

# FPGA Implementation of an Interior-Point Solution for Linear Model Predictive Control <sup>\*</sup>

Adrian Wills\* Adam Mills\* Brett Ninness\*

*\* School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW 2308, Australia (e-mail: {Adrian.Wills, Adam.Mills, Brett.Ninness}@newcastle.edu.au)*

---

**Abstract:** The work here is directed at examining a model predictive control (MPC) implementation that takes advantage of recent advances in the availability of high performance computing platforms at modest cost. The focus here is on the potential for developing custom architecture solutions on field programmable gate array (FPGA) platforms. This is illustrated by demonstrating the solution of a disturbance rejection problem on a real 14<sup>th</sup> order lightly damped resonant system at 200 $\mu$ s sampling rate, using only 30 $\mu$ s to compute the control action.

*Keywords:* Model Predictive Control, Field Programmable Gate Array (FPGA), Interior-Point

---

## 1. INTRODUCTION

The benefits of model predictive control (MPC), particularly its capacity to address multivariable systems subject to constraints, have led to widespread interest in the approach Mayne et al. (2000); Qin and Badgwell (2003); Morari and Lee (1999); Maciejowski (2002). Unfortunately, balancing the advantages of the MPC approach, is the difficulty of the associated computational requirements. These arise from the need to solve a constrained optimisation problem within the chosen sampling period.

Historically, this has resulted in the benefits of MPC being realised only on systems with relatively slow dynamics, for which the sampling period is commensurately long and provides sufficient time for the associated compute platform to solve the necessary optimisation Maciejowski (2002).

More recently, there has been significant expansion in the availability of high performance computing platforms at low cost. In particular, field programmable gate array (FPGA) platforms offering very significant custom computing architecture resources, have expanded rapidly in capability at very modest cost Underwood (2004).

The control community has noticed this, and has begun a vigorous program of examining how these computer architecture advances can be employed to deliver the advantages of MPC on a wider range of control problems ACC (2006); CDC (2006); ECC (2009).

One line of research, known as “explicit MPC” has examined how the control solution can be decomposed offline into piecewise affine control laws valid over precomputed affine state region Tøndel et al. (2003); Johansen et al. (2007). This approach takes advantage of recent advantages in the availability of cheap high density computer

memory, capable of storing the potentially large number of affine controllers and state regions Wang and Boyd (2010); Johansen et al. (2007).

Another line of research has examined how advances in processor speed and/or the potential for implementation of custom designed FPGA based compute platforms may be exploited Wang and Boyd (2010); Ling et al. (2006); Constantinides (2009); Lau et al. (2009); Vouzis et al. (2009).

This paper is a contribution to this latter line of research. Here, we present the development of an FPGA-based custom architecture platform for MPC control based on an interior-point approach to the underlying optimisation problem. This complements previous work by the authors where the use of an alternative active-set method for solving the optimisation problem was examined Wills et al. (2010). Examining the relative benefits and tradeoffs involved with these two approaches to MPC implementation is a topic of current interest Lau et al. (2009), and it is intended that the work here in combination with Wills et al. (2010) contributes to this area.

Relative to a standard architecture microprocessor solution an FPGA-based custom architecture solution (such as presented here and in Wills et al. (2010)) has two advantages. It can simply and directly be “dropped in” as a computing core in a larger embedded systems design. Alternatively, it can be an advanced initialisation of a custom application specific integrated circuit (ASIC) design.

The capabilities of this design are illustrated by profiling its performance in solving a disturbance rejection problem for a 14<sup>th</sup> order resonant structure involving a lightly damped flexible beam. The computing platform involves a widely available and modest cost Altera Stratix III EP150F115C2 FPGA, clocked at 70 MHz.

---

\* This work was supported by the Australian Research Council through their Discovery Project Program.

The dynamics of the relevant process are relatively fast and necessitate a  $200\mu\text{s}$  sample period. Relative to this, the solution presented here is illustrated to require less than  $30\mu\text{s}$  of this sample period to compute the control action.

These results are presented as encouraging evidence that there is clear potential for the benefits of MPC to be more widely applied by taking advantage the current and future advances in the the availability of flexible, cheap and high performance computing hardware.

## 2. PROBLEM FORMULATION

This paper considers the control of linear, time invariant, discrete time systems, which are modelled in state-space form, according to

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad (1)$$

$$y_t = Cx_t + Du_t + v_t, \quad (2)$$

Here  $u_t \in \mathbb{R}^m$  is the system input (control variable),  $y_t \in \mathbb{R}^p$  is the system output,  $x_t \in \mathbb{R}^n$  is the state variable, and  $w_t \in \mathbb{R}^n$  and  $v_t \in \mathbb{R}^p$  are unknown disturbances on the state and output, respectively.

The model predictive control (MPC) approach considered here delivers the control variable  $u_{t+1|t}$  by solving at time  $t$  a constrained optimisation problem of the form

$$\mathbf{u}_t^* = \arg \min_{\mathbf{u}_t} V(\mathbf{u}_t, x_{t+1}) \quad \text{s.t.} \quad b_\ell \leq \mathbf{u}_t \leq b_u \quad (3)$$

where, for some user chosen prediction horizon  $N$ ,

$$\mathbf{u}_t \triangleq [u_{t+1|t}, u_{t+2|t}, \dots, u_{t+N|t}]. \quad (4)$$

In the above, the subscript  $t+k|t$  is used to denote a future control action at time  $t+k$ , which is based on measurements at time  $t$ .

