Nonlinear Model Predictive Control of an Inverted Pendulum

Adam Mills, Adrian Wills, Brett Ninness

***Abstract*— In this paper, nonlinear model predictive control is applied to an inverted pendulum apparatus. The sample interval for control calculations is 25 milli-seconds and the associated non-convex constrained optimisation problem involves 61-variables with 241-constraints. Despite this being a challenging problem, it was solved on-line using a standard sequential quadratic programming approach on a modest hardware platform. The efficacy of the control algorithm is validated via experimental results.**

I. INTRODUCTION

Nonlinear Model Predictive Control (NMPC) is an attractive strategy for controlling complex systems [1]. It has been used for decades within process control industries [2], [3], because it offers good dynamic performance while ensuring operation within certain physical limits. This latter feature enables plant operators to run the plant near constraint boundaries, which can increase productivity and reduce product quality variation [2], [3].

However, a fundamental difficulty of this NMPC approach is that the implementation platform must be capable of solving a constrained optimisation problem within a specified time limit. This time decreases as the speed of the dynamics to be controlled increases. As a result, the implementation of NMPC has to date been generally limited to plants with slow or otherwise very simple dynamics so that the time constraints in computing a solution are relaxed.

Surmounting this difficulty of computational overhead to achieve the benefits of Model Predictive Control for linear systems has recently attracted significant research attention [4], [5], [6], [7].

A similar trend is emerging for NMPC applications. For example, [8] applies NMPC to prevent the flooding of of a river system, where a sample time of 5 minutes is used. In [9] NMPC is applied to the problem of normalising the blood glucose levels in the critically ill. This application has a sample time of 5 minutes which resolves some of the issue of computation time.

NMPC is applied to maximising the production rate of E. Coli in [10]. Here, online optimisation is eschewed in favour of a less computationally intensive control vector parametrisation method. NMPC is applied to Planar Vertical Take-Off and Landing (PVTOL) aircraft in [11] by using the structure of the model to convert a non-convex optimisation problem into two convex problems of lesser dimension.

Adam Mills, Adrian Wills and Brett Ninness are with the School of Electrical Engineering and Computer Science, University of Newcastle, Callaghan, NSW, 2308, Australia adam.mills@newcastle.edu.au, adrian.wills@newcastle.edu.au, brett.ninness@newcastle.edu.au

These applications have either slow dynamics which makes computation time less critical or use a simplified model to lessen the computation load. However, NMPC has also been applied to systems with fast dynamics.

In [12], NMPC is applied to a run-of-mine ore milling circuit problem. A successful simulation concluded that NMPC performed well at a sample time of 10 seconds. An investigation of NMPC in [13] for terrain avoidance in an unmanned rotorcraft found that NMPC performed well, in simulation, at a sample time of 20 ms. The authors of that paper note the difficulties in applying this simulation to a real-time situation.

In [14], the possibilities of applying NMPC to a rotary inverted pendulum are explored. However, the strategy for this case is not to find the optimal input for the pendulum, but to reduce the error at the end of the prediction horizon. An application of NMPC to the control of a robotic hand-eye system is described in [15] where a sample time of 1ms is used.

In this paper we are also interested in applying NMPC to systems with fast dynamics. In particular, we employ standard numerical algorithms running on a modest hardware platform to control an inverted pendulum apparatus at a 25 milli-seconds sample interval. Both input and state constraints are considered and the resulting non-convex optimisation problem has 61-variables with 241-inequality constraints.

The inverted pendulum problem was selected because it displays nonlinear dynamic behaviour, it is unstable about the desired operating point (pendulum standing up), and it is non-minimum phase. As an aside, it is representative of some practical applications. The Segway PT is a two wheeled (in parallel), self-balancing vehicle that transports a single person which uses the properties of the inverted pendulum. A walking humanoid robot displays inverted pendulum characteristics [16].

