $U_t^*$, which is obtained by solving $\mathcal{MPC}$.*
6) *Preload DAC with $u_{t+1}$.*

Memory requirements necessary to perform the MPC algorithm are shown in Table I, where $n$ is the number of states, $p$ is the number of outputs, $m$ is the number of inputs and $N$ is the prediction horizon.

## VI. Experimental Results

In this section, some experimental results are presented for the cantilever beam apparatus described in Section II and the MPC algorithm described in Sections III, IV and V.

In terms of the offline procedure from Section V, we use the following values. The noise covariances are given by $\sigma_\omega = 1$ and $\sigma_\nu = 10^{-12}$ and the controller weighting matrices are set to $Q = \text{diag}\{10, 100\}$ and $R = 1$. The constraint set $\mathbb{U}$ was constructed as (simple bounds on actuator voltage prior to amplification - see Section II)

$$\mathbb{U} = \{U \in R^N : -0.2 \leq U_i \leq 0.2, \text{ for } i = 1, \ldots, N\}.$$

[2] For specific details and code, please contact Adrian Wills via `Adrian.Wills@newcastle.edu.au`

[3] This algorithm is based on ideas used in OOQP by [7]. In particular, Mehrotra's predictor-corrector approach [17] and Gondzio's multiple centrality correctors [10] are both employed.

The first experiment is intended to show that the controller performs to a satisfactory level in the absence of hitting constraints. To achieve this, we used a periodic chirp disturbance ranging from 5Hz to 800Hz and adjusted the disturbance gain so that actuator limits were not encountered. Figure 5 shows the open and closed loop response for this controller in both simulation and in practice. It can be seen that the controller is performing well and the match between simulation and actual results is satisfactory.

In light of this we proceeded to test the constraint handling capabilities of MPC by increasing the gain of the periodic chirp disturbance to ensure that limits were encountered.

In terms of comparing the performance we also tested an LQG controller that is "clipped" when hitting constraints. In fact, the particular LQG controller gain matrix is given by Equation (6) so that $u_{t+1} = K\widehat{x}_{t+1|t}$ is the optimal control action in the absence of constraints. Actually, by construction this control action coincides with that obtained from MPC when constraints are inactive. In fact, there exists a region for which SLQG and MPC coincide even when hitting constraints [4]. To be more precise the Saturated LQG control law (SLQG) is given by

$$u_{t+1} = \begin{cases} 0.2 & \text{if } K\widehat{x}_{t+1|t} > 0.2, \\ -0.2 & \text{if } K\widehat{x}_{t+1|t} < -0.2, \\ K\widehat{x}_{t+1|t} & \text{otherwise.} \end{cases}$$

Figure 6 shows time-domain plots of the measured beam tip velocity when disturbed by a band-pass filtered step function under the control of MPC and SLQG at 5kHz sample rate with prediction horizon $N = 12$. To generate the filtered step function an 8th order Butterworth filter is used with pass band between 230–270Hz, which corresponds to the 4th mode - see Figure 3.

Although there is visible evidence from Figure 6 that MPC outperforms SLQG, Figure 7 confirms this by showing the output and input energy signals for a series of filtered step disturbances under both MPC and SLQG.

In order to test the limitations of the current MPC implementation, we increased the sample rate from 5kHz to 25kHz. In so doing, we are faced with the reality that not only does the available computation time decrease by a factor of 5, but the prediction horizon $N$ needs to be increased by the same factor in order to have the same predictive capabilities as the 5kHz controller. In practice, this compounding effect forces us to reduce the prediction horizon from $N = 12$ to $N = 4$ instead of increasing it to $N = 60$ - compare with Table II which shows the worst-case solution time for the QP method under different prediction horizons. Figure 8 shows the output and input energy for the case of 25kHz sampling, and as expected there is now a marginal difference between MPC and SLQG.

The above results illustrate that MPC has an associated performance benefit for this application, although some care needs to be exercised when considering the sampling interval.

In terms of solving the quadratic program online and within the chosen sampling interval, we recorded the worst-case (i.e. maximum) time to solve the QP for different prediction horizons $N$ in Table II. For these values we considered a sample rate of 5kHz and used periodic chirp disturbance.

For each case, the controller is run for around 5 minutes (approximately 1,500,000 calls to the QP routine) and the maximum QP solve times are recorded. Also recorded in Table II are the number of times constraints were active at the solution during the test period.

Table III provides some idea of the distribution for QP solve times with prediction horizon $N = 12$. The disturbance signal applied is the same as for the previous test condition. Table IV shows the number of constraints active at the solution of the QP versus the number of times each case occurred.

## VII. Conclusion

While MPC has attracted enormous research interest, it is commonly perceived that difficulties associated with the required on-line optimisation limit their applicability. The empirical study of this issue has been a central motivation for this paper. While we believe that the achieved active structural control results are of interest in their own right, the problem was also chosen for its suitability in examining these computational overhead issues. In relation to this, a key outcome was to illustrate that using a very inexpensive hardware platform, the MPC action based on a non-trivial model (18 state), a reasonable prediction horizon (12 samples) and with constraints, can be straightforwardly computed in less than $150\mu$s.