MPC operates by using the first element  $u_{t+1|t}^*$  of  $\mathbf{u}_t^*$  as the control action to be applied at the next time interval. It then moves on to the next time instant,  $t+1$ , and solves (3) again to deliver  $u_{t+2|t+1}^*$  as the first element of  $\mathbf{u}_{t+1}^*$ , and so on. Accordingly, there is delay of one sample between measurement and control action that is intrinsic to the MPC approach (3) - see (Maciejowski, 2002, Section 2.5) for further discussion of this point.

In this paper, the control cost  $V(\mathbf{u}_t, x_{t+1})$  is assumed to have the following quadratic form

$$V(\mathbf{u}_t, x_{t+1}) \triangleq \mathbf{u}_t^T H \mathbf{u}_t + \mathbf{u}_t^T f(x_{t+1}) \quad (5)$$

where the matrix  $H \in \mathbb{R}^{Nm \times Nm}$  is assumed to be positive definite, and  $f(x_{t+1}) : \mathbb{R}^n \rightarrow \mathbb{R}^{Nm}$  is assumed to have the affine form

$$f(x_{t+1}) = \Phi x_{t+1} \quad (6)$$

for some user chosen matrix  $\Phi \in \mathbb{R}^{Nm \times n}$ .

The constraints in (3) represent simple bounds on allowed control action, which for example accommodates physical limits on the actuator movement. While this is a significant restriction to other MPC solutions that handle more general constraints, it does cater for an commonly encountered class of problems (see e.g. Wills et al. (2008)).

Importantly, this MPC approach requires knowledge of the future system state  $x_{t+k}$ , for  $k = 1, \dots, N$ . It must

therefore be predicted based on information available at the present time  $t$ , and these predictions substituted for  $x_{t+k}$ .

We will denote this prediction as  $\hat{x}_{t+k|t}$ . When  $k = 1$ , the predictor employed here will be of the linear observer form

$$\hat{x}_{t+1|t} = A\hat{x}_{t|t-1} + Bu_{t|t-1}^* + Le_t, \quad (7)$$

$$e_t = y_t - C\hat{x}_{t|t-1} - Du_{t|t-1}^*. \quad (8)$$

The observer gain matrix  $L$  is user-specified. In Section 5, it will be selected as the steady state Kalman filter gain. This is a common choice due to the predicted state then being of minimum variance.

This minimum variance property can be preserved for  $k > 1$  by employing the predictor

$$\hat{x}_{t+k|t} = A\hat{x}_{t+k-1|t} + Bu_{t+k|t}, \quad \text{for } k > 1 \quad (9)$$

which is ‘‘initialised’’ for  $k = 2$  by using the predictor (7),(8) in the right hand side of (9).

In what follows in section 5.4, a prediction  $\hat{y}_{t+k|t}$  of future system responses will also be required. This can be derived from the predicted state in the obvious fashion

$$\hat{y}_{t+k|t} = C\hat{x}_{t+k|t} + Du_{t+k|t}. \quad (10)$$

## 3. PROPOSED INTERIOR-POINT SOLUTION

The MPC approach described above requires the solution of the constrained optimisation problem (3). Via the cost specification (5), this is in the form of a strictly convex quadratic program that may be effectively solved using standard algorithms Wills and Ninness (a).

The two dominant approaches in this context are so-called ‘‘active-set’’ and ‘‘interior-point’’ methods. This paper elects to employ an interior-point technique (see e.g. Wills and Heath (2004)) to solve the quadratic program. In particular, we use a primal barrier approach since it may be implemented in a particularly elegant manner for this problem. The authors have also considered the active-set method in related work Wills et al. (2010); Knagge et al. (2009); Wills et al. (2008).

A primal barrier method for solving problem (3) transforms the problem into a class of related problems that do not have explicit constraints. Rather, the constraints are represented via a smooth ‘‘barrier’’ function. This resulting class of problems is typically parametrized by the positive scalar  $\mu$  as

$$V(\mathbf{u}_t, x_{t+1}, \mu) \triangleq \mathbf{u}_t^T H \mathbf{u}_t + \mathbf{u}_t^T f(x_{t+1}) + \mu B(\mathbf{u}_t), \quad (11)$$

$$B(\mathbf{u}_t) \triangleq - \sum_{i=1}^{Nm} \ln(b_u(i) - \mathbf{u}_t(i)) + \ln(\mathbf{u}_t(i) - b_\ell(i))$$

In the above,  $B(\cdot)$  is known as the logarithmic barrier function (Fiacco and McCormick, 1968) and the notation  $\mathbf{u}_t(i)$ ,  $b_\ell(i)$  and  $b_u(i)$  is used to refer to their  $i$ 'th respective elements.

The utility of writing the problem in the form of (11) is that it becomes a smooth problem that is directly amenable to Newton's method. Importantly, in the limit as  $\mu \rightarrow 0$  the solutions to (11) converge to the solution of (5) (see e.g. Fiacco and McCormick (1968)).