This provides a suitably challenging test case for NMPC and the experimental results presented in this paper are encouraging. A video of this application can be found at `http://sigpromu.org/mpc/pilot.html`.

The paper is organised as follows. Section II provides some background to the NMPC problem formulation. Section III discusses sequential quadratic programming, which is used to solve the NMPC optimisation problem. Section IV details the application of NMPC to the inverted pendulum, and Section V provides experimental results.

II. Nonlinear Model Predictive Control

The literature on Nonlinear Model Predictive Control (NMPC) is vast. Therefore, in this section we outline one variant of NMPC that, while still quite general (see e.g. [1]), caters for the types of applications presented in this paper. Namely, systems whose dynamic behaviour can be adequately described via a discrete-time nonlinear state-space model

$$x_{k+1} = f_k(x_k, u_k) \tag{1}$$

where the state $x_k \in \mathbb{R}^n$, the input $u_k \in \mathbb{R}^m$ and the function $f_k(\cdot,\cdot) \in \mathbb{R}^n$ maps the current state and input to the next state x_{k+1}.

The utility of this model is that we can predict the state trajectory over a prediction horizon N, so that suitable control action can be taken in order to move the state to the origin. More precisely, if we start at $k = 1$ with x_1 known, then

$$\begin{aligned} x_2 &= f_1(x_1, u_1) \\ &\vdots \\ x_{N+1} &= f_N(x_N, u_N) \end{aligned}$$

The point is that the state at some point in the future, say at sample k, depends only on the initial state x_1 and all the inputs from $\{u_1, \ldots, u_{k-1}\}$.

Therefore, provided that the input has enough control authority over the state from $t = 1, \ldots, N$, it is possible to choose an input sequence

$$\mathbf{u} = \{u_1, \ldots, u_N\} \tag{2}$$

that moves the initial state x_1 towards a desired region, e.g. the origin. This latter goal is typically achieved by minimising a cost function that determines control performance. One commonly occurring cost function is defined as

$$V(\mathbf{u}) = \sum_{k=1}^{N} x_{k+1}^T Q x_{k+1} + u_k^T R u_k \tag{3}$$

where $Q \in \mathbb{R}^{n\times n}$ is assumed to be a positive semi-definite matrix used to penalise state-movements about the origin. Similarly, R is assumed to be a positive definite matrix that penalises input movements from the origin.

With the above in mind, the NMPC control action is determined by minimising V_N over the future control moves $\mathbf{u}$. The first control action is applied to the system and the procedure repeats at the next time sample based on new measurements. Figure 1 shows a typical timing diagram for NMPC.

A salient feature of this approach is that physical limits on the system inputs and states can also be directly included into the optimisation problem. More precisely, if the input must lie in a set $\mathbb{U}$ and the state must lie in a set $\mathbb{X}$ for all time samples, then the NMPC optimisation problem becomes

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} V_N(\mathbf{u}) \quad \text{s.t.} \quad u_k \in \mathbb{U},\ x_{k+1} \in \mathbb{X} \tag{4}$$

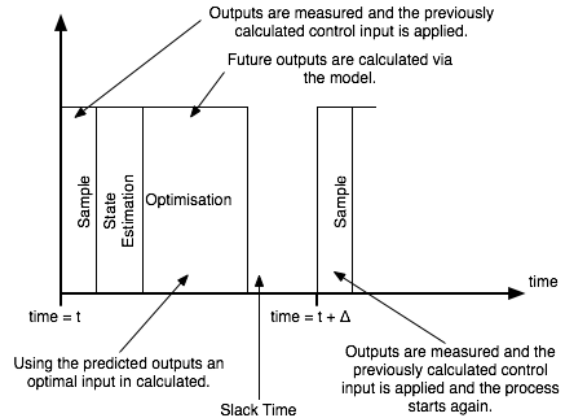

Fig. 1. Timing Diagram of NMPC

Solving this problem is non-trivial and usually forms the bulk of the computational work required to obtain the next control action in NMPC. Consequently, the sample times must be sufficiently large to allow for a solution to be found. This has resulted in applications of NMPC for systems with relatively large sample times. In the next section we outline a standard approach to solving these types of problems.