With regard to this particular application, the quadratic program associated with MPC falls within a class of quadratic programs that can be modelled using a sector-bounded nonlinearity [12]. As such, standard robust stability results (including model uncertainty) are directly applicable. Although this issue falls outside the scope of this paper, it aligns well with previous work using robust control for active structures.

## VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the help of Will Heath for technical discussions and feedback. Will's help was vital to the development of this work.

## References

[1] H. Chen and F. Allgöwer. A quasi-infinite nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[2] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control, parts 1 and 2. *Automatica*, 23(2):137–148, 1987.

[3] Analog Devices. *Evaluation Kit for ADSP-21262 SHARC Processor*. `http://www.analog.com`, 1995-2005.

[4] J. De Dona, G. Goodwin, and M. Seron. Anti-windup and model predictive control: Reflections and connections. *European Journal of Control*, 6(5):467–477, 2000.

[5] C. R. Fuller, S. J. Elliott, and P. A. Nelson. *Active Control of Vibration*. Academic Press, 1996.

[6] C. E. Garcia, D. M. Prett, and M. Morari. Model predictive control: Theory and practice - A survey. *Automatica*, 25(3):335–348, 1989.

[7] E. M. Gertz and S. J. Wright. Object-oriented software for quadratic programming. Technical report, Optimization Technical Report 01-02, Computer Sciences Department, University of Wisconsin-Madison, October 2001. `http://www.cs.wisc.edu/∼swright/papers/`.

[8] D. Goldfarb and A. Idnai. A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27:1–33, 1983.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations, Third Edition*. The Johns Hopkins University Press, Baltimore, Maryland, 1996.

[10] J. Gondzio. Multiple centrality corrections in a primal-dual method for linear programming. *Computational Optiomization and Applications*, 6:137–156, 1996.

[11] G. C. Goodwin and K. S. Sin. *Adaptive filtering prediction and control*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1984.

[12] W. P. Heath and A. G. Wills. The inherent robustness of constrained linear model predictive control. Accepted for IFAC World Congress, Prague, 2005.

[13] J. M. Maciejowski. *Predictive Control with Constraints*. Pearson Education Limited, Harlow, Essex, 2002.

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

[15] T. Mckelvey, H. Akcay, and L. Ljung. Subspace based multivariable system identification from frequency response data. *IEEE Transactions on Automatic Control*, 41(7):960–978, July 1996.

[16] T. Mckelvey, A. J. Fleming, and S. O. R. Moheimani. Subspace-based system identification for an acoustic enclosure. *Transactions of the ASME, Journal of Vibration & Acoustics*, 124(3):414–419, 2002.

[17] S. Mehrotra. On the implementation of a primal-dual interior-point method. *SIAM Journal on Optimization*, 2:575–601, 1992.

[18] S. O. R. Moheimani, D. Halim, and A. J. Fleming. *Spatial Control of Vibration: Theory and Experiments*. World Scientific, 2003.

[19] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262–287, 1993.

[20] M. J. D. Powell. On the Qudratic Programming Algorithm of Goldfrab and Idnani. *Mathematical Programming Study*, 25:46–61, 1985.

[21] Andre Preumont. *Vibration Control of Active Structures*, volume 50 of *Solid Mechanics and its Applications*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[22] S. J. Qin and T. A. Badgewell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[23] K. Schittkowski. QL: A Fortran code for convex quadratic programming - User's guide, Report, 2003. Department of Mathematics, University of Bayreuth.

[24] M. J. Todd. The many facets of linear programming. Technical report, Ithaca, NY 14853, 2000.

[25] A. G. Wills and W. P. Heath. EE03016 – Interior-Point Methods for Linear Model Predictive Control. Technical report, School of Electrical Engineering and Computer Science, University of Newcastle, Australia, 2003.

[26] A. G. Wills and W. P. Heath. Application of barrier function model predictive control to an edible oil refining process. *Journal of Process Control*, 15(2):183–200, March 2005.

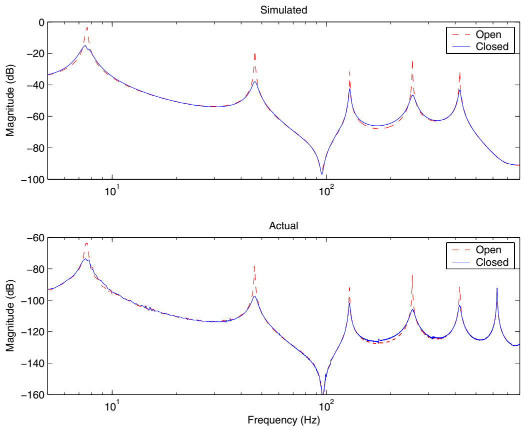

Fig. 5. Open and closed loop frequency responses (no actuator limits encountered)