Along this line, the application of Newton's method to (11) results in the following update for the control action

$$\mathbf{u}_t \leftarrow \mathbf{u}_t - \alpha \mathcal{H}^{-1} g \quad (12)$$

where  $g$  is the gradient vector,  $\mathcal{H}$  is the Hessian matrix and the damping factor  $\alpha$  is a positive scalar that ensures a reduction in the cost function. More specifically, for the problem in (11), the  $i$ 'th element of the gradient vector can be represented as

$$g(i) = H(i, :)\mathbf{u}_t + f(i) + \frac{\mu}{b_u(i) - \mathbf{u}_t(i)} - \frac{\mu}{\mathbf{u}_t(i) - b_\ell(i)} \quad (13)$$

where  $H(i, :)$  refers to the  $i$ 'th row of  $H$  and  $f(i)$  to its  $i$ 'th element. The Hessian matrix can be expressed as

$$\mathcal{H} = H + \mu D \quad (14)$$

where  $D$  is a diagonal matrix whose  $i$ 'th diagonal element is given by

$$D(i) = \frac{1}{(b_u(i) - \mathbf{u}_t(i))^2} + \frac{1}{(\mathbf{u}_t(i) - b_\ell(i))^2} \quad (15)$$

Crucially, the above gradient and Hessian expressions are only true when the variables  $\mathbf{u}_t$  are strictly primal feasible (hence the name primal barrier method), i.e. when they satisfy  $b_\ell < \mathbf{u}_t < b_u$ . However, provided that  $b_u(i) \neq b_\ell(i)$  for all  $i$ , it is always possible to find an initial  $\mathbf{u}_t$  that satisfies this, for example

$$\mathbf{u}_t(i) = \frac{b_u(i) - b_\ell(i)}{2}. \quad (16)$$

Moreover, during the update of the control action in (12), the scalar  $\alpha$  can be chosen to maintain strict feasibility. These ideas are made concrete in the following algorithm.

---

#### Algorithm 1 Interior-Point Solution

---

Given an initial value for  $\mu$ , a final value  $\mu_\epsilon$ , and a value  $\zeta \in (0, 1)$ , perform the following steps:

- 1: Initialise the control action via (16).
  - 2: **while**  $\mu > \mu_\epsilon$  **do**
  - 3: Compute the search direction  $\rho = \mathcal{H}^{-1}g$  via (13), (18) and (15).
  - 4: Find a scalar  $\alpha$  such that  $V(\mathbf{u}_t - \alpha\rho, x_{t+1}, \mu) < V(\mathbf{u}_t, x_{t+1}, \mu)$  and so that  $b_\ell < \mathbf{u}_t - \alpha\rho < b_u$ .
  - 5: Update the control action via  $\mathbf{u}_t \leftarrow \mathbf{u}_t - \alpha\rho$ .
  - 6: Reduce the barrier weighting according to  $\mu \leftarrow \zeta\mu$ .
  - 7: **end while**
- 

The most computationally demanding step in Algorithm 1 involves the solution of  $\rho$ , which typically requires  $O(N^3 m^3)$  floating point operations on a serial processor. There are many ways to achieve this, but the approach considered here is to employ a conjugate gradient method (see e.g. Section 10 in Golub and Loan (1996)). Other researchers have also considered this approach within the MPC context (Roldao-Lopes et al., 2009).

Our rationale for using this approach is that it is an iterative method whose numerical operations are relatively simple. Importantly, it is often acceptable to stop the method after just a few iterations, which reduces the overall complexity while often delivering good approximate solutions (Roldao-Lopes et al., 2009).

To be clear, we are interested in using a conjugate gradient method for solving

$$\mathcal{H}\rho = g, \quad (17)$$

which only changes line 3 in Algorithm 1. These ideas are made more concrete in the following full algorithm description.

---

#### Algorithm 2 Final Interior-Point Algorithm

---

Given a value  $\nu \in (0, 1)$ , positive integers  $K$  and  $L$ , and an initial value for  $\mu > 0$  and perform the following steps:

- 1: Initialise the control action via  $\mathbf{u}_t(i) = (b_u(i) - b_\ell(i))/2$  for  $i = 1, \dots, Nm$ .
  - 2: **for**  $j = 1 : K$  **do**
  - 3: Compute an initial estimate of  $\rho$  by setting  $\rho(i) = g(i)/(H(i, i) + \mu D(i))$  for  $i = 1, \dots, Nm$ .
  - 4: Compute the residual  $r = g - H\rho - \mu D\rho$ .
  - 5: Set  $\sigma = r^T r$ .
  - 6: **for**  $k=1:L$  **do**
  - 7: **if**  $k=1$  **then**
  - 8:  $p = r$ .
  - 9: **else**
  - 10:  $\beta = \sigma/\eta$ .
  - 11:  $p \leftarrow r + \beta p$ .
  - 12: **end if**
  - 13:  $w = Hp + \mu Dp$ .
  - 14:  $\gamma = \sigma/p^T w$ .
  - 15:  $\rho \leftarrow \rho + \gamma p$ .
  - 16:  $r \leftarrow r - \gamma p$ .
  - 17:  $\eta = \sigma$ .
  - 18:  $\sigma = r^T r$ .
  - 19: **end for**
  - 20: **for**  $i = 1 : Nm$  **do**
  - 21: **if**  $\rho(i) < \mathbf{u}_t(i) - b_u(i)$  **then**
  - 22:  $\rho(i) = \nu(\mathbf{u}_t(i) - b_u(i))$ .
  - 23: **else if**  $\rho(i) > b_\ell(i) - \mathbf{u}_t(i)$  **then**
  - 24:  $\rho(i) = \nu(\mathbf{u}_t(i) - b_\ell(i))$ .
  - 25: **end if**
  - 26: **end for**
  - 27: Update the control action via  $\mathbf{u}_t \leftarrow \mathbf{u}_t - \alpha\rho$ .
  - 28: Update the barrier weighting via  $\mu \leftarrow \zeta\mu$ .
  - 29: **end for**
- 

A hardware description of this algorithm is provided in the next section together with specific choices for the parameter values it requires.