III. Solving the NMPC Optimisation Problem

In order to implement the NMPC algorithm we must be able to solve non-convex optimisation problems of the type in (4), which is a difficult task in general. Compounding this difficulty is the real-time aspects of NMPC, where a result must be ready within the chosen sample interval. In general we have to be satisfied in finding only a local minimum of the cost function, and even this is non-trivial within the time constraints.

Amongst the many approaches used to solve these types of problems, Sequential Quadratic Programming methods have emerged as a clear competitor [17].

Ignoring all previous definitions of x and f, for a moment, the types of problems we are interested in solving are of the following form

$$x^\star = \arg\min_x f(x), \quad \text{s.t.} \quad c(x) \le 0 \tag{5}$$

where the variables $x \in \mathbb{R}^n$, the cost $f : \mathbb{R}^n \to \mathbb{R}$ and the inequality constraint function $c : \mathbb{R}^n \to \mathbb{R}^m$.

In essence, SQP replaces the cost and constraint functions with a quadratic and linear approximation, respectively, to obtain a quadratic subproblem

$$\begin{aligned} (p^\star, \lambda^\star) = \arg\min_{p,\lambda} & \ p^T L_x(x,\lambda) + \frac{1}{2} p^T L_{xx}(x,\lambda) p \\ \text{s.t.} \quad & c(x) + p^T c_x(x) \le 0 \end{aligned} \tag{6}$$

where L is the Lagrangian function defined as

$$L(x,\lambda) = f(x) + \lambda^T c(x) \tag{7}$$

and L_x and L_{xx} denote the gradient and Hessian of L with respect to x.

In line-search SQP methods, this quadratic subproblem is solved to obtain the search direction $p^\star$ and Lagrange multipliers $\lambda^\star$, which is then used in a line search procedure to obtain a new estimate for x that reduces a merit function. We have used the ℓ_1 merit function

$$\phi(x,\mu) = f(x) + \frac{1}{\mu}\|c(x)\|_1 \tag{8}$$

where we employ $\mu = 1/(\|\lambda^\star\|_\infty + 0.1)$ as per (18.35) in Chapter 18 of [18].

Returning now to the NMPC case in (4). Since x_{k+1} is only a function of x_1 (which is known and given) and the inputs $u_1, \ldots, u_k$ then we can represent the constraints $u_k \in \mathbb{U}$ and $x_{k+1} \in \mathbb{X}$ for $k = 1, \ldots, N$ as

$$c(\mathbf{u}) \leq 0 \tag{9}$$

then the Lagrangian function becomes

$$L(\mathbf{u},\lambda) = V(\mathbf{u}) + \lambda^T c(\mathbf{u}) \tag{10}$$

Computing the Hessian of the Lagrangian function, denoted as $L_{\mathbf{uu}}$, involves second order derivative terms and is computationally expensive to calculate. Therefore, we crudely approximate $L_{\mathbf{uu}}$ by exploiting the sum-of-squares formulation in (3). Indeed, we can rewrite (3) as

$$V(\mathbf{u}) = e^T(\mathbf{u})e(\mathbf{u}) \tag{11}$$

where e is defined as

$$e(\mathbf{u}) = \begin{bmatrix} Q^{1/2}x_2 \\ \cdots \\ Q^{1/2}x_{N+1} \\ R^{1/2}u_1 \\ \cdots \\ R^{1/2}u_N \end{bmatrix}$$