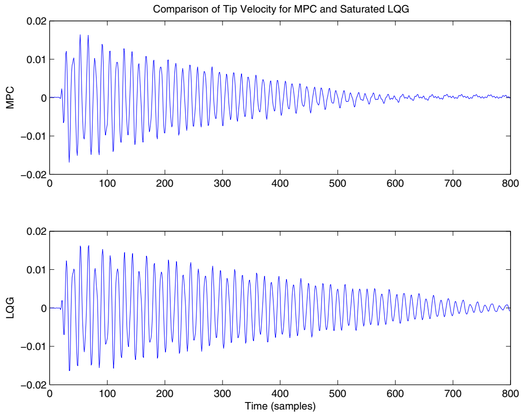

Fig. 6. Comparison of tip velocity for MPC and Saturated LQG (SLQG) running at 5kHz sampling rate when a filtered step function (for 4th mode) is applied to the disturbance patch. Notice a distinct reduction in settling time.

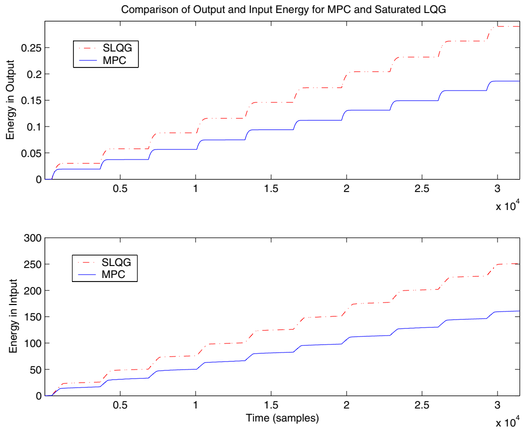

Fig. 7. Comparison of output and input energy for MPC and Saturated LQG (SLQG) running at 5kHz sampling rate when a series of filtered step functions (for 4th mode) are applied to the disturbance patch. Notice that both in terms of input and output energy, MPC outperforms SLQG.

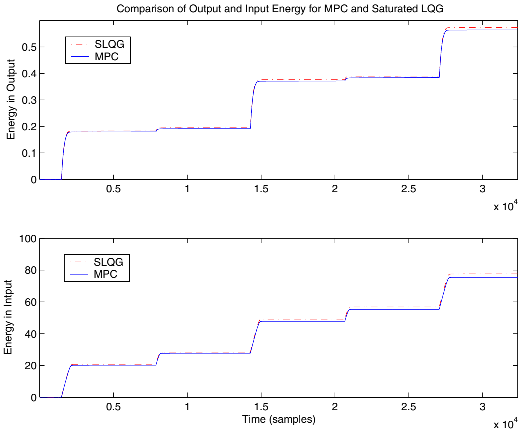

Fig. 8. Comparison of output and input energy for MPC and Saturated LQG (SLQG) running at 25kHz sampling rate when a series of filtered step functions (for 4th mode) are applied to the disturbance patch. There is a reduced performance saving compared with the 5kHz case.

| Data for | No. of 32 bit words |
|---|---|
| Observer | $n(n+m+p+2)+m+p+3$ |
| QP | $4(Nm)^2+Nm(n+6)+5$ |

TABLE I

MEMORY REQUIREMENTS FOR OBSERVER AND QUADRATIC PROGRAMMING SOLVER ON DSP.

| N | QP clock cycles | Hit constraints | Max. time ($\mu$s) |
|---|---|---|---|
| 4 | 5962 | 74748 | 29.81 |
| 6 | 10347 | 75609 | 51.735 |
| 8 | 13745 | 75088 | 68.725 |
| 10 | 20356 | 74379 | 101.78 |
| 12 | 27742 | 75384 | 138.71 |

TABLE II

WORST-CASE TIME TO SOLVE QP FOR DIFFERENT HORIZONS $N$.

| QP clock cycles | No. times QP solved |
|---|---|
| c $<$ 5000 | 1197566 |
| 5000 $\leq$ c $<$ 7500 | 90002 |
| 7500 $\leq$ c $<$ 10000 | 70154 |
| 10000 $\leq$ c $<$ 12500 | 48030 |
| 12500 $\leq$ c $<$ 15000 | 40836 |
| 15000 $\leq$ c $<$ 17500 | 32899 |
| 17500 $\leq$ c $<$ 20000 | 12444 |
| 20000 $\leq$ c $<$ 22500 | 4667 |
| 22500 $\leq$ c $<$ 25000 | 4923 |
| 25000 $\leq$ c $<$ 27500 | 534 |
| 27500 $\leq$ c $<$ 30000 | 5 |
| c $>$ 30000 | 0 |

TABLE III

SPREAD OF QP SOLVE TIMES FOR $N = 12$

| No. active | No. times |
|---|---|
| 1 | 106084 |
| 2 | 114076 |
| 3 | 55391 |
| 4 | 23208 |
| 5 | 5644 |
| 6 | 68 |
| 7–11 | 0 |
| 12 | 23 |

TABLE IV

NUMBER OF CONSTRAINTS ACTIVE AT SOLUTION (LEFT COLUMN) AND NUMBER OF TIMES ACTIVE (RIGHT COLUMN) FOR $N = 12$.