## 4. FPGA IMPLEMENTATION

This section provides a summary of the custom architecture developed to implement the MPC approach discussed above. Some key features of this architecture include:

- All calculations are performed using a custom floating point number representation;
- It provides a complete controller implementation, including state estimation;
- Pipelining and parallel computations are employed wherever possible, and in particular when computing inner products and performing matrix/vector multiplications.

A top-level design of the custom hardware solution is shown in Figure 1. The control algorithm is implemented in VHDL and compiled for an Altera Stratix III EP3K10K100-10C2 FPGA, which is interfaced to an A/D, D/A board in order to measure system responses and output control actions.

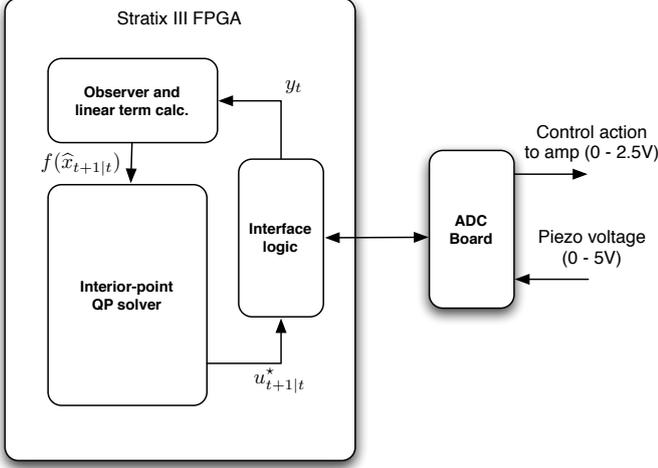


Fig. 1. Top level MPC circuit design

In terms of the logical flow, the circuit performs the following operations:

- At the beginning of a sample interval  $t$  the FPGA obtains new measurements of the system output  $y_t$  from the A/D and converts them into a custom floating point number format;
- The state observer uses these measurements to predict the state  $\hat{x}_{t+1|t}$  and compute the linear term  $f(\hat{x}_{t+1|t}) = \Phi \hat{x}_{t+1|t}$ ;
- The QP is solved to produce a new control action  $u_{t+1|t}^*$ ;
- This control action is converted into the correct format for the D/A circuit, which is loaded with the new value at the beginning of the next sample interval.

Almost all the above steps require various numerical operations which fundamentally depend on a finite precision number system. An essential part of this design is that it employs a custom floating point number format, in which a quantity  $z$  is represented according to

$$z = (-1)^s \times m \times 2^x \quad (18)$$

where  $s$  is sign bit,  $m$  is the  $n_m$ -bit unsigned mantissa and  $x$  is the  $n_x$ -bit exponent.

With the limited space available, it is not feasible to explain the entire circuit in full detail. Having said this, here we will attempt to explain aspects of the interior-point solver as outlined in Algorithm 2 since it is by far the most computationally demanding circuit.

Figure 2 provides a top level schematic for Algorithm 2 in which each of the blocks represents a key operation of the algorithm. Internally, they each depend on the ability to perform basic floating point operations in order to produce their respective outputs. As such, the design here includes custom circuits for addition, multiplication and division that have been designed to cater for the custom floating point number format used in this work. By way of example, the initial control action calculation from line 1 of Algorithm 2 is illustrated in Figure 3 and utilises the addition, and multiplication circuits.

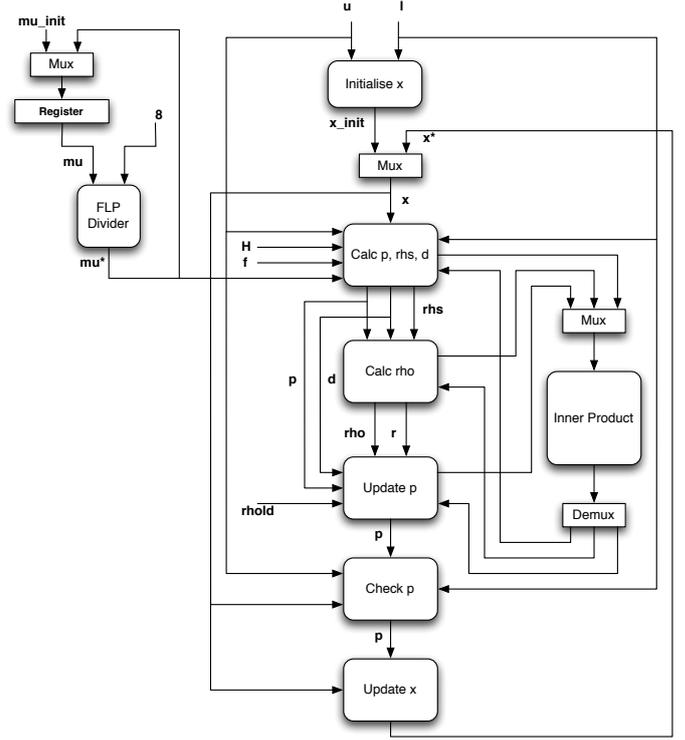


Fig. 2. Top level interior-point circuit design

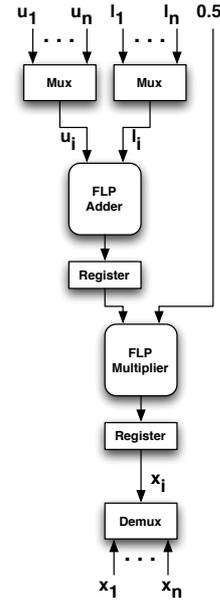


Fig. 3. Initialise the control action circuit from line 1 of Algorithm 2.