A local model for $e(\mathbf{u})$ is provided by the first order Taylor expansion

$$e(\mathbf{u}+p) \approx e(\mathbf{u}) + J(\mathbf{u})p \tag{12}$$

where J is the Jacobian matrix defined by

$$J(\mathbf{u}) = \begin{bmatrix} Q^{1/2}\frac{\partial x_2}{\partial u_1^T} & 0 & 0 & \cdots & 0 \\ Q^{1/2}\frac{\partial x_3}{\partial u_1^T} & Q^{1/2}\frac{\partial x_3}{\partial u_2^T} & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ Q^{1/2}\frac{\partial x_{N+1}}{\partial u_1^T} & Q^{1/2}\frac{\partial x_{N+1}}{\partial u_2^T} & \cdots & \cdots & Q^{1/2}\frac{\partial x_{N+1}}{\partial u_N^T} \\ R^{1/2} & 0 & 0 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & \cdots & 0 & R^{1/2} \end{bmatrix} \tag{13}$$

Therefore, to find the Jacobian, partial derivatives of the state with respect to the inputs are needed. However, each partial derivative is a function of a previous partial derivative so that finding the Jacobian is not as difficult as it first seems. Using the chain rule

$$\frac{\partial x_{k+1}}{\partial u_j^T} = \frac{\partial f_k(x_k,u_k)}{\partial x_k^T}\frac{\partial x_k}{\partial u_j^T} + \frac{\partial f_k(x_k,u_k)}{\partial u_k^T}\frac{\partial u_k}{\partial u_j^T}, \quad j = 0, \ldots, k \tag{14}$$

With this in mind the approximate Hessian of the Lagrangian is given by

$$L_{\mathbf{uu}}(\mathbf{u},\lambda) \approx J^T(\mathbf{u})J(\mathbf{u}) \tag{15}$$

which is positive definite since R is assumed to be positive definite.

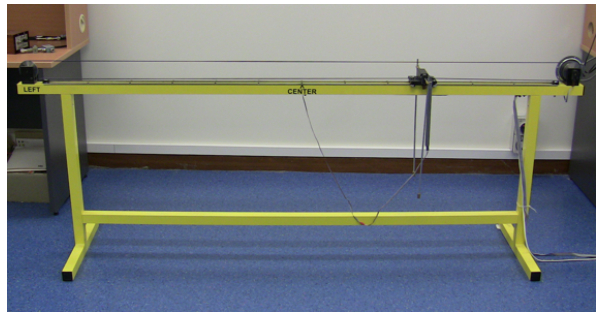

Fig. 2. Photo of Inverted Pendulum apparatus used in experiments.

IV. NMPC Applied to an Inverted Pendulum

In the remainder of this paper we consider the application of NMPC to an inverted pendulum apparatus shown in Figure 2. Controlling an inverted pendulum is challenging task because it exhibits nonlinear dynamic behaviour, it is unstable about the desired operating point (i.e. pendulum standing up), and is non-minimum phase.

The inverted pendulum apparatus consists of a cart which can freely move along a straight beam. A free body diagram is shown in Figure 3. The rigid pendulum is attached to the cart and able to rotate freely. A motor is used to drive the belt, which in turn moves the cart along the rail. The objective is to use the motor (a single input) to swing the pendulum into the upright position and then stabilise the pendulum around this upright position. The measured outputs of the system are the pendulum angle and the cart position.

From the free body diagram of the pendulum in Figure 3 we can obtain equations of motion in state-space form as

$$\dot{x}(t) = g(x(t), u(t)) \tag{16}$$

where the state vector $x^T(t) = [p(t), \theta(t), v(t), \omega(t)]$ and $p(t)$ represents cart position, $\theta(t)$ is pendulum angle, $v(t)$ is cart velocity and $\omega(t)$ is pendulum angular velocity. The mapping $g(\cdot)$ is given by

$$g(x(t),u(t)) = \begin{bmatrix} v(t) \\ \omega(t) \\ \frac{a_1 w_1(x(t),u(t)) + w_2(x(t))\cos\theta(t)}{d(x(t))} \\ \frac{w_1(x(t),u(t))\cos\theta(t) + a_2 w_2(x(t))}{d(x(t))} \end{bmatrix} \tag{17}$$