A vital part of the interior-point algorithm involves inner products of various vectors. To ensure that these operations complete in a timely manner, the inner product circuit is implemented in parallel wherever feasible. In particular, Figure 4 shows the inner product circuit used in the design for  $b_i = A_i x$ , where  $A_i$  is a row vector and  $x$  is a column vector.

The circuit consists of an initial multiplier bank for obtaining  $A_i(j)x(j)$  products, followed by several layers of addition in order to sum product pairs. This structure nat-

usually allows for Matrix/vector multiplications by adding a pipeline around the inner product. Each clock cycle allows a new row of the matrix  $A$  to be passed into the inner product circuit and each cycle after the circuit delay, it will provide  $b_i$ . This feature was heavily exploited in the design.

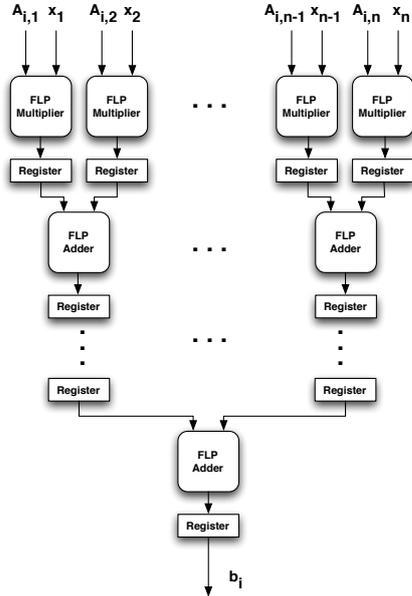


Fig. 4. Inner product circuit.

## 5. EXPERIMENTAL RESULTS

The performance of the custom architecture MPC design just summarised is illustrated in this section via the control of a lightly damped resonant structure.

### 5.1 Apparatus Description and Control Objective

The experimental setup comprises a uniform aluminium beam, clamped at one end, and free at the other, as illustrated diagrammatically in Figure 5. It is a representation of many systems encountered in the field of active vibration control Fuller et al. (1996).

The beam is 970mm in length, 5mm in thickness, and 25mm in width. Control and disturbance forces may be applied via *Physik Instrumente* PIC151 piezoelectric ceramic transducers which are 70mm in length, 25mm in width, and 0.25mm in thickness. The transducer centers are mounted 105mm (control) and 195mm (disturbance) from the clamped base. They are activated by 200V PDL200 high voltage amplifiers. These induce lateral control  $u_t$  and disturbance  $d_t$  beam displacements that are proportional to the applied voltage.

The resulting displacement  $y_t$  that occurs 105mm from the base is proportional to the mechanical strain at that point, and this is measured by buffering and acquiring the induced open-circuit voltage of a further piezoelectric transducer mounted there, on the other side of the beam to the actuation transducers.

The control objective is a disturbance rejection one. Namely, the control action  $u_t$  is to be used to minimize

displacement  $y_t$  resulting from the disturbance  $d_t$ . The supply rail limits of the voltage amplifiers imply hard constraints on the control authority  $u_t$ , which should be respected in the control design.

### 5.2 Apparatus Model

The MPC strategy considered in this paper is dependent on a model for the system to be controlled. For the flexible beam apparatus just described, this model may be obtained by first principles physical laws Fuller et al. (1996). The success of this approach depends on very careful and accurate physical measurement. Hence here we elect to obtain the model via system identification techniques.

For this purpose, the frequency response between the actuations  $u_t$ ,  $d_t$  and displacement  $y_t$  was measured at 3201 non-equidistant points in the range 1–500Hz. These were used together with the subspace-based identification method developed in McKelvey et al. (1996) to provide an initial  $n = 14$ 'th order state-space system estimate of the form

$$\xi_{t+1} = A_p \xi_t + B_p u_t + B_d d_t, \quad (19)$$

$$y_t = C_p \xi_t + D_p u_t + D_d d_t + e_t \quad (20)$$

$$\begin{bmatrix} d_t \\ e_t \end{bmatrix} \sim \mathcal{N} \left( \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_d^2 & 0 \\ 0 & \sigma_e^2 \end{bmatrix} \right) \quad (21)$$

where the subscript ‘‘p’’ denotes ‘‘plant’’ and  $A_p \in \mathbb{R}^{14 \times 14}$ . This model is then used as an initialisation that is further refined to deliver a final maximum-likelihood estimate using the techniques developed in Wills et al. (2009b). This dual stage approach was implemented using the freely available system identification toolbox Wills and Ninness (b); Wills et al. (2009a).

The frequency response of the resulting model from  $u_t$  to  $y_t$  is shown as the solid line in Figure 6, which can be compared to the measured frequency response shown as a dash dot line. The close agreement suggests accurate modeling. Similar comments apply to the modeling from disturbance  $d_t$  to  $y_t$  illustrated in Figure 7.

This model is now augmented due to some practical considerations. The first is that the buffer amplifier connected to the piezoelectric sensor can induce a constant offset of the displacement measurement  $y_t$ . Ignoring this issue will result in a MPC solution with artificially high DC gain.

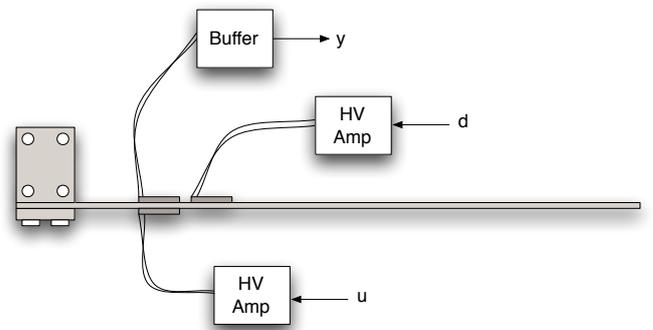


Fig. 5. Plan view schematic of the experimental apparatus.

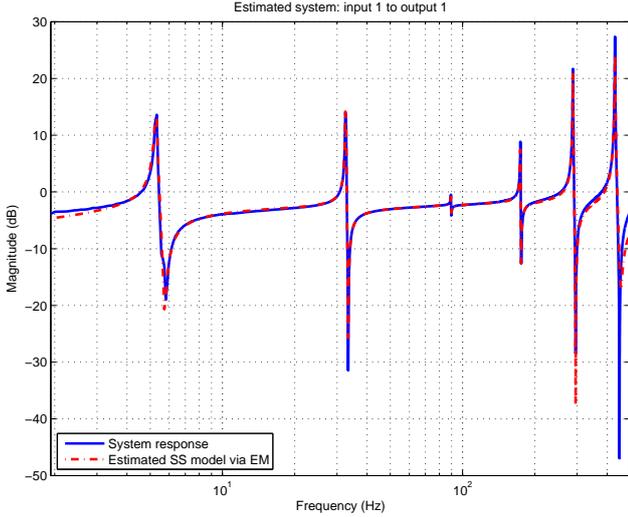


Fig. 6. Magnitude (dB)  $u_t$  to  $y_t$  frequency response of identified beam model (solid line) versus measured frequency response (dash dot line)

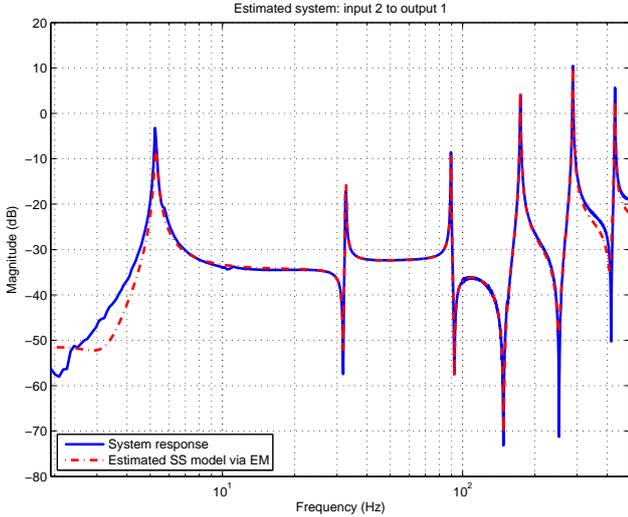


Fig. 7. Magnitude (dB)  $d_t$  to  $y_t$  frequency response of identified beam model (solid line) versus measured frequency response (dash dot line)

We address this by augmenting the estimated dynamics model (19),(20) so as to induce integral action in the MPC strategy. This is achieved by adding a new state  $\zeta_t$  as follows

$$\begin{bmatrix} \xi_{t+1} \\ \zeta_{t+1} \end{bmatrix} = \begin{bmatrix} A_p & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \xi_t \\ \zeta_t \end{bmatrix} + \begin{bmatrix} B_p \\ 0 \end{bmatrix} u_t + \begin{bmatrix} B_d & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} d_t \\ \mu_t \end{bmatrix}, \quad (22)$$

$$y_t = [C_p \ 1] \begin{bmatrix} \xi_t \\ \zeta_t \end{bmatrix} + D_p u_t + D_d d_t + e_t \quad (23)$$

where  $\mu_t$  is the noise associated with the unknown DC component from the buffer, which is modeled as

$$\mu_t \sim \mathcal{N}(0, \sigma_\mu^2). \quad (24)$$

The second important modeling consideration for this application is that of high frequency modes not captured

by the description (22),(23). Any high frequency control action that excites such modes will have a devastating effect on control performance.

This can be addressed by penalizing control action at any frequency above that of the highest modeled modes shown in Figure 6 and 7. This is achieved via a standard technique involving augmenting the model (22),(23) to deliver a new signal  $u_t^{\text{hf}}$ , which is a high-pass filtered version of  $u_t$ . The purpose of this is that  $u_t^{\text{hf}}$  may then be included as part of the penalty term  $V$  for the MPC action (3).

In this paper, a state space model

$$\eta_{t+1} = A_f \eta_t + B_f u_t \quad (25)$$

$$u_t^{\text{hf}} = C_f \eta_t + D_f u_t \quad (26)$$

was computed to correspond to an eighth order Butterworth high-pass filter with 3dB cut-off point at 500Hz. Adding this to the model (19), (20) delivers a final augmented model

$$x_{t+1} = A x_t + B u_t + w_t, \quad (27)$$

$$z_t = C x_t + D u_t + v_t \quad (28)$$

where

$$x_t = \begin{bmatrix} \xi_t \\ \zeta_t \\ \eta_t \end{bmatrix}, \quad z_t = \begin{bmatrix} y_t \\ u_t^{\text{hf}} \end{bmatrix}, \quad (29)$$

$$w_t = \begin{bmatrix} B_d & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_t \\ \mu_t \end{bmatrix}, \quad v_t = \begin{bmatrix} D_d & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} d_t \\ e_t \end{bmatrix}, \quad (30)$$

$$A = \begin{bmatrix} A_p & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & A_f \end{bmatrix}, \quad B = \begin{bmatrix} B_p \\ 0 \\ B_f \end{bmatrix}, \quad (31)$$

$$C = \begin{bmatrix} C_p & 1 & 0 \\ 0 & 0 & C_f \end{bmatrix}, \quad D = \begin{bmatrix} D_p \\ D_f \end{bmatrix}. \quad (32)$$

The above model has 23-states comprised of 14 for the beam dynamics, 8 for the high-pass filter and 1 for the DC component.