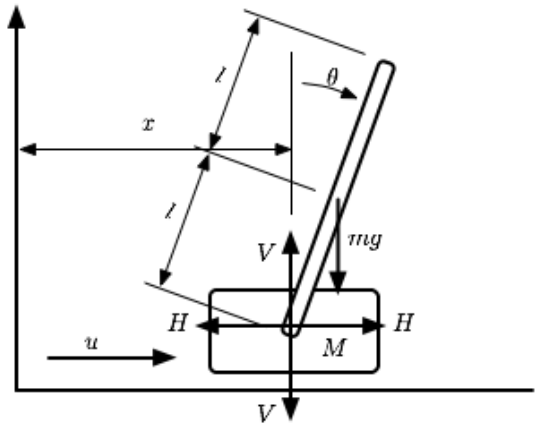

Fig. 3. Inverted Pendulum Free Body Diagram

The remaining terms are given by

$$\begin{aligned} w_1(x(t), u(t)) &= k_1 u(t) - (\omega(t))^2 \sin\theta(t) - k_2 v(t) \\ w_2(x(t)) &= g \sin\theta(t) - k_3 \omega(t) \\ d(x(t)) &= b - \cos^2\theta(t) \end{aligned}$$

and the constants

$$a_1 = \frac{J_p}{ml}, \qquad a_2 = \tfrac{1}{l}, \qquad b = \frac{J_p}{ml^2},$$
$$k_1 = \frac{c_1}{ml}, \quad k_2 = \tfrac{f_c - c_2}{ml}, \quad k_3 = \frac{f_p}{ml}$$

Where J_p is the moment of inertia of the pendulum in relation to the axis or rotation, m is the equivalent mass of the cart and pendulum, l is the distance from the axis of rotation to the centre of mass of the system, f_c is the dynamic friction of the cart, f_p is the rotational friction coefficient, c_1 is the control force to PWM signal ratio and c_2 is the control force to cart velocity ratio. This set of differential equations are derived by taking into account the centre of mass of the system as a whole. It is not assumed that the centre of gravity of the pendulum is the middle of the rod.

A. Discrete-Time Model

The above model describes the dynamic behaviour in continuous-time. However, the required format for the NMPC controller outlined in Section II is a discrete-time model (1). To achieve the latter requirement we sample the continuous-time system using Euler's method; this is done at a much faster rate than the control interval to provide better accuracy.

More precisely, if the control interval is Δ seconds, then we update the input at $t = k\Delta$ for $k = 1, \ldots$, so that

$$u(k\Delta) = u_k, \qquad x(k\Delta) = x_k \tag{18}$$

Consider the case where $k = 0$, so that we start at u_0 and x_0. According to (1) we want a mapping from u_0 and x_0 to $x_1 = x(\Delta)$. This will be achieved by repeated application of Euler's method sampled at $\delta = \Delta/M$ for some suitable integer $M \geq 1$. This corresponds to

$$\begin{aligned} x((i+1)\delta) &= x(i\delta) + \delta g(x(i\delta), u(i\delta)) \\ &= x(i\delta) + \delta g(x(i\delta), u_0) \end{aligned}$$

where the latter equality comes from the constant control action over $\Delta = M\delta$ seconds, i.e.

$$u(i\delta) = u_0 \quad \text{for} \quad i = 0, \ldots, M-1$$

Therefore, in terms of (1), we have that the discrete-time mapping $f_0(x_0, u_0)$ is given by the above equations where $x(0) = x_0$ and $x_1 = x(M\delta)$. Generalising this for all $k \geq 0$ gives the required function f_k.

B. State Estimation

In order to apply the NMPC strategy outlined in Section II we must have access to the state at time k. While two of the state values are measured directly, namely the cart position $p(t)$ and pendulum angle $\theta(t)$, we need to estimate the cart velocity $v(t)$ and angular velocity $\omega(t)$. Again with a view to using the most straightforward approach, we simply approximate the time derivatives via a finite difference approximation

$$\widehat{v}_k = \frac{p_k - p_{k-1}}{\Delta} \qquad \widehat{\omega}_k = \frac{\theta_k - \theta_{k-1}}{\Delta} \tag{19}$$

In high noise situations this approximation is a very bad idea, but for this application it suffices.

C. Control Cost Function

We would like the state $x(t)$ to be at the origin, which corresponds to the cart position at the centre of the beam, the pendulum angle of 0 radians (i.e. upright), the cart not moving and the pendulum not rotating. Furthermore, we would like to be economical with control action and thus penalise input movements. This objective can be described in terms of the cost function in (3) via the choices

$$Q = \mathrm{diag}\{q_1,\ q_2,\ q_3,\ q_4\}, \qquad R = r \tag{20}$$

Since it is important that the pendulum angle is zero, q_2 receives a high value. It is less important, but not insignificant, that the cart position is zero, so q_1 has the next highest value. Cart velocity and pendulum angular velocity are not so important, so $q_{3,4}$ receive a low value.

For the experimental results shown in Section V this corresponds to

$$q_1 = 1, \quad q_2 = 10, \quad , q_3 = q_4 = 10^{-4}, \quad r = 0.1 \tag{21}$$

D. Constraints

While the above model is good at describing the dynamics, it does not capture the whole situation physically. Importantly, there are physical limits on the control input and the cart position, which correspond to constraints on the supply current to the motor and the length of the beam, respectively. The input to the motor is constrained to lie between $-0.5 \leq u(t) \leq 0.5$ and the cart position must lie between $-0.7 \leq p(t) \leq 0.7$.

E. Solving the Problem

For the online implementation we employed the SQP method described in Section III, but we restricted the number of iterations to be no more than four. This choice was made based on the average time taken per iteration, and the sample time of 25ms. It would be expected that this rather arbitrary choice of the number of allowed iterations results in poor control performance, yet this was not observed in practise. We believe that this curious situation deserves further attention.

The prediction horizon is set to $N = 60$ samples (1.5s based on 25ms sample interval). With an additional slack variable for soft state constraints, this relates to 61 variables and 241 inequality constraints. The QP problems are solved via an active-set strategy based on [19], [20].

The algorithm was implemented in C on a 2.33GHz MacBook Pro and connected to the pendulum apparatus via ethernet cable, as explained in the next section.

F. Communication with Pendulum

Figure 4 shows a schematic diagram of communication between a Laptop (PC) and the pendulum apparatus. The connection from the pendulum hardware to the power interface box includes an analog signal to drive a 22.5V DC motor and 4 digital signals, two for each of the optical encoders used to measure cart position and pendulum angle.

The power interface box is connected to a custom made Signal Conversion Device via a 20-pin connector. This device connects the Laptop via ethernet cable to the Power Interface Box and is based on Altera Stratix II Development Board.

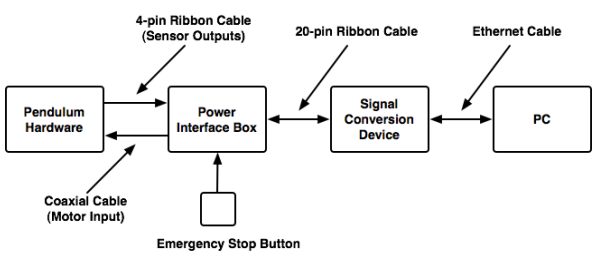

Fig. 4. Pendulum to PC complete path.

V. Results

In this section we provide some experimental results from applying the NMPC controller described above to the inverted pendulum shown in Figure 2. By way of summary, there are 4-states, 1-input, the sample time is $\Delta = 25$ms, the prediction horizon is $N = 60$ samples, the input is constrained via $|u(t)| \leq 0.5$ and the cart position is constrained via $|p(t)| \leq 0.7$. The SQP solver is limited to 4-iterations and is trying to solve a nonlinear optimisation problem with 61-variables and 241-inequality constraints. All plots in this section show data recorded from the physical apparatus by the Laptop.