Finally, it is important to address the fact that due to the finite 200V amplifier supply rails, this linear model is only valid for input amplitudes satisfying  $|u_t| \leq 0.5$ Volts. This is modeled via the constraint for  $\Gamma \mathbf{u}_{t+1} \leq b(x_{t+1})$  used in the MPC formulation (3) with the choices

$$\Gamma = \begin{bmatrix} I_N \\ -I_N \end{bmatrix}, \quad b(\cdot) = 0.5 [1_{2N}], \quad b_0 = 0, \quad \Psi = 0. \quad (33)$$

In the above,  $I_N$  denotes the  $N \times N$  identity matrix and  $1_{2N}$  is used to denote a  $2N \times 1$  column vector with all entries equal to 1.

### 5.3 Observer Design

An essential use for the model just derived is the computation of an estimate  $\hat{x}_{t+1|t}$  of the system state  $x_{t+1}$  via an observer of the form (7). This involves choosing the observer gain  $L$  in (7), and in this paper we use the steady state Kalman gain. This approach depends on the state and measurement noise in the model (27),(28) obeying the Gaussian distribution

$$\begin{bmatrix} w_t \\ v_t \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} Q_o & S_o \\ S_o^T & R_o \end{bmatrix}\right). \quad (34)$$



and the scaling for  $\mu$  was selected as  $\zeta = 1/8$ . The multiplier  $\nu$  for retracting from the constraint boundaries was chosen as  $\nu = 0.95$ . Finally the integers  $K$  and  $L$  that determine the number of outer iterations of the interior-point method, and the number of iterations for the conjugate gradient method, respectively, were selected at  $K = 8$  and  $L = 2$ . These deserve some further comment.

Since the algorithm reduces  $\mu$  by  $\zeta = 1/8$  at each iteration, the final  $\mu$  value will be approximately  $7.63 \times 10^{-6}$ , which is small enough for our purposes here. On the other hand  $L = 2$  iterations of the conjugate gradient method is not likely to produce good approximate solutions. Nevertheless, it appears to perform well in practice.

As a final point, the custom floating point number format used in the FPGA circuit was chosen with  $n_m = 16$  mantissa bits and  $n_x = 6$  exponent bits.

Based on this configuration, an experiment was designed to confirm that the custom architecture FPGA implementation does in fact achieve the desired disturbance rejection objective. In this experiment the disturbance  $d_t$  was chosen as a periodic linear swept sine-wave signal (chirp) with a period of 20 seconds starting at 200Hz and finishing at 400Hz. The amplitude of  $d_t$  was set at 1.6 Volts, which was chosen to ensure that the input  $u_t$  would encounter the constraint limits  $\pm 0.5$  Volts.

Figures 8 and 9 show different sections of the open (grey outer envelope) and closed-loop (blue inner envelope) responses. Figure 8 shows 12 seconds of data while Figure 9 shows only 2.5 seconds in order to more clearly show the input signal.

It can be seen from the top plots in each Figure that the beam vibrations due to the chirp disturbance have been significantly reduced. The bottom plots in each Figure illustrate that the controller is “hitting” constraint limits in order to achieve the reduced vibrations. This demonstrates the efficacy of the custom architecture based MPC solution proposed in this paper.

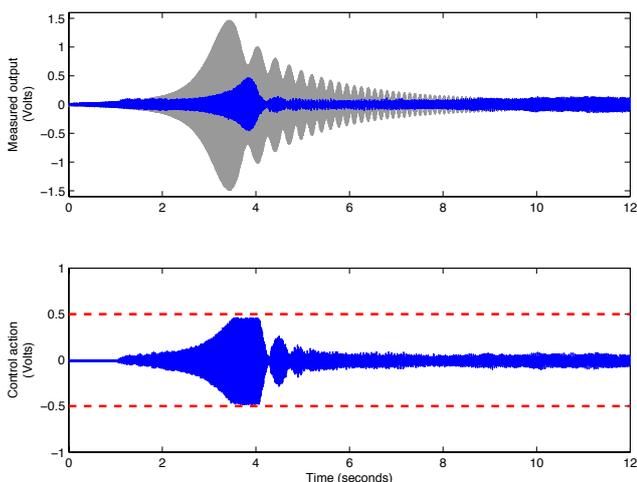


Fig. 8. Comparison of open-loop versus closed-loop control. Top: measured output response where the grey line indicates open loop response. Bottom: control action for closed-loop response with input limits shown as red dashed lines.

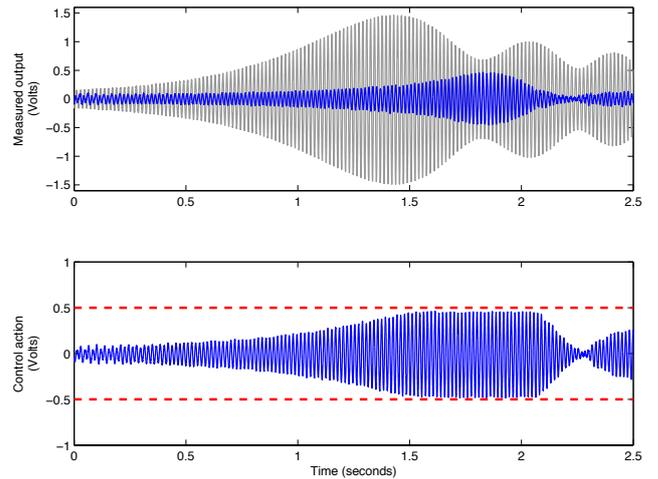


Fig. 9. Zoomed in comparison of open-loop versus closed-loop control. Top: measured output response where the grey line indicates open loop response. Bottom: control action for closed-loop response with input limits shown as red dashed lines.