In Figure 5 the initial position of the cart is in the centre of the beam with the pendulum tip is down (in the stable position). NMPC is switched on and the response can be seen in the figure. Note that both the DC motor input and the cart position hit their respective limits.

To gauge the utility of the NMPC controller, a large disturbance was manually applied to the pendulum tip whilst in the upright position. Figure 6 shows the response to this disturbance. Note that the cart position and input obey their respective limits while the pendulum is promptly returned to the upright position.

Possibly of more interest, a histogram of the time taken by the SQP algorithm for this latter disturbance experiment is shown in the Figure 7(top). Four peak areas are visible which correspond to the number of iterations. Clearly the algorithm spent most of its time in the 3-4 iterations region. Also note that none of the times exceed the limit of 25ms. The variability comes from numerous sources, but not least of all from the non-deterministic back-stepping algorithm line search used within SQP. A histogram of the time taken to compute the cost function is shown in Figure 7(bottom). The variability seen here is now at the operating system level.

VI. Conclusions

This paper details the application of NMPC to an inverted pendulum apparatus. While the good performance of NMPC for this application may be of independent interest, the key point is that a reasonably challenging control problem can be dealt with via NMPC in real-time on a modest hardware platform at a 40Hz sample rate. A video of this application can be found at `http://sigpromu.org/mpc/pilot.html`.

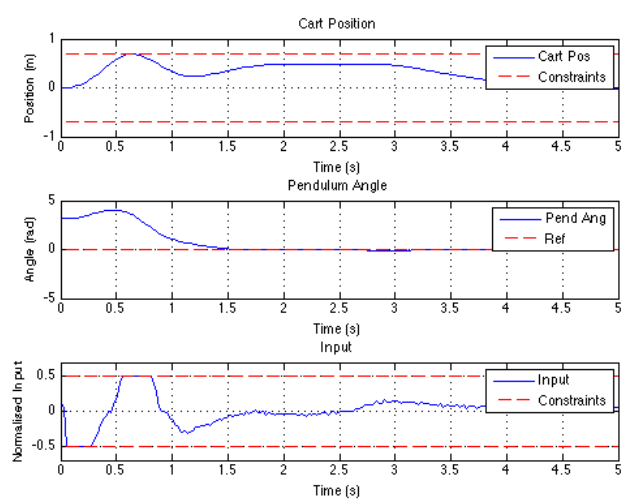

Fig. 5. NMPC Inverted Pendulum Swing Up Response

References

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

[2] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice Hall, 2002.

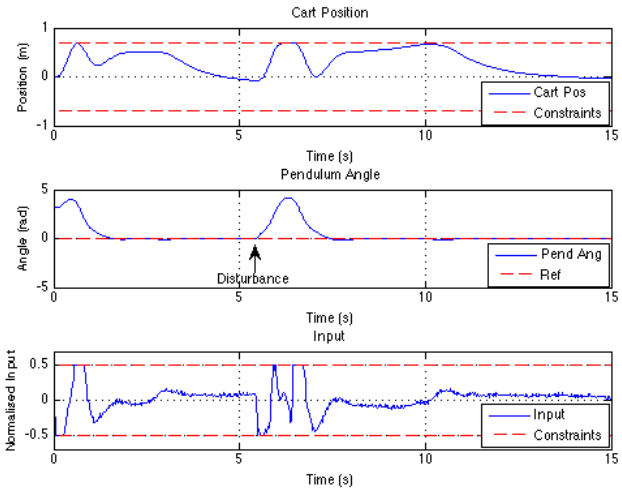

Fig. 6. NMPC controller with a larger manual disturbance introduced at the tip of the pendulum and hitting state constraints.