## 6. CONCLUSION

This paper demonstrates the use of custom computing architectures for the purpose of implementing a model predictive control (MPC) algorithm. The approach proposed here is a full solution in that it incorporates A/D and D/A connectivity, a state observer, and an interior-point method for solve the MPC optimisation problem. Importantly, the design is capable of computing the required control action in less than  $200\mu$  seconds.

*Acknowledgement* The authors would like to sincerely thank Dr. Andrew Fleming for his generous assistance in the building and commissioning of the experimental apparatus used in this paper.

## REFERENCES

- (2006). Implementation of optimization-based control on a chip. Regular Session at the American Control Conference. Minneapolis, Minnesota USA.
- (2006). Model predictive control for fast nonlinear systems: existing approaches, challenges, and applications. 1-day pre-congress workshop at the 45th IEEE Conference on Decision and Control. San Diego, CA, USA.
- (2009). Invited session: Hardware implementation of model predictive control. European Control Conference, Budapest, Hungary.
- Constantinides, G. (2009). Tutorial paper: Parallel architectures for model predictive control. In *Proceedings of the European Control Conference, Budapest*, 138–143.
- Fiacco, A.V. and McCormick, G.P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. John Wiley & Sons Inc., New York.
- Fuller, C., Elliott, S., and Nelson, P. (1996). *Active Control of Vibration*. Academic Press.
- Golub, G. and Loan, C.V. (1996). *Matrix Computations: third edition*. Johns Hopkins University Press.
- Johansen, T., Jackson, W., Schreiber, R., and Tondel, P. (2007). Hardware synthesis of explicit model predictive controllers. *IEEE Transactions on Control System Technology*, 15(1), 191–197.

- Knagge, G., Wills, A., Mills, A., and Ninness, B. (2009). Asic and fpga implementation strategies for model predictive control. In *Proceedings of the European Control Conference (ECC)*.
- Lau, M., Yue, S., Ling, K., and Maciejowski, J. (2009). A comparison of interior point and active set methods for fpga implementation of model predictive control. In *Proc. European Control Conference*. European Union Control Association, Budapest.
- Ling, K., Yue, S., and Maciejowski, J. (2006). A FPGA Implementation of Model Predictive Control. In *Proceedings of the 2006 American Control Conference*, 1930–1935. Minneapolis, Minnesota, USA.
- Maciejowski, J. (2002). *Predictive Control with Constraints*. Pearson Education Limited, Harlow, Essex.
- Mayne, D., Rawlings, J., Rao, C., and Sokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- McKelvey, T., Akçay, H., and Ljung, L. (1996). Subspace-based multivariable system identification from frequency response data. *IEEE Transactions on Automatic Control*, 41, 960–979.
- McKelvey, T. and Helmersson, A. (1996). State-space parametrizations of multivariable linear systems using tridiagonal matrix forms. In *Proceedings of the 35th IEEE Conference on Decision and Control*, volume 4, 3654–3659.
- Morari, M. and Lee, J. (1999). Model predictive control: Past, present and future. *Computers and Chemical Engineering*, 23(4-5), 667–682.
- Qin, S.J. and Badgwell, T. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11, 733–764.
- Roldao-Lopes, A., Shahzad, A., Constantinides, G., and Kerrigan, E. (2009). More flops or more precision? accuracy parameterizable linear equation solvers for model predictive control. 209–216.
- Tøndel, P., Johansen, T., and Bemporad, A. (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica*, 39(3), 489–497.
- Underwood, K. (2004). Fpgas vs. cpus: trends in peak floating-point performance. In *FPGA '04: Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, 171–180. ACM, New York, NY, USA. doi: <http://doi.acm.org/10.1145/968280.968305>.
- Vouzis, P., Kothare, M., Bleris, L., and Arnold, M. (2009). A system-on-a-chip implementation for embedded real-time model predictive control. *IEEE Transactions on Control Systems Technology*, 17(5), 1006–1017. doi: 10.1109/TCST.2008.2004503.
- Wang, Y. and Boyd, S. (2010). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.
- Wills, A.G. and Heath, W.P. (2004). Barrier function based model predictive control. *Automatica*, 40(8), 1415–1422.
- Wills, A., Knagge, G., and Ninness, B. (2010). Fast model predictive control via custom integrated circuit architecture. *Submitted to IEEE Transactions on Control System Technology*.
- Wills, A., Bates, D., Fleming, A., Ninness, B., and Moheimani, S. (2008). Model predictive control applied to constraint handling in active noise and vibration control. *IEEE Transactions on Control Systems Technology*, 16(1), 3–12.
- Wills, A., Mills, A., and Ninness, B. (2009a). A matlab software environment for system identification. In *Proceedings of the 15th IFAC Symposium on System Identification, St. Malo, France*.
- Wills, A. and Ninness, B. (????a). QPC - Quadratic Programming in C. Webpage. <http://sigpromu.org/quadprog/>.
- Wills, A. and Ninness, B. (????b). UNIT - University of Newcastle Identification Toolbox. Webpage. <http://sigpromu.org/idtoolbox/>.
- Wills, A., Ninness, B., and Gibson, S. (2009b). Maximum likelihood estimation of state space models from frequency domain data. *IEEE Transactions on Automatic Control*, 54(1), 19–33.