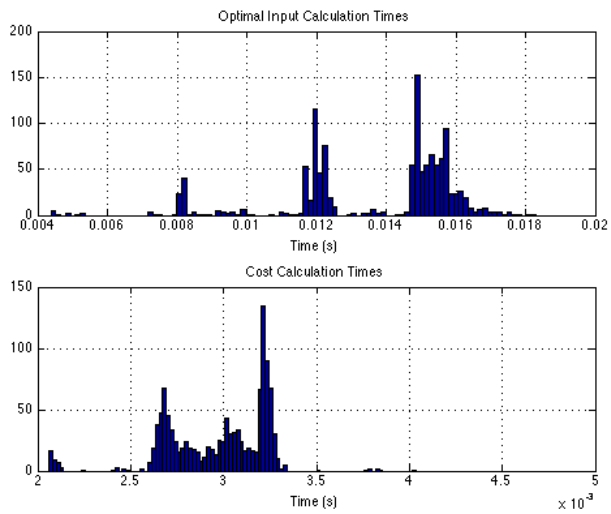

Fig. 7. Timing histogram of both optimal input calculation and cost calculation.

[3] S. Qin and T. Badgewell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, pp. 733–764, 2003.

[4] A. G. Wills, D. Bates, A. J. Fleming, B. Ninness, and S. Moheimani, "Model predictive control applied to constraint handling in active noise and vibration control," 2005.

[5] T. A. Johansen, W. Jackson, R. Schreiber, and P. Tondel, "Hardware synthesis of explicit model predictive controllers," *IEEE Transactions on Control System Technology*, vol. 15, no. 1, pp. 191–197, January 2007.

[6] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *International Journal of Robust and Nonlinear Control. In Press.*, 2008.

[7] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[8] T. B. Blanco, P. Willems, B. D. Moor, and J. Berlamont, "Flooding prevention of the demer river using model predictive control," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[9] N. Haverbeke, T. V. Herpe, M. Diehl, G. V. den Berghe, and B. D. Moor, "Nonlinear model predictive control with moving horizon state and disturbance estimation - application to the normalization of blood glucose in the critically ill," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[10] G. Hafidi, S. Tebbani, D. Dumur, and A. V. Wouwer, "Nonlinear model predictive control applied to e. coli cultures," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[11] A. Chemori and N. Marchand, "Global discrete-time stabilization of the pvtol aircraft based on fast predictive control," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[12] L. Coetzee, I. Craig, and E. Kerrigan, "Nonlinear model predictive control of a run-of-mine ore milling circuit," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[13] B. Guerreiro, C. Silvestre, and R. Cunha, "Terrain avoidance model predictive control for autonomous rotorcraft," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[14] S. Jung and J. T. Wen, "Nonlinear model predictive control for the swing-up of a rotary inverted pendulum," *Journal of Dynamic Systems, Measurement, and Control*, vol. 126, pp. 666–673, 2004.

[15] T. Murao, H. Kawai, and M. Fujita, "Predictive visual feedback control with eye-in/to-hand configuration via stabilizing receding horizon approach," *Proceedings of the 17th World Congress The International Federation of Automatic Control*, 2008.

[16] J. H. Park and K. D. Kim, "Biped robot walking using gravity-compensated inverted pendulum mode and computed torque control," *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, 1998.

[17] P. T. Boggs and J. W. Tolle, "Sequential quadratic programming," *Acta Numerica*, vol. 4, pp. 1–51, 1996.

[18] J. Nocedal and S. J. Wright, *Numerical Optimisation*. Springer-Verlag New York, Inc., 1999.

[19] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *MatheMatical Programming*, vol. 27, pp. 1–33, 1983.

[20] M. J. D. Powell, "On the Quadratic Programming Algorithm of Goldfarb and Idnani," *Mathematical Programming Study*, vol. 25, pp. 46–61, 1985.